الجامعة الافتراضية السورية

SYRIAN VIRTUAL UNIVERSITY

مقدمة في نظم التشغيل

ISSN: 2617-989X

Books

مقدمة في نظم التشغيل

من منشورات الجامعة الافتراضية السورية

الجمهورية العربية السورية 2018

هذا الكتاب منشور تحت رخصة المشاع المبدع – النسب للمؤلف – حظر الاشتقاق (CC– BY– ND 4.0)

https://creativecommons.org/licenses/by-nd/4.0/legalcode.ar

يحق للمستخدم بموجب هذه الرخصة نسخ هذا الكتاب ومشاركته وإعادة نشره أو توزيعه بأية صيغة وبأية وسيلة للنشر ولأية غاية تجارية أو غير تجارية، وذلك شريطة عدم التعديل على الكتاب وعدم الاشتقاق منه وعلى أن ينسب للمؤلف الأصلي على الشكل الآتي حصرا:

الإجازة في تقانة المعلومات، من منشورات الجامعة الافتراضية السورية، الجمهورية العربية السورية، 2018

متوفر للتحميل من موسوعة الجامعة https://pedia.svuonline.org/

Introduction to Operating systems

Publications of the Syrian Virtual University (SVU)

Syrian Arab Republic, 2018

Published under the license:

Creative Commons Attributions- NoDerivatives 4.0

International (CC-BY-ND 4.0)

https://creativecommons.org/licenses/by-nd/4.0/legalcode

Available for download at: https://pedia.svuonline.org/

ISSN: 2617-989X

الفهرس

- مقدمة حول نظم التشغيل.....1
  - ماهو نظام التشغيل.....3
  - النظام الحاسوبي ونظام التشغيل.....3
  - نظام التشغيل(ملاءمة أم فعالية).....6
  - التصنيفات الرئيسية لأنواع نظم التشغيل.....6
  - نظم المهمة الوحيدة.....7
  - نظم المهمات المتعددة ونظم المشاركة بزمن المعالج.....10
  - نظم الحواسب الشخصية.....12
  - النظم التفرعية.....14
  - النظم الموزعة.....15
  - نظم الزمن الحقيقي.....16
  - التمارين.....17
- بنية نظام التشغيل.....21
  - عمل النظام الحاسوبي.....23
  - بنية الدخل والخرج.....24
  - وحدات التخزين في النظم الحاسوبية.....25
  - هرمية التخزين.....26
  - اتساق الذاكرة الخبيئة.....27
  - حماية العتاديات.....28
  - حماية الدخل والخرج.....28
  - حماية الذاكرة ووحدة المعالجة.....28
  - ادارة الاجرائيات(مكونات نظام التشغيل).....29
  - ادارة الذاكرة الرئيسية.....30
  - ادارة الملفات.....30
  - خدمات نظام التشغيل.....31
  - استدعاء النظام.....31
  - برامج النظام.....32
  - التمارين.....33
- أنظمة تشغيل الحواسب الشخصية.....37
  - أنظمة تشغيل الحواسب الشخصية.....39
  - مقدمة عن نظام التشغيل.....39
  - مهمات نظام التشغيل.....40
  - واجهة المستخدم.....40
  - ادارة المجلدات والملفات.....41
  - ادارة التطبيقات.....42
  - ادارة العتاد.....42
  - موارد النظام.....43
  - ارقام طلبات المقاطعة.....44
  - عناوين الذاكرة.....46
  - عناوين الدخل والخرج.....47
  - قناة الوصول المباشر للذاكرة.....47
  - اقلاع النظام.....47
  - ايجاد وتحميل نظام التشغيل.....48
  - اسئلة.....49
  - تمارين عملية.....50
  - التمارين.....52
- مسالك التنفيذ.....55
  - مسالك التنفيذ.....57

ISSN: 2617-989X

الفهرس

- فوائد مسالك التنفيذ.....57
- مقارنة بين الاجراءات ومسالك التنفيذ.....58
- مسالك على مستوى النواة.....58
- مسالك على مستوى المستخدم.....59
- نماذج تعدد مسالك التنفيذ.....59
- ايقاف مسالك التنفيذ.....62
- مجمع مسالك التنفيذ.....62
- مكتبة التعامل مع مسالك التنفيذ.....62
- التمارين.....64

❖ ادارة العتاديات.....67

- المودم.....69
- سرعة المودم.....70
- اعدادات منفذ الاتصال.....71
- تنصيب المودم.....72
- ادوات اختبار المودم.....72
- تنصيب كرت الشبكة.....73
- مشاركة الملفات على الشبكة.....73
- أسئلة.....74
- تنصيب ومشاركة الطابعة.....74
- استخدام طابعة مشتركة على الشبكة.....75
- مشاكل الطابعة.....76
- تمارين.....77
- اختيار العتاديات المناسبة للحاسب.....78
- اختيار البرمجيات المناسبة للحاسب.....78
- تمارين.....79
- الحواسب المحمولة.....80
- الحواسب المسطحة.....80
- المساعد الرقمي الشخص(PAD).....81
- التمارين.....82

❖ جدولة وحدة المعالجة.....85

- جدول وحدة المعالجة.....87
- دفعات وحدة المعالجة والدخل والخرج.....87
- مجدول وحدة المعالجة.....87
- مسند المهمات.....88
- معايير الجدولة.....88
- خوارزميات الجدولة.....89
- الجدولة وفق أرتال متعددة المستوى.....91
- الجدولة وفق ارتال متعددة المستوى بتغذية راجعة.....92
- تمرين.....93
- التمارين.....95

❖ ادارة الاجراءات.....99

- مفهوم الاجراء.....101
- الاجراء.....101
- حالة الاجراء.....102
- كتلة تحكم الاجراء.....103
- جدولة الاجراءات.....104
- المجدولات.....105
- تبديل السياق.....107

ISSN: 2617-989X

الفهرس

- انشاء الاجراءات.....108
- انهاء الاجراءات.....109
- الاجراءات المتعاونة.....109
- الاتصال بين الاجراءات.....109
- الاتصال المباشر بين الاجراءات.....110
- الاتصال اللامباشر بين الاجراءات.....111
- التزامن بين الاجراءات المتصلة.....112
- رتل الانتظار.....112
- التمارين.....113

❖ أنظمة الملفات.....117

- اقراص التخزين.....119
- نظام الملفات.....119
- مستويات نظام الملفات.....120
- طرق حجز كتل الملفات.....121
- أنظمة الملفات الشهيرة.....122
- نظام ملفات الـ CD-ROM.....122
- بنية المجلد في نظام ISO 9660.....123
- نظام الملفاتCP/M.....124
- نظام ملفات MS-DOS.....125
- جدول تخصيص الملفات في نظام الملفاتMS-DOS.....126
- بنية جدول تخصيص الملفات.....127
- نظام ملفاتWINDOWS 98.....128
- أسماء الملفات في WINDOWS 98.....129
- نظام ملفات UNIX V7.....130
- نظام ملفات WINDOWS 2000.....132
- البنية الفيزيائية لنظام الملفاتNTFS.....133
- بنية جدول الملفات الرئيسيMFT.....134
- أهم واصفات الملف.....134
- البحث عن ملف في نظام NTFS.....135
- الملفات المترفعة في نظام NTFS.....136
- المجلدات.....137
- ألية حماية المعطيات من الضياع.....137
- ضغط الملفات في نظام NTFS.....138
- استخدام ضغط الملفات.....139
- نظام تشفير الملفات EFS.....139
- استخدام نظام تشفير الملفات.....140
- ميزات نظام NTFS.....141
- ميزات نظام FAT.....141
- اختيار نظام الملفات.....142
- التحويل بين أنظمة الملفات.....142
- التمارين.....143

❖ ادارة الذاكرة.....146

- بنية الخزن.....148
- هرمية الخزن.....148
- اتساق الخبيئة.....149
- ادارة الذاكرة الرئيسية.....149
- ادارة الخزن الثانوي.....150
- عناوين الذاكرة الفيزيائية و المنطقية.....151

ISSN: 2617-989X

الفهرس

- اقسام الذاكرة.....152
- التبديل.....153
- صفحات الذاكرة.....154
- جدول الصفحات.....155
- حماية الصفحات.....156
- التقطيع.....157
- تمرين.....157
- الذاكرة الافتراضية.....158
- نظام طلب الصفحات.....159
- ادارة مساحة التبديل.....160
- حجز أطر الذاكرة.....160
- استبدال الصفحات.....161
- خوارزميات التبديل.....162
- تمرين.....162
- التمارين.....163

❖ تزامن الاجراءات.....167

- الاجراءات المتعاونة.....169
- مشكلة المنتج والمستهلك.....169
- مشكلة المقطع الحرج.....170
- حلول لمشكلة المقطع الحرج في حالة اجراءين.....171
- حلول في حالة عدة اجرائيات(خوارزمية المخبز).....174
- تعليمات التزامن على مستوى العتاديات.....175
- سيمافور.....178
- التحقيق البرمجي للسيمافور.....178
- الاقفال المتبادل.....180
- مشكلات التزامن – مشكلة المنتج والمستهلك.....181
- مشكلة القراء والكتاب.....182
- مشكلة الفلاسفة المفكرين والطاعمين.....184
- المراقب.....185
- نشاط (مسائل).....186
- التمارين.....188

مقدمة حول نظم التشغيل

ISSN: 2617-989X

الكلمات المفتاحية:

نظام تشغيل، عتاديات، برمجيات، زمن المعالجة، تصادم، نواة، النظم ذات المهمة الوحيدة، النظم متعددة المهمات، نظم المشاركة بزمن المعالج، نظم الحاسبات الشخصية، النظم التفرعية، النظم الموزعة، نظم الزمن الحقيقي، المشاركة بالموارد، تسريع المعالجة، الموثوقية، الذاكرة الافتراضية، المنصات، محطات العمل، الميقاتية، معالج صِغَري، حاسب، مشاركة الأعباء، حساسات، نظم الزمن الحقيقي القاسية، نظم الزمن الحقيقي المرنة، نظام متعدد المعالجات متناظر، نظام متعدد المعالجات غير متناظر.

ملخص:

سنتستعرض في هذه الجلسة مفهوم نظام التشغيل من حيث تعريفة وأنواعه والاستراتيجيات المستخدمة من أجل تصميم الأنواع المختلفة من نظم التشغيل الحاسوبية.

أهداف تعليمية:

يتعرف الطالب في هذا الفصل على ما يلي:

- ما هو نظام التشغيل.
- ما هو النظام الحاسوبي وأين يتوضع نظام التشغيل في بنيته.
- كيف كانت بدايات تصميم نظم التشغيل مع النظم ذات المهمة الوحيدة.
- لمحة عن تطور نظم التشغيل ونموها لتدعم عدة مهمات في آن واحد.
- لمحة عن نظم المشاركة بزمن المعالج وكيف تعمل.
- لمحة عن نظم الحاسبات الشخصية.
- لمحة عن النظم التفرعية.
- لمحة عن النظم الموزعة.
- لمحة عن نظم الزمن الحقيقي وأنواعها.

1. ما هو نظام التشغيل

- تعريف:

يُعرَّف نظام التشغيل بأنه برنامج يدير عتاديات الحاسوب بحيث يوفر البرمجيات والتطبيقات الضرورية لتشغيل هذه العتاديات، كما يعمل كوسيط بين المستخدم والحاسوب يسمح للمستخدم باستثمار الحاسوب وتطبيقاته.

- أسلوب تصميم نظام التشغيل:

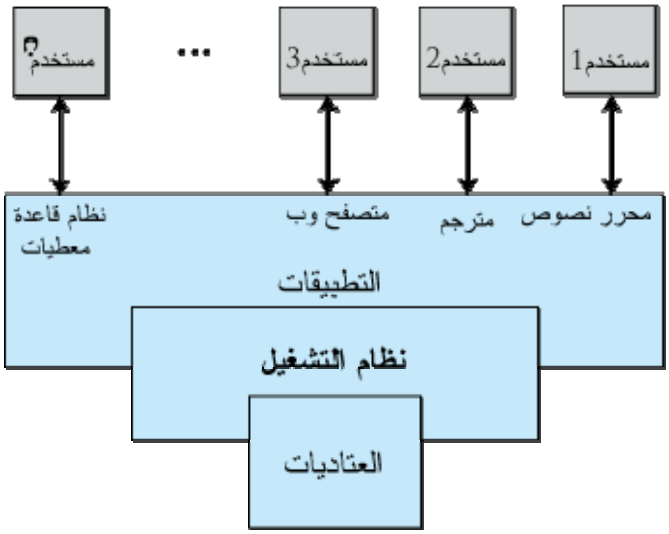
يختلف تصميم نظام التشغيل حسب البيئة التي يُفترض أن يعمل عليها، إذ يصمم نظام التشغيل الذي يعمل على المخدمات على نحو يستطيع فيه استثمار العتاديات بالشكل الأمثل، في حين يصمم نظام التشغيل المُعدّ للعمل على الحاسبات الشخصية ليدعم تطبيقات متنوعة. بالتالي نلاحظ اختلاف وجهة التصميم لتكون إما ملائمة للمُستخدم النهائي في حالة الحواسب الشخصية أو فعالة في استثمارها للعتاديات في حالة المخدمات.

2. النظام الحاسوبي ونظام التشغيل

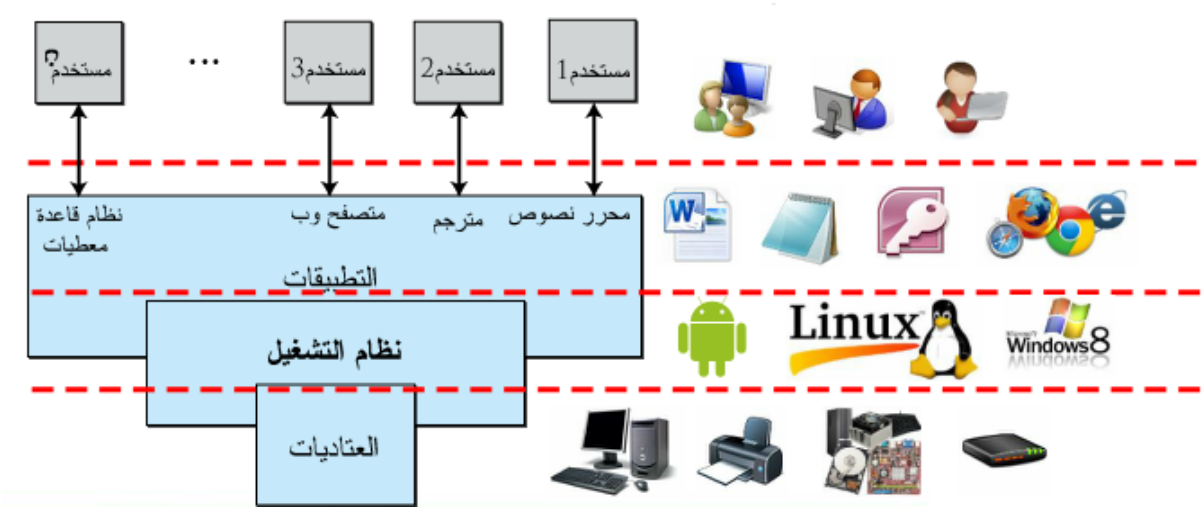
يعتبر نظام التشغيل جزءاً هاماً من كافة الأنظمة الحاسوبية، بحيث يمكن أن نقسم النظام الحاسوبي إلى أربعة مكونات رئيسية وهي:

- العتاديات
- نظام التشغيل
- التطبيقات
- المستخدمين

يمثل الشكل التالي بنية توضيحية للنظام الحاسوبي، ويُبين توضع نظام التشغيل ضمن تلك البنية:



يتولى نظام التشغيل مهمة الإشراف والمراقبة وتوفير البيئة الملائمة للتطبيقات والمستخدمين لكي ينفذوا أعمالهم ويستثمروا موارد الحاسوب وتطبيقاته. إذ تشكل العتاديات في النظام الحاسوبي الموارد التي يجري الاعتماد عليها عند استثمار الحاسب، وهي تشمل وحدة المعالجة المركزية، والذاكرة، وتجهيزات الدخل/خرج وغيرها، في حين تعبر التطبيقات عن الأدوات التي يستخدمها المستخدمون النظام لاستثمار الموارد.



على كل حال، يمكن النظر إلى نظام التشغيل كمحصص للموارد، وكنظام تحكُم، و كنواة لتشغيل التطبيقات الحاسوبية:

- نظام التشغيل كمحصص للموارد:

يتكون النظام الحاسوبي من العديد من الموارد العتادية والبرمجية (وحدة معالجة مركزية، وحدات خزن معطيات، ذاكرة رئيسية ...الخ)، حيث يتولى نظام التشغيل مهمة إدارة تلك الموارد وتوزيعها على المستخدمين بالشكل الذي يحقق الإنصاف ويضمن فعالية أداء النظام الحاسوبي. وتبرز أهمية وقدرة نظام التشغيل على الإدارة في أسلوب معالجته للطلبات التي يمكن أن تؤدي إلى تعارض في استخدام الموارد.

- نظام التشغيل كبرنامج تحكم:

يمكن النظر إلى نظام التشغيل كبرنامج يتحكم بكيفية تنفيذ برامج المستخدمين بهدف منع حدوث الأخطاء، ومنع الاستخدام غير السليم للحاسب وخاصة فيما يتعلق باستخدام تجهيزات الدخل/خرج والتحكم فيها.

- نظام التشغيل كنواة:

إن المفهوم الذي يعتبر نظام التشغيل أداة تحصيص أو أداة تحكم يولد بالضرورة تصوراً حول مكونات نظام التشغيل من البرمجيات، لذا يجدر بنا التنويه إلى التعريف الأكثر شيوعاً لنظام التشغيل –الذي يُطلق عليه اسم النواة– والذي يشير لنظام التشغيل على أنه البرنامج الذي يكون بحالة تنفيذ دائمة والذي تعمل تحت إشرافه التطبيقات البرمجية الأخرى.

3. نظام التشغيل (ملاءمة أم فعالية)

يعتبر تعريف أنظمة التشغيل من خلال وصف أعمالها ومهامها أكثر سهولة من وصفها اعتماداً على مكوناتها، فقد تم تطوير أنظمة التشغيل عبر الخمسين سنة الماضية اعتماداً على اتجاهين أساسيين، هما:

- أنظمة تشغيل تحاول جدولة النشاطات الحاسوبية لضمان أداء أمثلي للنظام ككل
- وأنظمة تشغيل أخرى تحاول إيجاد بيئة عمل مناسبة لتطوير وتنفيذ التطبيقات بما يلائم المستخدم.

نلاحظ مما سبق وجود تصنيف أنظمة التشغيل بحسب الملاءمة أو الفعالية، فعندما ننظر إلى أنظمة تشغيل حاسوبية بسيطة كتلك التي تعمل على الحاسب الشخصي، نجد أن الهدف الأساسي منها هو خدمة المستخدم وتأمين احتياجاته بالطريقة الأسهل والأكثر بساطةً، ولا يهمنا في هذه الحالة ما هو معدّل استهلاك الموارد لأن هدفنا الرئيسي هو الملاءمة.

أما بالنظر إلى أنظمة التشغيل التي تعمل على المحطات الكبيرة أو المنصّات التي يلج إلى مواردها عدد كبير من المستخدمين في آن واحد، فلابد هنا من إعطاء مسألة إدارة الموارد أهمية كبيرة وزيادة معدّل استخدام الموارد لضمان استخدام زمن وحدة المعالجة المركزية، والذاكرة، ووحدات الدخل/خرج، بالشكل الأكثر إنصافاً وفعاليةً.

4. التصنيفات الرئيسة لأنواع نظم التشغيل

تطورت نظم إدارة الحاسبات تطوراً كبيراً منذ أن نشأت وحتى الآن، سواء كان ذلك التطور يؤثر على طبيعة نظام التشغيل بحد ذاته، أو كان يعبر عن جيل آخر من الأنظمة يقدم خدمات مغايرة أكثر تطوراً وتنوعاً من حيث دعمها للتطبيقات المختلفة وما تقدمه من مهمات، تجارية كانت أم علمية.

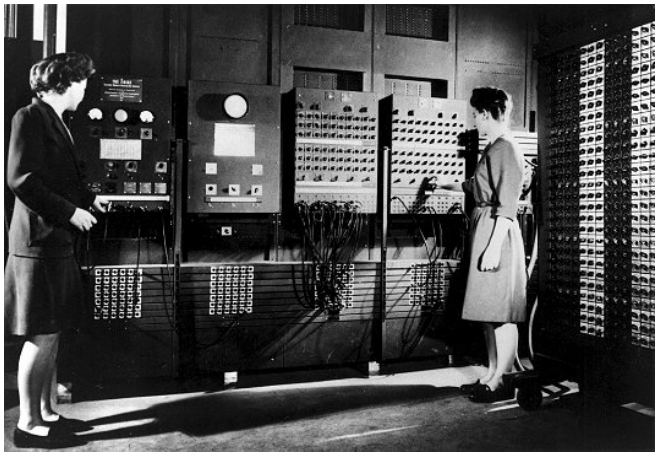
لقد مرت دورة حياة نظم التشغيل بالعديد من المراحل فبدأت من خلال النظم ذات المهمة الوحيدة، وتطورت بعد ذلك لتصبح نظم تدعم عدة مهمات في آن واحد، ثم بدأت تتشارك بالموارد كالمعالج أو الذاكرة، وترافق ذلك مع أجيال نظم الحواسب الشخصية التي انتشرت انتشاراً واسعاً بين المستخدمين.

سنراعي في الشرائح التالية دراسة كل نوع من أنواع تلك النظم وصولاً إلى الأنواع الأكثر تعقيداً كالنظم التفرعية والنظم الموزعة ونظم الزمن الحقيقي التي تسمح للمستخدم بالتفاعل المباشر مع الحاسب.

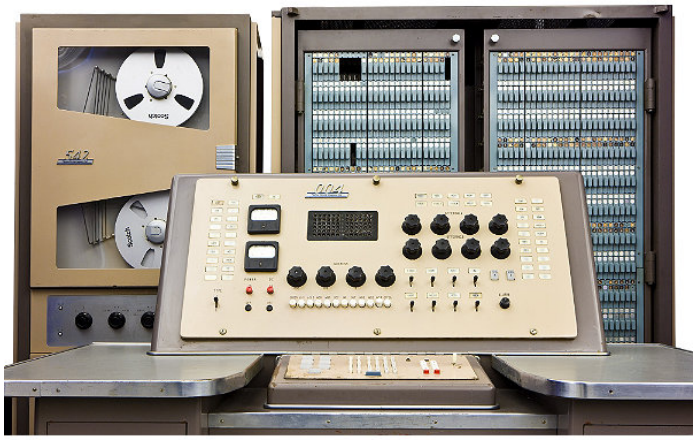
5. نظم المهمة الوحيدة

تعبر نظم المهمة الوحيدة عن نظم التشغيل البسيطة التي كان الحاسب فيها يقوم بتنفيذ تطبيق واحد فقط.

تمثل نظم المهمة الوحيدة الشكل الأول لنظم التشغيل عند بداية ظهورها، حيث كانت الحاسبات في ذلك الوقت ذات حجوم ضخمة جداً وكانت تُدار من خلال واجهات تعليمات خاصة، أما أدوات الدخل/خرج فقد كانت تتمثل بقارئات البطاقات المثقبة وسواقات الأشرطة، كما كانت وسائط التخزين تتمثل عموماً بالبطاقات المثقبة.

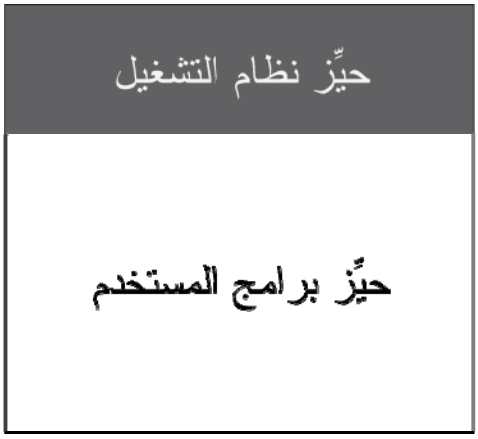


حاسب من الجيل الأول



حاسب من الجيل الثاني

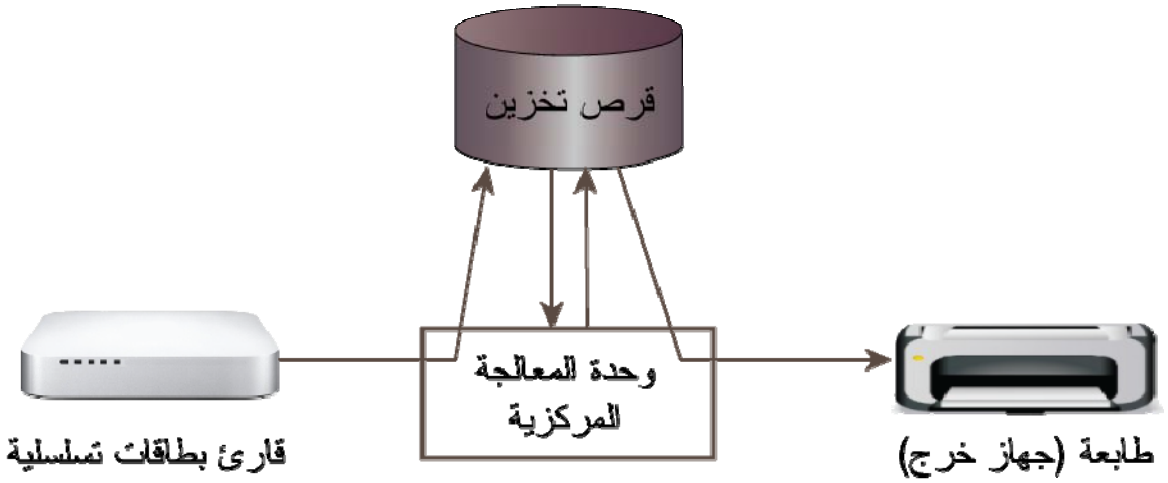
كان نظام التشغيل في ذلك الحين من البساطة إلى حدّ كانت مهماته تنحصر بنقل التحكم من عمل إلى آخر آلياً، كما أنه كان يقيم في الذاكرة بشكل دائم بحيث يُخصص له الحيّز المطلوب بشكل مستمر:



أما بالنسبة لأسلوب تنفيذ الأعمال في نظام التشغيل وحيد المهمة، فقد كان يتم جمع الأعمال المتماثلة مع بعضها البعض وتنفيذها دفعة واحدة ضمن مجموعة واحدة وذلك لتسريع المعالجة، مما كان يتطلب من المبرمجين أن يتركوا برامجهم لمشغّل النظام والذي كان يقوم بدورة بجمع العمليات المتماثلة وتنفيذها دفعة واحدة كلما كان استخدام الموارد متاحاً، بحيث يتم إرسال خرج كل عمل إلى المبرمج المناسب.

تتميز وحدة المعالجة في هذه النظم بأنها تكون غالباً بلا عمل بحيث أن سرعة تجهيزات الدخل/خرج الميكانيكية أبطأ بالضرورة من سرعة وحدة المعالجة الإلكترونية، وقد استمرت المعاناة من هذه المشكلة مع مرور السنين ومع استمرار تطور أنظمة التشغيل بحيث مازال التزايد في سرعة المعالج أكبر بكثير من التطورات التي تشهدها تجهيزات الدخل/خرج.

ظهرت في هذه الفترة أيضاً تقنية جديدة أطلق عليها اسم تقنية القرص، والتي ساهمت بإمكانية جدولة كافة الأعمال التي ينبغي أن يقوم بها نظام التشغيل من خلال تخزينها على القرص بدلاً من إبقائها في قارئ البطاقات التسلسلي كما يوضح الشكل التالي:



6. نظم المهمات المتعددة ونظم المشاركة بزمن المعالج

تعبر نظم المهمات المتعددة عن نظم التشغيل التي تستثمر الموارد على نحو يزيد من معدل استخدام وحدة المعالجة المركزية وبحيث يتم تنفيذ إجرائية أو عمل في كل وقت.

يجري في هذا النوع من نظم التشغيل استغلال الموارد بالشكل الأمثل كما يلي:

1. قراءة مجموعة من الأعمال المخزنة على القرص ونقلها إلى الذاكرة لكي يتم تنفيذها معاً.
2. اختيار أحد تلك الأعمال والبدء بتنفيذها.
3. الإنتقال إلى العمل التالي عندما يتطلب العمل قيد التنفيذ إجراء عملية دخل/خرج معينة، وذلك لكي يتم استغلال المعالج ريثما تنتهي عملية الدخل/الخرج والتي تتميز بفارق كبير في زمن التنفيذ عن الزمن الذي يستهلكه المعالج.
4. يتم تكرار الأسلوب السابق طالما وُجدت أعمال للتنفيذ.

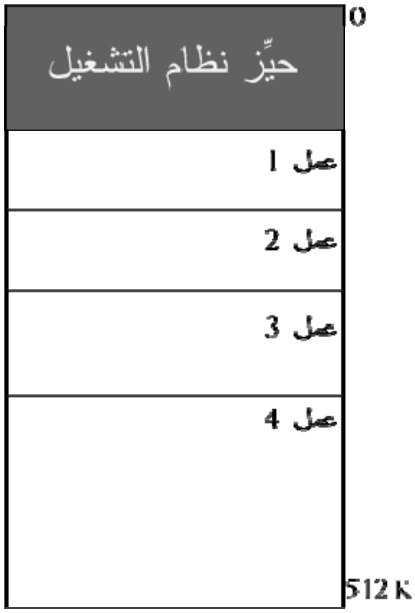
يتم تخزين الأعمال في قرص تخزين، كما يتم انتقاء مجموعة من تلك الأعمال ونقلها إلى الذاكرة لكي يتم تنفيذها معاً، ولا يتم نقل كافة الأعمال المخزنة لأنه غالباً ما تكون المعطيات المخزنة على القرص أكبر من سعة التخزين في الذاكرة؛ تسمى عملية انتقاء الأعمال التي ينبغي اختيارها أولاً بجدولة الأعمال.

يمكن أن تتوافر عدة أعمال جاهزة للتنفيذ في آن واحد، مما يتطلب وجود أسلوب لانتقاء المهمة التي ينبغي تنفيذها أولاً من بين المهمات الموجود في الذاكرة، يطلق على هذه العملية اسم جدولة وحدة المعالجة المركزية.

إن توافر عدة أعمال في الذاكرة معاً بالإضافة إلى نظام التشغيل، يولد بالضرورة مهاماً إضافية تتعلق بأسلوب إدارة الذاكرة.

سيتم تخصيص جلسات خاصة لتفصيل مفهوم جدولة الأعمال وجدولة وحدة المعالجة المركزية بالإضافة إلى مفهوم إدارة الذاكرة.

يمثل الشكل التالي ذاكرة نظام تشغيل متعدد المهمات، بحيث نلاحظ فيها كيفية احتلال نظام التشغيل لحيز من الذاكرة في حين تتقاسم بقية الأعمال المساحات الأخرى المتوفرة:



يطلق على نظم المهمات المتعددة اسم نظم المشاركة بزمن المعالج، بحيث تسمح نظم المشاركة بزمن المعالج لأكثر من مستخدم بالتشارك بالحاسب في نفس الوقت، وهنا تبرز خاصة هامة وهي الشفافية في استخدام الحاسب بحيث يعتقد كل مستخدم بأنه يمتلك الحاسب ككل مع أن الحاسب مشترك بين عدّة مستخدمين.

تزداد الأمور تعقيداً في نظم المشاركة بزمن المعالج خاصة فيما يتعلق بمسائل إدارة الذاكرة وتأمين الحماية بحيث يكون لكل مستخدم برنامج واحد على الأقل في الذاكرة يطلق عليه اسم الإجراء.

تتميز نظم المشاركة بزمن المعالج بإمكانية تبديل وحدة المعالجة المركزية بين الإجراءات بتواتر عالٍ، مما يسمح لكل مستخدم أن يتفاعل مع تطبيقاته أثناء تنفيذها.

لابد من الإشارة إلى المشاكل الأخرى التي يمكن أن نلاحظها فيما يتعلق بإدارة الذاكرة في نظم المشاركة بزمن المعالج، بحيث ينبغي هنا تقسيم الذاكرة بين المستخدمين وكذلك بين الإجراءات، مما يزيد من احتمال أن تبلغ الذاكرة حجماً يزيد عن الحجم الفيزيائي المخصص لها، وهنا يبرز مفهوم الذاكرة الافتراضية (التي سنتطرق إليها لاحقاً) والتي تؤمن حلاً لهذه المشكلة وذلك بتخصيص جزءاً من القرص للعمل كذاكرة يتم التخاطب معها وتبادل المعطيات.

هناك العديد من الخصائص والميزات الأخرى التي ينبغي أن تتوافر في هذا النوع من نظم التشغيل، كنظام للملفات، بالإضافة إلى نظام إدارة الأقراص المسؤول عن توزيع الملفات على الأقراص وكذلك وجود أسلوب

متطور لجدولة وحدة المعالجة المركزية بما يضمن تنفيذ الأعمال المختلفة في آن واحد من دون الدخول في حالات إقفال متبادل ينتظر فيها أحد الأعمال انتهاء العمل الآخر والذي ينتظره بدوره إلى ما لا نهاية.

7. نظم الحواسب الشخصية

مع الانخفاض الكبير في تكلفة الحاسبات الصغرية مقارنةً مع نظم المنصات أصبح بالإمكان امتلاك المستخدم لنظامه الحاسوبي الخاص به، أُطلق على هذا النوع من النظم اسم نظم الحاسبات الشخصية.

تزامن ظهور هذا النوع من النظم مع تطور التجهيزات الحاسوبية تطوراً كبيراً على صعيد الشكل والأداء، فعلى سبيل المثال تغيرت معظم أساليب الدخل التي كانت سائدة لتتحول إلى طرائق استخدام للوحة المفاتيح والفأرة، كما تغيرت معظم أساليب الخرج لتصبح من خلال شاشات عرض أو طابعات صغيرة الحجم عالية الأداء.

عانت وحدات المعالجة المركزية في الفترة ما بين السبعينات والثمانينات من النقص في السمات اللازمة لحماية نظام التشغيل من برامج المستخدم، فلم تكن تلك النظم متعددة المستخدمين أو متعددة المهام؛ إلا أن أهداف تلك النظم تغيرت مع الزمن لتحقق أعلى قدر ممكن من ملاءمة المستخدم.

تُعد كلاً من شركتي Microsoft وApple من أوائل الرواد في تصميم نظم الحواسب الشخصية، بحيث ظهر نظام MS–DOS لتلحق به عدة إصدارات من نظام تشغيل Windows من شركة Microsoft، مع العلم أن شركة IBM قد قامت بتطوير نظام MS–DOS وتحويله إلى نظام OS/2 المتعدد المهام، كذلك طورت شركة Apple عدة إصدارات من نظام تشغيل Apple Macintosh.

تختلف استراتيجيات وقرارات التصميم المتعلقة بنظم تشغيل المنصات عن النظم الأخرى الأصغر منها، وخصوصاً فيما يتعلق بتوجه نظم المنصّات نحو مفهوم الفعّالية والاستفادة المثلى من قدرات المنصّات باهظة الثمن في حين تتوجه نظم التشغيل التي تعمل على الحواسب الصغرية نحو تحقيق ملاءمة المستخدم بالدرجة الأولى، إلا أن بعض مبادئ التصميم الأخرى لا تزال تُطبّق على كلتا النوعين (أو أصبح من الضروري أخذها بعين الاعتبار)، فمسألة حماية الملفات لم تكن ضرورية على الحاسبات الشخصية في البدايات، ولكن، ومع التطور التكنولوجي وتوسع الشبكات الحاسوبية وارتباطها مع شبكات أخرى من خلال الانترنت بحيث يمكن أن يتم النفاذ إلى الملفات عن بعد، أصبح لا بد من النظر إلى مفهوم الحماية كخاصية أساسية من خصائص نظم تشغيل الحواسب الشخصية.

8. النظم التفرعية

تُعْرَف نظم التشغيل التفرعية بأنها نظم حاسوبية تمتلك أكثر من معالج وحيد بحيث تتشارك المعالجات بمسرى الحاسب نفسه كما تتشارك أحياناً بالذاكرة والتجهيزات الطرفية. يُطلق على هذا النوع من نظم التشغيل أيضاً اسم النظم متعددة المعالجات.

تتميز نظم التشغيل التفرعية بخصائص متعددة يمكن تصنيفها وفق النقاط التالية:

1. **مردود أكبر:**

يمكن من خلال النظم الموزعة القيام بالعمل الذي كان يتم إنجازه بواسطة معالج وحيد باستخدام عدة معالجات، مما يسرّع وتيرة العمل ويؤدي إلى زيادة المردودية، مع العلم أن نسبة التسريع المتوقعة لا تساوي عدد المعالجات المضافة بل أقل منها، وذلك لنشوء عبء إضافي ينتج عن وجوب أن تتوافر طريقة لإدارة عملية التخاطب بين المعالجات بالإضافة إلى العبء الناتج عن تنافس المعالجات على الموارد المشتركة.

2. **توفير في التكاليف العتادية:**

تُمكننا النظم الموزعة من الاقتصاد في تكاليف التجهيزات العتادية في المؤسسة، بحيث يمكننا التشارك في الطرفيات أو وسائط التخزين أو وحدات التغذية عوضاً عن استخدام عدة تجهيزات.

3. **موثوقية أكبر:**

إن حدوث أي خلل أو إخفاق في أحد معالجات النظام التفرعي لا يؤدي بالضرورة إلى فشل النظام ككل بحيث تتحمل بقية المعالجات العبء المترتب على المعالج المعطَّل، وهو ما يُكسب هذا النوع من النظم خاصة مميزة يطلق عليها اسم خاصة التسامح مع الأخطاء.

أنواع نظم التشغيل التفرعية:

1. نموذج متعدد المعالجات متناظر:

وهو النموذج المتبع الأكثر شيوعاً في نظم التشغيل التفرعية المنتشرة، وفيه يشغِّل كل معالج نسخة متطابقة من نظام التشغيل مع النسخ الأخرى المشغَّلة على بقية المعالجات، بحيث تتصل هذه النسخ مع بعضها البعض حسب الحاجة.

2. نموذج متعدد المعالجات غير متناظر:

وفيه تُسند مهمات خاصة لكل معالج بحيث نميز نوعين من المعالجات، السيد والعبد، يسيطر المعالج السيد ويتحكم بالنظام ويتولى عملية جدولة الأعمال وتوزيعها على المعالجات الأخرى التي تنتظر بدورها التعليمات من المعالج السيد.

9. النظم الموزعة

يعتمد الاتجاه الحالي في تصميم نظم الحاسبات على مفهوم توزيع الحسابات بين عدة معالجات، يختلف هنا المفهوم المطروح عن مفهوم النظم التفرعية من مبدأ أن المعالجات لا تشترك بالذاكرة أو بالميقاتية إذ يمتلك كل معالج منها ذاكرته المحلية الخاصة، كما يتم التخاطب بين المعالجات من خلال أسلوب اتصال مناسب كشبكة محلية أو خطوط هاتف أو أية وسيلة أخرى. يُطلق على هذا النوع من النظم اسم النظم الموزعة.

يمكن أن تختلف المعالجات المكونة للنظام الموزع حجماً أو أداءً، فيمكن أن تكون عبارة عن معالجات صغرية أو محطات عمل أو حواسب شخصية أو حتى منصّات، كما يمكن الإشارة إليها بأسماء مختلفة كمواقع أو عقد أو حاسبات ...الخ وتختلف التسمية بحسب السياق الذي يتم فيه الإشارة إلى تلك المعالجات.

تختلف أسباب اعتماد النظم الموزعة وتتنوع، ومن تلك الأسباب:

1. التشارك بالموارد:

عندما يتكون النظام الموزع من عدة عقد ذات إمكانيات أداء مختلفة فإنه يمكن لمستخدم موجود في إحدى تلك العقد أن يستفيد من الموارد المتوافرة على العقد الأخرى، وبشكل عام يمكن أن تقدم النظم الموزعة آليات مختلفة من نمط التشارك بالملفات على مواقع متباعدة أو معالجة معطيات قاعدة معطيات موزّعة أو استخدام تجهيزات عتادية عن بعد كالطابعات مثلاً وغير ذلك...

2. تسريع المعالجة:

يمكن من خلال النظام الموزع أن يتم تقسيم الأعمال وتوزيع معالجتها على عدة مواقع في النظام الموزع ليتم تنفيذها بشكل متساير مما يساهم في تسريع الأداء، هذا من جهة، كما يمكننا من جهة أخرى أن نساهم في توزيع العبء المترتب على أحد المواقع في النظام الموزع لنقوم بتنفيذ بعض أعماله على موقع آخر أقل عبئاً، يطلق على هذه العملية اسم مشاركة الأعباء.

3. الموثوقية:

ينبغي أن تكون الموثوقية صفة هامة من صفات النظام الموزع، بحيث أن فشل أحد المواقع لا يؤثر على بقية المواقع المكونة للنظام.

10. نظم الزمن الحقيقي

تُعتبر نظم الزمن الحقيقي أحد أنواع نظم التشغيل التي تعمل ضمن ظروف ومتطلبات قاسية سواء كانت في التجهيزات أو سرعة المعالج أو في شدة تدفق المعطيات، كما ترتبط بها قيود زمنية محددة ومعرّفة جيداً، فالمعالجة فيها ينبغي أن تتم ضمن القيود المحددة وإلا فشل النظام، فنظام الزمن الحقيقي يؤدي وظيفته بالشكل الصحيح فقط عندما يعيد النتيجة الصحيحة ضمن القيود الزمنية المحددة له.

غالباً ما يتم إدخال المعطيات إلى نظام الزمن الحقيقي من خلال حساسات خاصة بحيث ينبغي على الحاسوب أن يقوم بتحليل المعطيات المدخلة.

تعد نظم التحكم بالتجارب العلمية أو نظم معالجة الصور الطبية أو التحكم الصناعي أو نظم تحريك الروبوتات أو نظم التسليح وغيرها، من نظم الزمن الحقيقي.

هناك نوعان أساسيان من نظم الزمن الحقيقي هما: نظام الزمن الحقيقي القاسي ونظام الزمن الحقيقي المرن.

1. نظام الزمن الحقيقي القاسي:

يستخدم هذا النوع من نظم التشغيل في إدارة المهمات الحساسة التي تتطلب دقة متناهية في العمل، وهو يتميز بانعدام نسبي في تأخيرات النظام ابتداءً من استحضار المعطيات وانتهاءً بإنهاء المتطلبات من تلك المعطيات، كما يتميز هذا النوع أيضاً بأنه لا يمتلك معظم سمات نظم التشغيل المتقدمة الأخرى كالذاكرة الافتراضية على سبيل المثال، وذلك لأن تلك السمات تسعى بشكل رئيسي إلى فصل المستخدم عن العتاديات مما يزيد من الشكوك المتعلقة بزمن الاستجابة المتوقع.

تتعارض نظم الزمن الحقيقي القاسي مع عمل نظم المشاركة بزمن المعالج، ولا يمكن المزج بين هذين النوعين من النظم معاً.

2. نظام الزمن الحقيقي المرن:

وفيه تكون مهمات الزمن الحقيقي الحساسة ذات أولوية أكبر على المهمات الأخرى، وتحتفظ المهمات الحساسة بأولويتها حتى ينتهي تنفيذها.

يمكن أن يتم مزج هذا النوع من النظم مع نظم تشغيل أخرى، إلا أن استخدامها في حالات التحكم الصناعي وتحريت الروبوتات محفوف بالمخاطر.

تختلف المجالات التي يمكن أن تكون فيها هذه النظم فعّالة ومفيدة، بحيث يمكن استخدامها في مجالات الوسائط المتعددة أو الحقيقة الافتراضية أو التجارب العلمية المختلفة كإكتشاف أعماق البحار على سبيل المثال.

تنتشر نظم الزمن الحقيقي المرن على العديد من نظم التشغيل الشائعة كبعض الإصدارات الرئيسية لنظام تشغيل Unix.

11. التمارين:

1. واحد مما يلي من المكونات الرئيسية للنظام الحاسوبي:

A. العتاديات

B. نظام التشغيل

C. التطبيقات

D. المستخدمين

E. جميع الإجابات صحيحة

2. يتولى نظام التشغيل مهمة الإشراف والمراقبة وتوفير البيئة الملائمة للتطبيقات والمستخدمين لكي ينفذوا أعمالهم ويستثمروا موارد الحاسوب وتطبيقاته:

A. صح

B. خطأ

3. يمكن النظر إلى نظام التشغيل:

A. كمخصص للموارد

B. نظام تحكم

C. نواة لتشغيل التطبيقات الحاسوبية

D. جميع الإجابات صحيحة

E. ولا واحد من الإجابات صحيح

4. يُعرَّف نظام التشغيل بأنه برنامج يدير عتاديات الحاسوب بحيث يوفر البرمجيات والتطبيقات الضرورية لتشغيل هذه العتاديات، كما يعمل كوسيط بين المستخدم والحاسوب يسمح للمستخدم باستثمار الحاسوب وتطبيقاته:

A. صح

B. خطأ

5. إذا كان الهدف من أنظمة التشغيل الحاسوبية خدمة المستخدم وتأمين احتياجاته بالطريقة الأسهل والأكثر بساطةً، يكون تصنيفها هنا بحسب الفعالية:

A. صح

B. خطأ

6. تعبر نظم المهمة الوحيدة عن نظم التشغيل البسيطة التي كان الحاسب فيها يقوم بتنفيذ تطبيق واحد فقط:

A. صح

B. خطأ

7. تعبر نظم المهمات المتعددة عن نظم التشغيل التي تستثمر الموارد على نحو يزيد من معدل استخدام وحدة المعالجة المركزية:

A. صح

B. خطأ

8. يتم في نظم المهمات المتعددة استغلال الموارد الأمثل من خلال:

A. قراءة مجموعة من الأعمال المخزنة على القرص ونقلها إلى الذاكرة لكي يتم تنفيذها معاً

B. اختيار أحد تلك الأعمال والبدء بتنفيذها

C. الإنتقال إلى العمل التالي عندما يتطلب العمل قيد التنفيذ إجراء عملية دخل / خرج معينة

D. جميع الإجابات صحيحة

9. يطلق على نظم المهمات المتعددة اسم نظم المشاركة بزمن المعالج:

A. صح

B. خطأ

10. ظهر نظام MS-DOS من شركة:

A. Apple

B. Microsoft

C. Sony

D. Samsung

11. نظم حاسوبية تمتلك أكثر من معالج وحيد بحيث تتشارك المعالجات بمسرى الحاسب نفسه كما تتشارك أحياناً بالذاكرة والتجهيزات الطرفية:

A. نظم التشغيل ذات المهمات المتعددة

B. نظم التشغيل ذات المهمة الوحيدة

C. نظم التشغيل التفرعية

D. النظم الموزعة

12. تتميز نظم التشغيل التفرعية بخصائص متعددة:

A. مردود أكبر

B. توفير في التكاليف العتادية

C. موثوقية أكبر

D. جميع الإجابات صحيحة

13. نموذج متعدد المعالجات غير المتناظر يعتبر من أنواع:

A. نظم التشغيل الموزعة

B. نظم الزمن الحقيقي

C. نظم التشغيل التفرعية

D. جميع الإجابات خاطئة

14. من أسباب اعتماد النظم الموزعة:

A. تسريع المعالجة

B. التشارك بالموارد

C. توفير في التكاليف العتادية

D. الإجابتين 1 و2

E. الإجابتين 2 و3

15. في نظام الزمن الحقيقي القاسي تكون مهمات الزمن الحقيقي الحساسة ذات أولوية أكبر على المهمات الأخرى، وتحتفظ المهمات الحساسة بأولويتها حتى ينتهي تنفيذها:

A. صح

B. خطأ

رقم التمرين	الإجابة الصحيحة
.1	(E)
.2	(A)
.3	(D)
.4	(A)
.5	(B)
.6	(A)
.7	(A)
.8	(D)
.9	(A)
.10	(B)
.11	(C)
.12	(D)
.13	(C)
.14	(D)
.15	(B)

بنية نظام التشغيل

الكلمات المفتاحية:

المقاطعة، التخبئة، الذاكرة الرئيسية، الذاكرة الخبيئة، الاتساق، نمط المراقب، نمط المستخدم، إدارة الإجرائية، استدعاءات النظام.

ملخص:

يركز هذا الفصل على البنى الأساسية لنظام التشغيل، وطرق حماية العتاديات، بالإضافة إلى مكونات وخدمات وبرامج نظام التشغيل.

أهداف تعليمية:

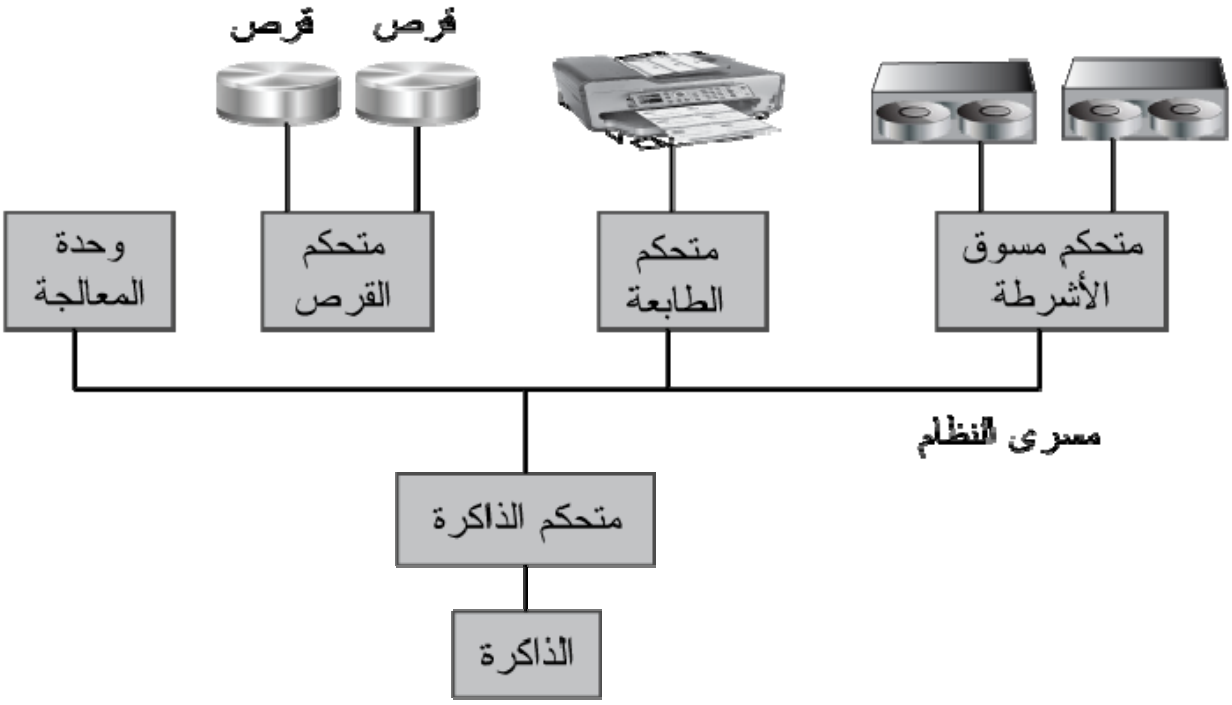
يهدف هذا الفصل إلى:

- التعرف على عمل النظام الحاسوبي.
- بنية الدخل/خرج، وبنية وهرمية التخزين.
- طرق حماية العتاديات (الدخل/خرج، الذاكرة، وحدة المعالجة).
- التعرف على مكونات نظام التشغيل (إدارة الإجرائيات، إدارة الذاكرة الرئيسية، إدارة الملفات).
- خدمات نظام التشغيل.
- استدعاءات النظام.
- برامج النظام.

1. عمل النظام الحاسوبي

يتألف النظام الحاسوبي من وحدة معالجة مركزية، وعدد من متحكمات التجهيزات، المتصلة فيما بينها بمسرى مشترك لتأمين اتصال هذه الوحدات بذاكرة مشتركة.

ينفذ الحاسب عند إقلاعه برنامج أولي يدعى برنامج الإقلاع حيث يقوم هذا البرنامج بتهيئة سجلات وحدة المعالجة، ومتحكمات التجهيزات، بالإضافة إلى الذاكرة، ثم يحدد هذا البرنامج مكان نواة نظام التشغيل ليقوم بتحميلها في الذاكرة، ومن ثم يبدأ نظام التشغيل بالعمل ويبقى في وضعية انتظار وقوع حدث ما.



يعلم نظام التشغيل بوقوع حدث من خلال المقاطعات وهي نوعان: مقاطعات عن طريق العتاديات، ومقاطعات عن طريق البرمجيات.

عندما تجري مقاطعة وحدة المعالجة فإنها تتوقف عن تنفيذ المهام الجارية وتنتقل مباشرةً لتنفيذ الرماز الخاص بالمقاطعة ومن ثم تعود لمتابعة عملها.

2. بنية الدخل/الخرج

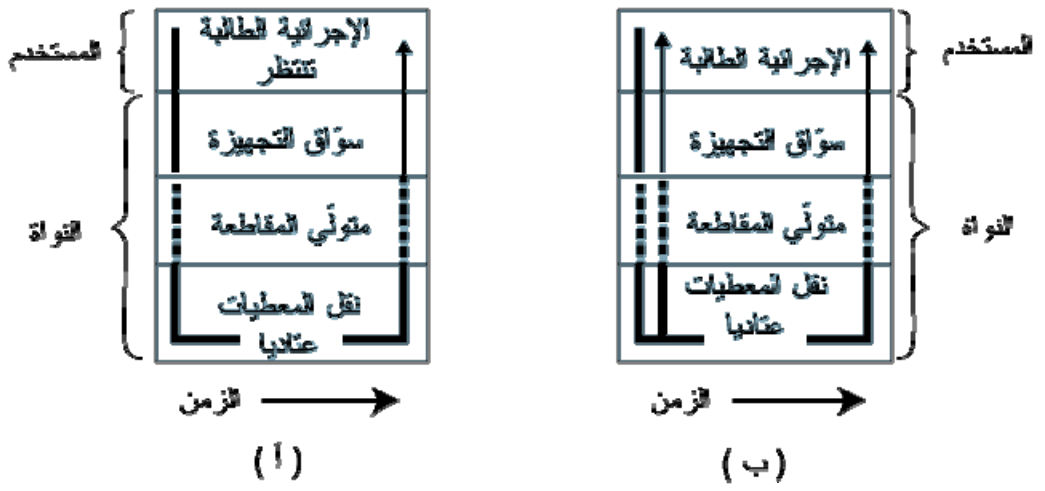
للبدء بعملية دخل/خرج، تُحمِّل وحدة المعالجة السجلات المناسبة ضمن متحكم الجهاز، يفحص متحكم الجهاز بدوره محتويات هذه السجلات لتحديد الفعل المطلوب القيام به.

يوجد نوعان من عمليات الدخل/خرج:

- الدخل/خرج المتزامن حيث يجري نقل التحكم إلى إجرائية المستخدم بعد انتهاء العملية
- الدخل/خرج غير المتزامن حيث يجري إرجاع التحكم إلى برنامج المستخدم دون انتظار انتهاء عملية الدخل/خرج.

يمكن انتظار اكتمال الدخل/خرج بطريقتين مختلفتين:

- إما بجعل وحدة المعالجة بلا عمل إلى أن تحدث المقاطعة التالية، وبهذا يمكن أن تجري عملية دخل/خرج واحدة على الأكثر
- أو إقلاع عملية الدخل/خرج ثم متابعة تنفيذ رماز نظام التشغيل (وبهذه الطريقة يمكن تنفيذ عدة عمليات دخل/خرج في نفس الوقت). نحتاج في هذه الطريقة إلى إمكانية تتبع عدة طلبات دخل/خرج في نفس الوقت، لذا، يستخدم نظام التشغيل جدولاً (يُدعى جدول حالة التجهيزات) يحتوي على قيد لكل جهاز دخل/خرج، يشير إلى نوع الجهاز وعنوانه وحالته.



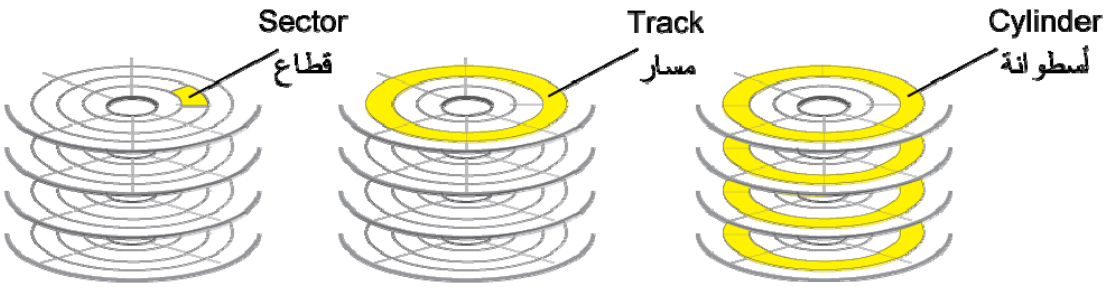
3. وحدات التخزين في النظم الحاسوبية

تُعتبر كلاً من الذاكرة الرئيسية (RAM) وسجلات المعالج، النمطين الوحيدين من وحدات التخزين اللذين يمكن لوحدة المعالجة النفاذ إليهما مباشرةً، لذا يجب نقل أية معطيات تحتاجها وحدة المعالجة، إلى الذاكرة الرئيسية (RAM) قبل بدء عملية المعالجة.

ويتعامل كل متحكم دخل/خرج مع سجلات تمكنه من الاحتفاظ بالأوامر والمعطيات، حيث تساعد مجموعة تعليمات خاصة في نقل المعطيات بين السجلات وذاكرة النظام، كما يتوفر للعديد من سجلات تجهيزات الدخل/خرج مساحات خاصة بها في الذاكرة تساعد في توفير نفاذ أكثر ملائمة لمتحكمات هذه التجهيزات، حيث يوضع مجال من عناوين الذاكرة جانباً ويجري ربطه بسجلات الجهاز.

بالإضافة لما سبق، توفر الأقراص المغناطيسية معظم وحدات التخزين الثانوي لنظم الحواسيب. حيث يُقسَّم سطح قرص التخزين منطقياً إلى مسارات دائرية والمسارات إلى قطاعات، وتسمى مجموعة المسارات الدائرية ذات البعد الواحد من مركز القرص بالأسطوانة.

قرص صلب



أقسام القرص الصلب

4. هرمية التخزين

تُعتبر التخبئة (caching) أحد المبادئ الهامة في عمل النظم الحاسوبية، حيث تحفظ المعلومات عادةً في نظام تخزين ويتم نسخها إلى نظام تخزين أسرع (الذاكرة الخبيئة) عند استخدامها.

عند الحاجة إلى معلومة معينة، يجري البحث عنها أولاً في الذاكرة الخبيئة، فإذا لم تتوفر فيها، يتم البحث عنها في أنظمة التخزين الأخرى.

وعندما نحصل على المعلومة من نظام التخزين الرئيسي يجري وضع نسخة منها في الذاكرة الخبيئة، لتسريع عملية الحصول عليها في المرة القادمة وخاصةً بوجود احتمال كبير لأن نحتاج إلى هذه المعلومات مرة ثانية بعد فاصل زمني قصير.

يمكن أن تكون حركة المعلومات بين مستويات هرمية التخزين إما حركة صريحة يكون نظام التشغيل مسؤولاً عنها، أو حركة ضمنية تكون العتاديات مسؤولة عنها، وذلك تبعاً لتصميم العتاديات وبرمجيات نظام التشغيل. فعلى سبيل المثال، تكون عملية نقل المعطيات من وحدات الدخل/خرج إلى سجلات المعالج، وظيفة عتادية، بينما تكون عملية نقل المعطيات من القرص إلى الذاكرة وظيفة يتحكم بها نظام التشغيل.

5. اتساق الذاكرة الخبيئة

يمكن أن تظهر، في بنية تخزين هرمية، نفس المعطيات في مستويات مختلفة من نظام التخزين.

في بيئة حاسوبية تنفذ فيها إجرائية واحدة فقط في وقت واحد، لا يسبب هذا الترتيب أية صعوبات، ولكن في بيئة متعددة المهمات حيث يتم تبديل وحدة المعالجة بين الإجرائيات يجب التأكد من حصول جميع الإجرائيات على أحدث قيمة لـA. وتصبح الحالة أكثر تعقيداً في بيئة متعددة المعالجات، حيث تحوي كل وحدة معالجة إضافة إلى السجلات الداخلية، ذاكرة خبيئة محلية خاصة بها.

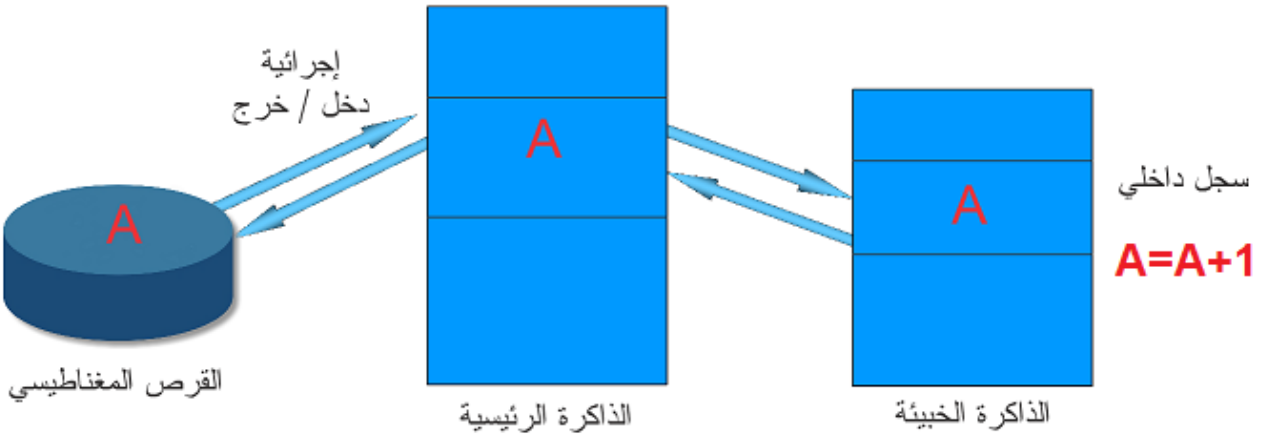
يُدعى هذا الوضع اتساق الذاكرة الخبيئة والذي يُعتبر مسألة عتادية تُعالج في مستوى أدنى من مستوى نظام التشغيل.

مثال:

نفرض أن العدد الصحيح A موجود في الملف B ويتوجب زيادة قيمته بمقدار 1، ولنفرض أن الملف B موجود على القرص المغناطيسي. يجري تنفيذ الزيادة كما يلي:

1. نسخ كتلة القرص التي تحوي A إلى الذاكرة الرئيسية بإجرائية دخل/خرج.
2. نسخ A إلى الذاكرة الخبيئة وإلى سجل داخلي.
3. يصبح لـ A قيماً مختلفة في نظم التخزين المختلفة بعد إجراء الزيادة في السجل الداخلي.
4. توحيد قيمة A عند نقل القيمة الجديدة لـA من السجل الداخلي إلى القرص المغناطيسي.

في بيئة متعددة المهمات و/أو متعددة المعالجات، حيث يجري استخدام وحدة المعالجة أو عدة وحدات معالجة من قبل مجموعة إجراءات، يجب التأكد من حصول جميع الإجراءات على أحدث نسخة من المعطيات.



6. حماية العتاديات

من أجل حماية نظام التشغيل والبرامج الأخرى والمعطيات المرتبطة بها من أي برنامج لا يعمل عملاً سليماً، تعتمد العديد من نظم التشغيل على توفير دعم عتادي يسمح بالتفريق بين أنماط تنفيذ متنوعة.

نحتاج إلى نمطي عمل مستقلين:

- نمط المستخدم
- نمط المراقب أو النظام

حيث يضاف بت يسمى بت النمط إلى عتاديات الحاسب للدلالة على النمط (0 نمط مراقب، 1 نمط مستخدم)، وبهذا نستطيع تفريق مهمة تنفذ لمصلحة نظام التشغيل عن مهمة تنفذ لمصلحة المستخدم.

تقوم فكرة حماية العتاديات على تعريف بعض تعليمات الآلة القابلة لإحداث ضرر على أنها تعليمات ذات امتياز لا يمكن تنفيذها إلا بنمط المراقب، وتعتبر تعليمة غير شرعية إذا جرت محاولة تنفيذها بنمط المستخدم، حيث يتم عندها التقاطها من قبل نظام التشغيل.

تُعامل التعليمة ذات الامتياز على أنها مقاطعة برمجية، حيث يتم الانتقال لتنفيذ الرماز الخاص بهذه المقاطعة بعد أن يوضع نمط المراقب في بت النمط.

7. حماية الدخل/الخرج

يمكن أن يسبب عمل برنامج المستخدم عطلاً في عمل النظام، وذلك بتشغيله لتعليمات دخل/خرج غير شرعية، أو بنفاذه إلى أماكن في الذاكرة ضمن نظام التشغيل نفسه، أو برفضه التخلي عن وحدة المعالجة.

لمنع المستخدمين من إجراء عمليات دخل/خرج غير شرعية، تُعرَّف جميع تعليمات الدخل/خرج على أنها تعليمات ذات امتياز، لذلك لا يستطيع المستخدمون إجراء تعليمات دخل/خرج مباشرةً، إنما يتوجب عليهم المرور بنظام التشغيل واستخدام استدعاءات محددة خاصة بالنظام.

8. حماية الذاكرة ووحدة المعالجة

من أجل حماية الذاكرة من التعديل من قبل برنامج المستخدم، ولحماية برامج المستخدمين من بعضها البعض، يتم تخصيص مساحة ذاكرة مستقلة لكل برنامج على حده، حيث يتم تحديد مجال العناوين الشرعية التي يمكن أن ينفذ إليها البرنامج، وبحيث تتم حماية الذاكرة الواقعة خارج هذا الفضاء.

تجري عملية الحماية من خلال استخدام سجلين: يُدعى الأول سجل القاعدة (base)، ويُدعى الثاني سجل الحد (limit)، يحتوي السجل الأول على أصغر عنوان شرعي في الذاكرة، بينما يحتوي السجل الثاني على حجم المجال، وتعتمد إجرائية الحماية على مقارنة كل عنوان يتم توليده ضمن نمط عمل المستخدم بمحتوى السجلين، بحيث تُعامَل أية محاولة يقوم بها البرنامج للوصول إلى ذاكرة متوضعة خارج المجال المتاح له، على أنها خطأ.

وتجري حماية وحدة المعالجة من برامج المستخدمين –بهدف تلافي دخولها في حلقة استخدام غير منتهية– من خلال استخدام مؤقت يتم إعداده لمقاطعة المعالج بعد مدة محددة. يجري استخدام نفس المؤقت من أجل المشاركة الزمنية وتوزيع زمن المُعالج في نظم التشارك بالزمن.

9. إدارة الإجرائيات (مكونات نظام التشغيل)

يتم بناء نظام ضخم ومعقد كنظام التشغيل من خلال تجزئته إلى قطع أصغر حجماً، بحيث يكون لكل جزء مداخل ومخارج ووظائف مُعرَّفة بشكل واضح، وتُعتبر إدارة الإجرائيات من أهم مكونات نظم التشغيل.

تُعرَّف الإجرائية بأنها عبارة عن برنامج في طور التنفيذ، إذ تُعتبر الإجرائية كياناً فعّالاً بينما يُعتبر البرنامج كيان غير فعال، وتحتاج الإجرائية لكي تنفذ مهمتها إلى موارد معينة نذكر منها:

- زمن وحدة المعالجة
- الذاكرة
- الملفات
- تجهيزات الدخل/الخرج.

ويتم حجز هذه الموارد عند إقلاع الإجرائية أو أثناء تنفيذها.

يجري تنفيذ الإجرائية بشكل تتابعي من قبل وحدة المعالجة (تعليمة تلو الأخرى)، بحيث يجري تنفيذ تعليمة واحدة من أي إجرائية في وقت معين.

لإدارة الإجرائيات، يمتلك نظام التشغيل الإمكانات التالية:

- إنشاء إجرائيات النظام والمستخدم وحذفها
- إيقاف تنفيذ الإجرائيات واستئناف تنفيذها
- توفير آليات مزامنة الإجرائيات
- توفير آليات الاتصال بين الإجرائيات
- توفير آليات التعامل مع استخدام الإجرائيات لموارد مشتركة

10. إدارة الذاكرة الرئيسية (مكونات نظام التشغيل)

تُعرَّف الذاكرة الرئيسية بأنها مخزن معطيات مشترك بين وحدة المعالجة وتجهيزات الدخل/الخرج، وهي جهاز التخزين الكبير الوحيد التي يمكن لوحدة المعالجة أن تعنونه وتنفذ إليه مباشرةً.

لإدارة الذاكرة، يمتلك نظام التشغيل الإمكانات التالية:

- تعقب أجزاء الذاكرة المستخدمة حالياً، وتعقب هوية مستخدميها
- اختيار الإجرائية التي يجب تحميلها في الذاكرة، حين يصبح فضاء الذاكرة متاحاً
- تقسيم فضاء الذاكرة إلى حصص، وإعادة تقسيمه بحسب الحاجة

11. إدارة الملفات (مكونات نظام التشغيل)

تُعتبر عملية حجب خصوصيات التجهيزات العتادية عن المستخدم، أحد مهام نظام التشغيل، حيث يكون نظام التشغيل مسؤولاً عن إدارة تجهيزات الدخل/الخرج وإدارة تجهيزات التخزين المقابلة لها.

لإدارة الملفات، يمتلك نظام التشغيل عادةً الإمكانات التالية:

- إنشاء الملفات وحذفها
- إنشاء المجلدات وحذفها
- العمليات الأساسية لمعالجة الملفات والمجلدات
- النسخ الاحتياطي للملفات على وسائط تخزين مستقرة

ولإدارة الأقراص الصلبة، يمتلك نظام التشغيل عادةً الإمكانات التالية:

- إدارة فضاء الذاكرة
- تحصيص عملية التخزين
- جدولة القرص

12. خدمات نظام التشغيل

يوفر نظام التشغيل البيئة اللازمة لتنفيذ البرامج، ويوفر عدة خدمات للبرامج ولمستخدميها:

- تنفيذ البرامج: يجب أن يكون نظام التشغيل قادراً على تحميل برنامج معين في الذاكرة، وتنفيذ ذلك البرنامج.
- عمليات الدخل/الخرج: يجب أن يكون نظام التشغيل قادراً على التعامل مع تجهيزات الدخل/خرج، إذ يمكن أن يحتاج البرنامج الجاري تنفيذه إلى دخل/خرج (سواء كان ملف أو جهاز دخل/خرج).
- معالجة نظام الملفات: يجب أن يكون نظام التشغيل قادراً على التعامل مع نظم الملفات، إذ تحتاج البرامج إلى قراءة وكتابة ملفات، بالإضافة إلى إنشائها وحذفها.
- الاتصالات: يجب أن يكون نظام التشغيل قادراً على توفير بيئة تواصل بين الإجراءات، إذ تحتاج الإجراءات إلى تبادل معلومات مع إجراءات أخرى، وبحيث يمكن أن تكون الإجراءات على نفس الحاسب أو على نظم حاسوبية مختلفة.
- اكتشاف الأخطاء: يجب أن يكون نظام التشغيل قادراً على اكتشاف الأخطاء المحتملة، سواء كانت أخطاء وحدة المعالجة، أو أخطاء عتاديات الذاكرة، أو أخطاء تجهيزات الدخل/الخرج.
- تحصيص الموارد: يجب أن يكون نظام التشغيل قادراً على توزيع الموارد بحصص عادلة عندما يدخل عدة مستخدمين إلى النظام في نفس الوقت، أو عندما يجري تنفيذ عدة أعمال في نفس الوقت.
- المحاسبة: يجب أن يكون نظام التشغيل قادراً على تعقب المستخدمين الذين يستخدمون موارد الحاسوب، وقادراً على تحديد كمية هذه الموارد وأنواعها.
- الحماية: يجب أن يكون نظام التشغيل قادراً على منع تداخل الإجراءات بعضها مع بعض وعلى منع تداخلها مع نظام التشغيل نفسه.

13. استدعاءات النظام

توفر استدعاءات النظام واجهة تواصل بين الإجراءات ونظام التشغيل. تجري استدعاءات النظام بعدة أساليب ويختلف نوع المعاملات وعددها تبعاً للاستدعاء، حيث توجد ثلاث طرق عامة لتمرير المعاملات:

- عبر سجلات شرط أن يكون عدد المعاملات أقل أو يساوي من عدد السجلات.
- عبر تخزين المعاملات في كتلة أو جدول في الذاكرة، ومن ثم تمرير عنوان هذه الكتلة كمعامل من خلال سجل.
- عبر وضع المعاملات في مكدس.

تُقسم استدعاءات النظام إلى خمس فئات رئيسية:

1. التحكم بالإجراءات وبالإجرائيات: الإنهاء والإيقاف، التحميل والتنفيذ، الإنشاء والإنهاء، الحصول على الواصفات، انتظار مدة معينة، انتظار حدث، إشارة إلى حدث، حجز الذاكرة وتحريرها.
2. إدارة الملفات: الإنشاء والحذف، الفتح والإغلاق، القراءة والكتابة، الحصول على الواصفات، تحديد الواصفات.
3. إدارة التجهيزات: طلب الولوج، القراءة والكتابة، الحصول على الواصفات، تحديد الواصفات.
4. صيانة المعلومات: الحصول على الوقت أو التاريخ، تحديد الوقت أوالتاريخ، الحصول على معطيات النظام، تعيين معطيات النظام، الحصول على واصفات إجراء أو ملف أو جهاز، تحديد واصفات إجراء أو ملف أو جهاز.
5. الاتصالات: إنشاء الاتصال، حذف الاتصال، إرسال رسائل، تسلم رسائل، نقل معلومات الحالة، إلحاق تجهيزات بعيدة أو إلغاء إلحاقها.

14. برامج النظام

إذا عدنا إلى الهرمية المنطقية للحاسب، نجد أن العتاديات تتوضع في المستوى الأدنى، يأتي بعدها نظام التشغيل، ثم برامج النظام، وأخيراً البرامج التطبيقية.

توفر برامج النظام بيئة مناسبة لبناء/تطوير البرامج وتنفيذها. إذ تمثل بعض هذه البرامج واجهات تعامل المستخدم مع استدعاءات النظام.

يمكن تقسيم برامج النظام إلى الفئات التالية:

- برامج إدارة الملفات: وهي البرامج التي تتيح إنشاء الملفات والمجلدات، وحذفها، ونسخها، وإعادة تسميتها، وطباعتها.
- برامج معلومات الحالة: وهي البرامج التي توفر لبرامج أخرى معلومات متنوعة مثل الوقت، أو التاريخ، أو حجم الذاكرة، أو المساحة المتاحة على القرص، أو عدد المستخدمين، أو أية معلومات أخرى.
- برامج تعديل الملفات: وهي البرامج التي تتيح التعامل مع الملفات المخزنة في القرص.
- برامج دعم لغات برمجة: وهي البرامج التي تتيح للمستخدم ضمن نظام التشغيل، ترجمة وتفسير لغات البرمجة الشائعة.
- برامج تحميل البرامج التطبيقية وتنفيذها.
- برامج الاتصالات: وهي البرامج التي توفر آلية بناء ارتباطات افتراضية بين الإجراءات والمستخدمين ونظم الحواسيب المختلفة.

15. التمارين:

1. يقوم هذا البرنامج بتهيئة سجلات وحدة المعالجة، ومتحكمات التجهيزات، بالإضافة إلى الذاكرة:
   A. نظام التشغيل
   B. التطبيقات
   C. برنامج الإقلاع
   D. جميع الإجابات خاطئة

2. يوجد نوعان من عمليات الدخل/خرج: الدخل/خرج المتزامن وغير المتزامن:
   A. صح
   B. خطأ

3. تُعتبر كلاً من الذاكرة الرئيسية (RAM) وسجلات المعالج، النمطين الوحيدين من وحدات التخزين اللذين يمكن لوحدة المعالجة النفاذ إليهما مباشرةً:
   A. صح
   B. خطأ

4. توفر الأقراص المغناطيسية معظم وحدات التخزين الثانوي لنظم الحواسيب:
   A. صح
   B. خطأ

5. تحفظ المعلومات عادةً في نظام تخزين ويتم نسخها إلى نظام تخزين أسرع (الذاكرة الخبيئة) عند استخدامها:
   A. التخبئة
   B. وحدة المعالجة
   C. الأقراص المغناطيسية
   D. جميع الإجابات صحيحة

6. يمكن أن يسبب عمل برنامج المستخدم عطلاً في عمل النظام من خلال:
   A. تشغيله لتعليمات دخل/خرج غير شرعية
   B. نفاذه إلى أماكن في الذاكرة ضمن نظام التشغيل نفسه
   C. رفضه التخلي عن وحدة المعالجة
   D. جميع الإجابات صحيحة

7. تجري عملية حماية الذاكرة من خلال استخدام سجلين: يُدعى الأول سجل القاعدة (base)، ويُدعى الثاني سجل الحد (limit):

A. صح

B. خطأ

8. تُعتبر الإجرائية كياناً فعّالاً بينما يُعتبر البرنامج كيان غير فعال:

A. صح

B. خطأ

9. تحتاج الإجرائية لكي تنفذ مهمتها إلى موارد معينة:

A. زمن وحدة المعالجة

B. الذاكرة

C. الملفات

D. جميع الإجابات صحيحة

10. لإدارة الإجرائيات، يمتلك نظام التشغيل الإمكانات التالية:

A. إنشاء إجرائيات النظام والمستخدم وحذفها.

B. إيقاف تنفيذ الإجرائيات واستئناف تنفيذها.

C. توفير آليات مزامنة الإجرائيات.

D. جميع الإجابات صحيحة

11. تُعرَّف الذاكرة الرئيسية بأنها مخزن معطيات مشترك بين وحدة المعالجة وتجهيزات الدخل/الخرج:

A. صح

B. خطأ

12. لإدارة الذاكرة، يمتلك نظام التشغيل الإمكانات التالية:

A. تعقب أجزاء الذاكرة المستخدمة حالياً

B. اختيار الإجرائية التي يجب تحميلها في الذاكرة

C. تقسيم فضاء الذاكرة إلى حصص

D. جميع الإجابات صحيحة

13. لإدارة الملفات، يمتلك نظام التشغيل عادةً الإمكانات التالية:

A. إنشاء الملفات وحذفها.

B. إنشاء المجلدات وحذفها.

C. العمليات الأساسية لمعالجة الملفات والمجلدات.

D. النسخ الاحتياطي للملفات على وسائط تخزين مستقرة.

E. جميع الإجابات صحيحة

14. يوفر نظام التشغيل البيئة اللازمة لتنفيذ البرامج، ويوفر عدة خدمات للبرامج ولمستخدميها:

A. عمليات الدخل/الخرج

B. المحاسبة

C. الحماية

D. تحصيص الموارد

E. جميع الإجابات صحيحة

15. تُقسم استدعاءات النظام إلى:

A. إدارة الملفات

B. الاتصالات

C. صيانة المعلومات

D. جميع الإجابات صحيحة

16. يمكن تقسيم برامج النظام إلى:

A. برامج الاتصالات

B. برامج تعديل الملفات

C. برامج صيانة المعلومات

D. الإجابتين 1 و2

E. الإجابتين 2 و3

رقم التمرين	الإجابة الصحيحة
.1	(C)
.2	(A)
.3	(A)
.4	(A)
.5	(A)
.6	(D)
.7	(A)
.8	(A)
.9	(D)
.10	(D)
.11	(A)
.12	(D)
.13	(E)
.14	(E)
.15	(D)
.16	(D)

أنظمة تشغيل الحواسيب الشخصية

ISSN: 2617-989X

الكلمات المفتاحية:

مفسر الأوامر، نواة نظام التشغيل، السجل، واجهة المستخدم، سواقة التجهيزة، نظام FAT، نظام NTFS، الBIOS، أرقام طلبات المقاطعة، قناة الوصول المباشر للذاكرة، المتحكم بالمقاطعات، تسجيلة الإقلاع الرئيسية، برنامج الإقلاع الرئيسي.

ملخص:

يركز هذا الفصل على مفهوم نظام التشغيل، مهمات نظام التشغيل، موارد النظام، بالإضافة إلى آلية إقلاع النظام.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- التعرف على نظام التشغيل.
- مهمات نظام التشغيل (واجهة المستخدم، إدارة المجلدات والملفات، إدارة التطبيقات، إدارة العتاد).
- موارد النظام (طلبات المقاطعة، عناوين الذاكرة، عناوين الدخل/الخرج، وقناة الوصول المباشر للذاكرة).
- إقلاع النظام.
- ايجاد وتحميل نظام التشغيل.

1. أنظمة تشغيل الحواسيب الشخصية، مثال: نظام تشغيل windows

يتألف نظام الحاسب من عتاد وبرمجيات، حيث يمثل العتاد الفيزيائي الجزء المرئي من نظام الحاسب، بينما تمثل البرمجيات ذكاء النظام، والذي يُمكِّن المكونات العتادية من العمل.

تعرفنا في الدروس السابقة على المفهوم النظري لنظام التشغيل ومكوناته وأنوعه، وسنركز في هذا الدرس، على أنظمة الحواسيب الشخصية الموجودة (على نظام windows بالتحديد)، وكيفية عملها وتحكمها بالعديد من الأجهزة العتادية، بالإضافة إلى التعرف على الواجهة التي يؤمنها النظام من أجل التحكم بهذه العتاديات.

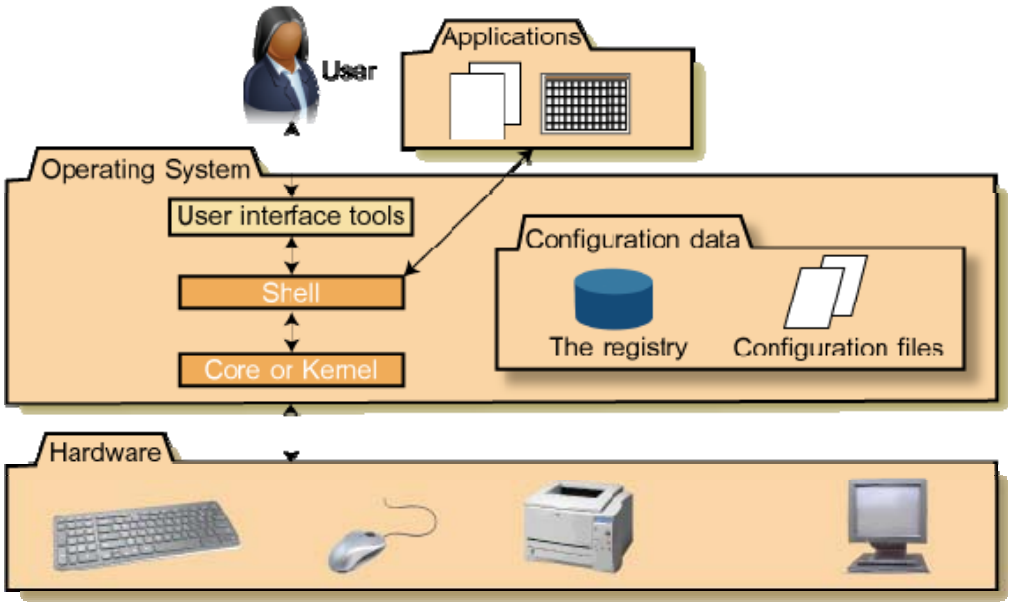
2. مقدمة عن نظام التشغيل

يمكننا تعريف نظام التشغيل على أنه مجموعة من البرمجيات التي تتحكم بالحاسب: فهي تُدير العتاديات، وتُشغِّل البرامج، وتُؤمن واجهة للمستخدمين، كما تكون مسؤولةً عن إدارة الملفات في النظام. بشكل عام يلعب نظام التشغيل دور الطبقة الوسطى بين التطبيقات والعتاد.

يتألف نظام التشغيل من مكونين داخليين أساسيين:

- مفسر الأوامر (shell): الجزء الذي يرتبط بالمستخدم والتطبيقات، حيث يقدم (أوامر، قوائم، أو واجهة من الأيقونات) للمستخدم للتعامل مع نظام التشغيل.
- النواة (kernel): يربط هذا الجزء بين مفسر الأوامر والعتاد، حيث يحوي على استدعاءات النظام الأساسية التي تتعامل مع العتاد.

يحتاج نظام التشغيل إلى حفظ معلومات معينة عن العتاد، أو عن التطبيقات في النظام، لذلك تستخدم أنظمة Windows لهذا الغرض قاعدة معطيات تدعى السجل (registry).



Operating System	نظام تشغيل
Applications	تطبيقات
User	مستخدم
User Interface Tools	واجهات المستخدم
Configuration Data	معطيات إعداد
The Registry	السجل
Configuration File	ملفات إعداد
Shell	واجهة الاستخدام
Core or Kernel	النواة
Hardware	العتاديات

3. مهمات نظام التشغيل

يقدم نظام التشغيل المهمات التالية:

- واجهة المستخدم: يطلب المستخدم من خلال هذه الواجهة، تنفيذ الأوامر من النظام.
- إدارة المجلدات والملفات: يدير نظام التشغيل المجلدات والملفات في النظام، فهو المسؤول عن تخزين وإدارة الملفات والمجلدات على أجهزة الخزن الدائمة.
- إدارة التطبيقات: يكون نظام التشغيل مسؤولاً عن إدارة كافة البرمجيات على الحاسب، بما في ذلك من تحميل وتشغيل للتطبيقات.
- إدارة العتاد: يتم الاتصال بين نظام التشغيل والعتاديات من خلال سواقات خاصة بالتجهيزات، فنظام التشغيل هو المسؤول عن الاتصال والتعامل مع العتاد.

4. واجهة المستخدم

يقوم المستخدم من خلال واجهة المستخدم، بطلب تنفيذ الأوامر من النظام.

يمكن أن تكون لواجهة المستخدم أحد الأشكال التالية:

- واجهة الأوامر: يتم من خلال هذه الواجهة، طباعة الأوامر اللازمة لنظام التشغيل من أجل تنفيذ عمليات معينة.
- مثال في نظام windows: اضغط start ثم اختر all programs، وبعدها accessories، ومن ثم قم بالضغط على command prompt.
- واجهة القوائم: يمكنك تنفيذ عملية معينة، من خلال قائمة من الخيارات المتاحة.

- واجهة الأيقونات: يتم تنفيذ عملية معينة من خلال الضغط على الأيقونة المناسبة، وهي أكثر الواجهات استخداماً.

5. إدارة المجلدات والملفات

يدير نظام التشغيل المجلدات والملفات في النظام، فهو المسؤول عن تخزين وإدارة الملفات والمجلدات على أجهزة الخزن الدائمة.

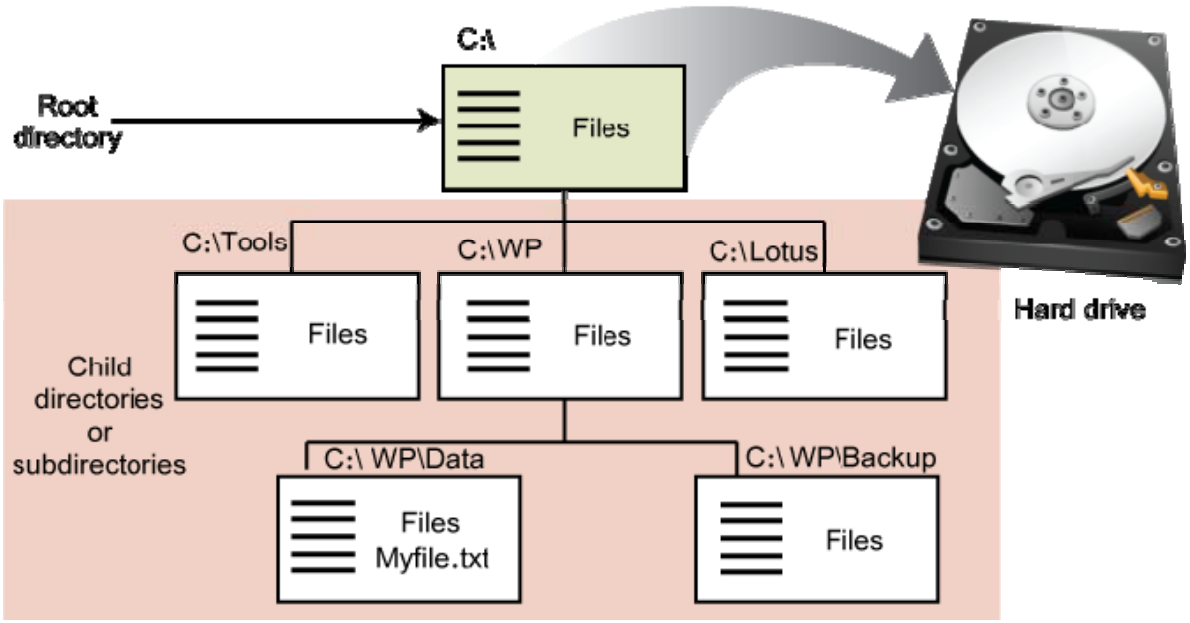
تدعى طريقة تنظيم الملفات في النظام بنظام الملفات.

يستخدم نظام windows عدد من نظم الملفات أهمها:

نظام FAT (file allocation table): يعتمد هذا النظام على جدول يحتوى على عناوين ومواقع الملفات على القرص.

نظام NTFS (new technology file system): يؤمن هذا النظام حماية أكبر.

مهما كان نظام الملفات المستخدم، يدير نظام التشغيل الملفات والمجلدات من خلال بنية شجرية، تبدأ بالمجلد الجذر، وتتوزع ضمنه المجلدات والملفات الأخرى بشكل شجري أيضاً.



Hard Drive	سواقة القرص الصلب
Root Directory	المجلد الأساسي (الجذر)
Child Directories or SubDirectories	المجلدات الفرعية
Files	ملفات

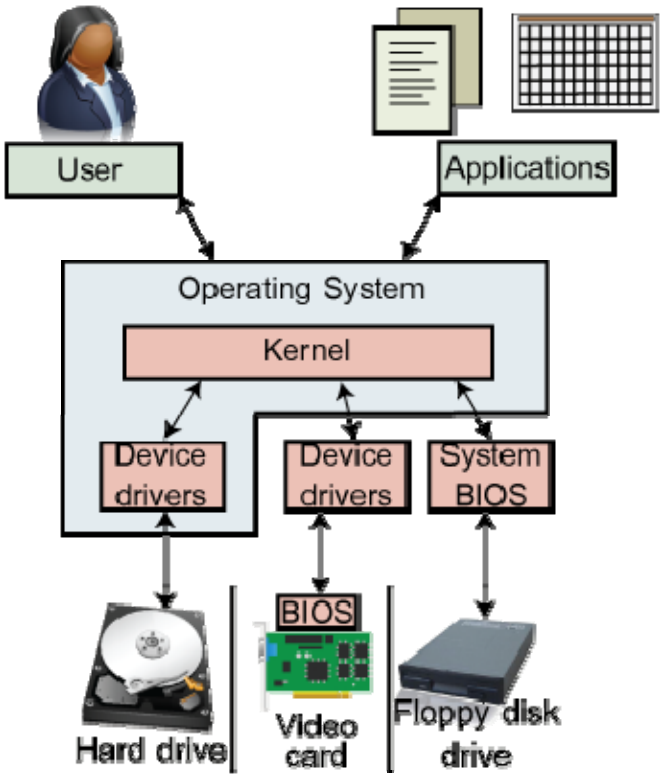
6. إدارة التطبيقات

يدير نظام التشغيل كافة البرمجيات الموجودة على الحاسب، بما في ذلك تحميل وتشغيل البرامج.
لا يستطيع التطبيق أن يقوم بتحميل نفسه في الذاكرة من أجل التنفيذ، إنما يحتاج لنظام التشغيل ليقوم بذلك.
كما أنه يعتمد على النظام في الوصول إلى العتاديات والتجهيزات المختلفة، وفي إدارة المعطيات الخاصة به في الذاكرة وعلى القرص الصلب أيضاً.

7. إدارة العتاد

على الرغم من أن نظام التشغيل هو المسؤول عن إدارة العتاد، إلا أنه لا يتصل مباشرةً بالعتاديات، إنما من خلال سواقات التجهيزات أو الـ BIOS.

- سواقات التجهيزات: عبارة عن برامج صغيرة مخزنة على القرص الصلب، وهي التي تقدم لنظام التشغيل آلية التعامل مع تجهيزة عتادية معينة (طابعة، كرت شبكة،...).
- الـBIOS: ذاكرة موضوعة على اللوحة الأم مباشرةً، وهي تحوي على ثلاث أنواع من البرامج، برامج للتحكم بتجهيزات الدخل/الخرج، برامج للتحكم بإقلاع الحاسب، بالإضافة إلى برامج لتغيير المعلومات الخاصة بإعدادات النظام.



Operating System	نظام التشغيل
Kernel	النواة
Device Drivers	سواقات التجهيزات
Floppy Disk Drive	سواقة أقراص مرنة
Hard Drive	سواقة قرص صلب
Video Card	بطاقة فيديوية
BIOS	وحدة دخل خرج أساسية
Application	تطبيق
User	مستخدم

8. موارد النظام

تُعتبر موارد النظام أدوات تُستخدم من قبل البرمجيات أو العتاد للاتصال فيما بينها، وهنالك أربع أنواع:

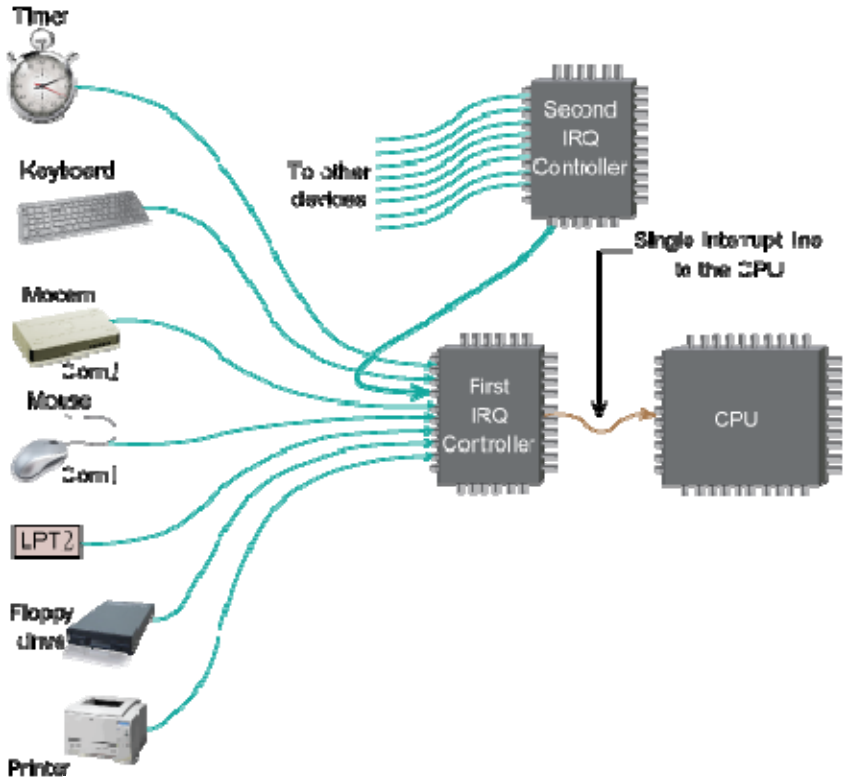
1. أرقام طلبات المقاطعة (IRQ): وهي خطوط على اللوحة الأم، يجري استخدامها من قبل التجهيزة لإعلام وحدة المعالجة بأنها تريد إجراء عملية معينة.
2. عناوين الدخل/الخرج: وهي عناوين يجري اسنادها للتجهيزات، بحيث تستخدمها البرمجيات، عند طلب أمر معين من التجهيزة.
3. عناوين الذاكرة: وهي عناوين يجري اسنادها للذاكرة الفيزيائية، بحيث تستخدمها البرمجيات، من أجل الوصول والتعامل مع الذاكرة.
4. قناة الوصول المباشر للذاكرة: يجري من خلال هذه القناة نقل المعطيات بين التجهيزة والذاكرة، وذلك من دون تدخل وحدة المعالجة.

9. أرقام طلبات المقاطعة

تكون أرقام طلبات المقاطعة (IRQ) عبارة عن خطوط على اللوحة الأم، يتم استخدامها من قبل التجهيزة، لإعلام وحدة المعالجة بأنها تريد إجراء عملية معينة.

يُسند لكل خط من خطوط المقاطعة رقم (من 0 و حتى 15).

لا تجري إدارة طلبات المقاطعات القادمة من التجهيزات من قبل وحدة المعالجة، إنما يدير المتحكم بالمقاطعات هذه الطلبات، ومن ثم يُعلِم وحدة المعالجة بتلك الطلبات.



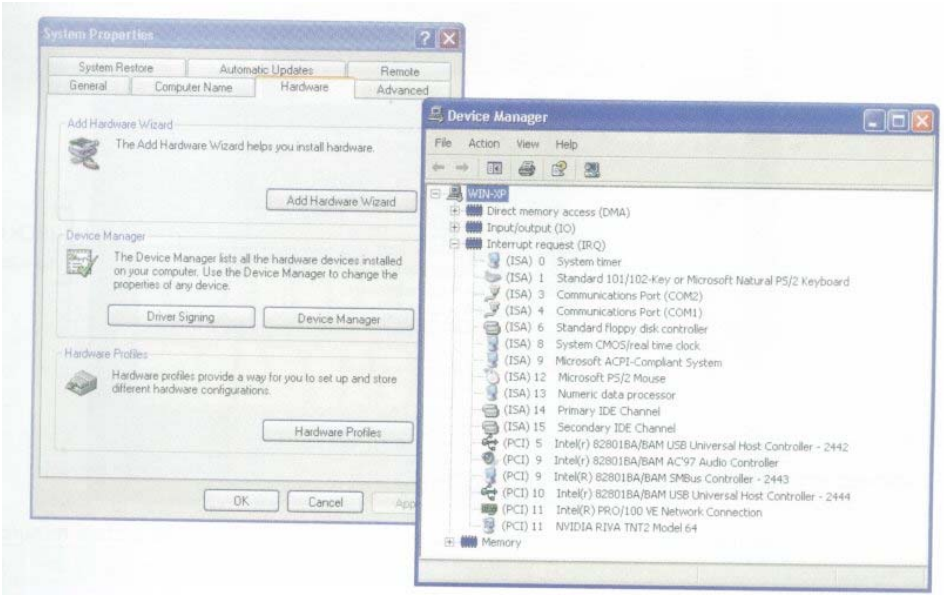
Second IRQ Controller	المتحكم الأول بالمقاطعة
First IRQ Controller	المتحكم الثاني بالمقاطعة
CPU	وحدة المعالجة المركزية
Modem	مودم
Keyboard	لوحة مفاتيح
Floppy drive	سواقة أقراص مرنة
Printer	طابعة
To Other Devices	إلى تجهيزات أخرى
Single Interrupt to the CPU	مقاطعة وحيدة باتجاه وحدة المعالجة
Timer	

يمكن إضافة طلبات مقاطعة جديدة إلى قائمة الطلبات، حيث يتم التعامل مع هذه المقاطعات الإضافية عن طريق متحكم ثاني بالمقاطعات.

مثال: في windows:

من أجل رؤية أرقام المقاطعات المُسندة لكل تجهيزة:

اضغط على start، ثم ضغط يميني على my computer، اختر خصائص properties، ومن ثم اختر صفحة hardware.

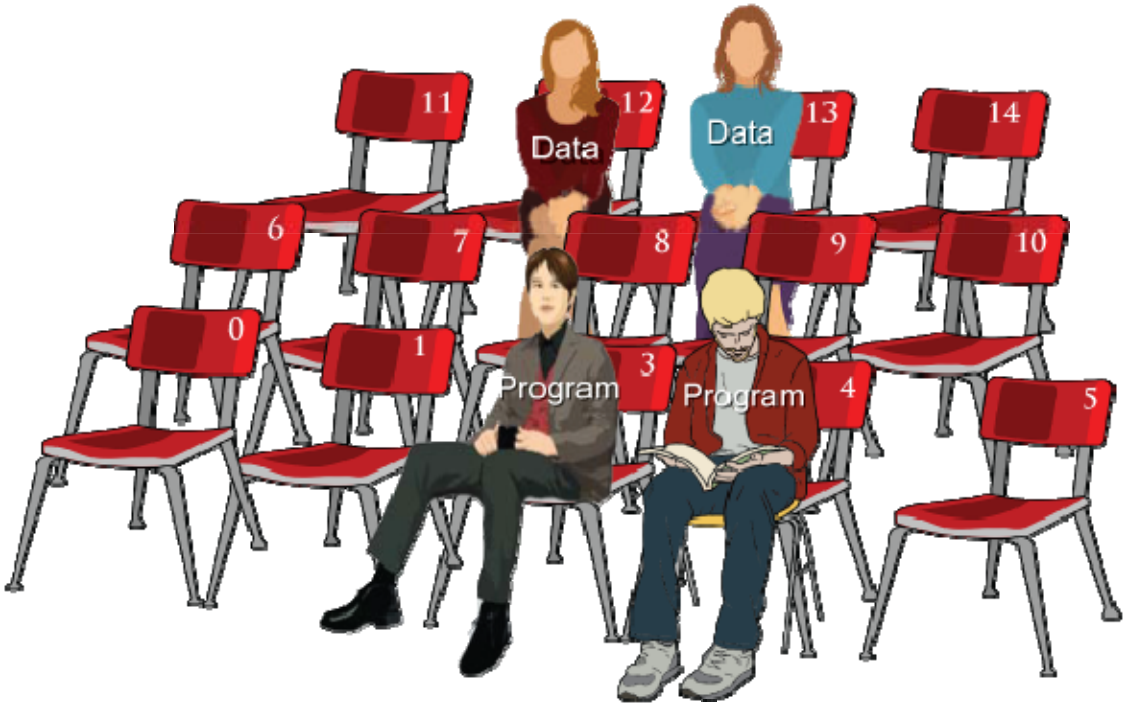


10. عناوين الذاكرة

يجري استخدام عناوين الذاكرة من قبل البرمجيات من أجل الوصول والتعامل مع الذاكرة.

يتعامل نظام التشغيل مع الذاكرة على أنها قائمة طويلة من الخلايا، يجري تخزين معطيات أو تعليمات فيها.

يُسند إلى كل خلية من الخلايا عدد يدعى عنوان، حيث يجري الوصول والتعامل مع هذه الخلية عن طريق هذا العنوان.



11. عناوين الدخل/الخرج

يجري استخدام عناوين التجهيزات (عناوين الدخل/الخرج) من قبل البرمجيات عند طلب أمر معين من التجهيزة.
يستخدم النظام عناوين الدخل/الخرج للوصول إلى التجهيزات، بطريقة مماثلة لاستخدام عناوين الذاكرة للوصول إلى الذاكرة.
تنتقل العناوين عن طريق مسرى على اللوحة الأم يدعى مسرى الاتصال، حيث يحمل هذا المسرى تارةً عناوين ذاكرة و تارةً أخرى عناوين الدخل/الخرج.

12. قناة الوصول المباشر للذاكرة

يجري عن طريق قناة الوصول المباشر للذاكرة نقل المعطيات بين التجهيزة والذاكرة، وذلك من دون تدخل وحدة المعالجة. فهذه القناة هي طريقة مختصرة تُمكن تجهيزة الدخل/الخرج من إرسال المعطيات مباشرة إلى الذاكرة.
تَستخدم بعض التجهيزات هذه القناة (مثل الطابعة)، بينما لا تَستخدمها تجهيزات أخرى (مثل الماوس).
هنالك عدة قنوات للوصول المباشر للذاكرة مرقّمة من 0 وحتى 7، تَستخدم كل تجهيزة قناة معينة.
يمكن تغيير إعدادات هذه القناة عن طريق الـBIOS.

13. إقلاع النظام

يُقسَّم إقلاع النظام إلى أربع خطوات أساسية:

- الخطوة1: تبدأ هذه الخطوة بإقلاع برنامج الـBIOS، ومن ثم يقوم هذا البرنامج بالاستعلام عن العتاديات الموجودة في النظام، ومقارنة المعلومات الخاصة بها مع المعلومات المخزنة سابقاً.
- الخطوة2: يقوم برنامج الـBIOS بالبحث عن نظام التشغيل من أجل تحميله، وغالباً يتم تحميل نظام التشغيل من السواقة المنطقية C على القرص الصلب.
- الخطوة3: يقوم نظام التشغيل بدوره بالاستعلام عن الموارد الموجودة (الذاكرة، مساحة التخزين المتوفرة،...)، ومن ثم يقوم النظام بتحميل البرمجيات الخاصة للتعامل مع الأجهزة (الماوس، الطابعة،...).
- الخطوة4: يُحمِّل نظام التشغيل في بعض الأحيان برنامج معين من أجل تنفيذه، حيث يعتبر تنفيذ هذا البرنامج من ضمن مرحلة الإقلاع.

14. إيجاد وتحميل نظام التشغيل

يقوم برنامج الBIOS بالبحث عن نظام التشغيل من أجل تحميله، وغالباً يتم تحميل نظام التشغيل من السواقة المنطقية C على القرص الصلب.

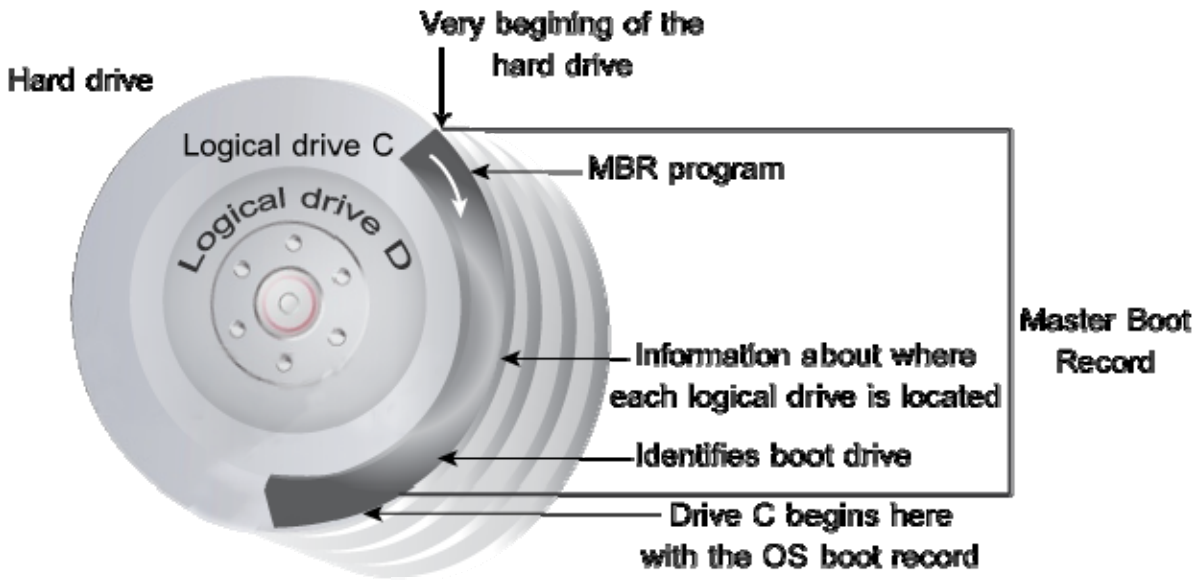
تُخزَن المعلومات اللازمة لإقلاع نظام التشغيل (والمخزنة على القرص الصلب) في تسجيلة الإقلاع الرئيسية (MBR): قطاع صغير (512 بايت) في بداية القرص الصلب.

يحوي MBR على عنصرين أساسيين:

- برنامج الإقلاع الرئيسي: البرنامج الأساسي لإيجاد بداية نظام التشغيل أثناء الإقلاع.
- جدول الأجزاء: يحوي هذا الجدول معلومات عن الأجزاء المنطقية على القرص الصلب (عدد الأجزاء المنطقية، بداية كل جزء، الجزء الفعّال،...).

يتم تحميل برنامج الإقلاع الرئيسي في الذاكرة، حيث يقوم هذا البرنامج بدوره بتحميل نظام التشغيل إلى الذاكرة.

في نظام windows 9x/Me برنامج الإقلاع Io.sys، أما في windows NT/2000/XP فهو Ntldr.



Very Beginning of the hard drive	**بداية القرص الصلب**
MBR Program	**برنامج MBR**
Information about where each logical drive is located	**معلومات عن توضع كل سواقة قرص صلب**
Drive C begins here with the OS boot Record	**تبدأ السواقة C من هنا مع تسجيلة إقلاع نظام التشغيل**
Drive D, Drive E	**السواقة D، السواقة E**
Identify boot drive	**تحديد سواقات الإقلاع**
Master Boot Record	**تسجيلة الإقلاع الرئيسية**
Hard Drive	**قرص صلب**

15. أسئلة

1. سمِّ أحد موارد النظام التي لا يحتاج إليها كرت الفيديو.
2. سمِّ أحد التجهيزات التي يتم التحكم بها عن طريق الـBIOS.
3. هل يتم التحكم بالماوس عن طريق متحكم بالتجهيزة أم عن طريق الـBIOS؟
4. إذا كان الحاسب يفحص إمكانية الإقلاع من القرص الصلب، ثم من القرص المرن، كيف تُغيِّر تتالي الاختبار لكي يفحص الحاسب القرص المرن أولاً؟

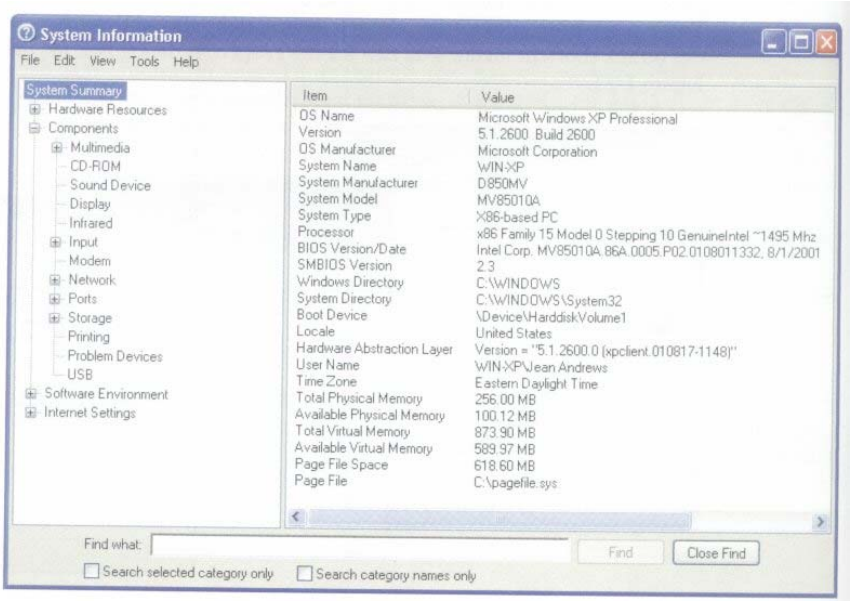
16. تمرين عملي: (الحصول على معلومات النظام)

الحصول على معلومات عن نظام windows:

اضغط على start، ثم run وبعدها اطبع Msinfo32.exe ثم ok، وعندها تظهر معلومات النظام.

أجب عن الأسئلة التالية:

- ما هو نظام التشغيل وما هو إصداره؟
- ما هي سرعة وحدة المعالجة؟
- ما هي نسخة الـBIOS ومكان التصنيع؟
- ما هي كمية الذاكرة RAM الخاصة بكرت الفيديو؟ اشرح كيف حصلت على هذه المعلومة؟
- ما هو اسم ملف التجهيزة الذي يتحكم بالمنفذ التفرعي؟ والملف الذي يتحكم بالمنفذ التسلسلي؟
- كيف تُستخدم IRQ رقم 10 ورقم 4، في النظام؟
- ما هي قنوات DMA المستخدمة في النظام؟ وكيف يتم استخدامها؟



17. تمرين عملي: (التحكم بالتجهيزات)

الحصول على معلومات التجهيزات في نظام windows:

اضغط بالزر اليميني على my computer، ثم properties وبعدها اختار صفحة device manager.

أجب عن الأسئلة التالية:

- هل يحوي النظام على كرت شبكة؟ وفي حال وجوده ما هي معلوماته؟
- ما هي الإعدادات الثلاث التي يمكن تغييرها في device manager؟
- ما هي التجهيزات الموجودة والتي تَعرّف عليها النظام؟

18. تمرين عملي: (إعدادات الـBIOS)

للحصول على معلومات التجهيزات في نظام windows:

اقرأ المعلومات التي تظهر على الشاشة عند بداية إقلاع الجهاز، لتعرف ما عليك فعله للدخول إلى BIOS، مثلاً اضغط على الزر Del للدخول إلى Setup.

أجب عن الأسئلة التالية:

- ما هو الزر الذي استخدمه للدخول؟
- ما هي معلومات الـBIOS المستخدمة؟
- ادخل إلى النافذة التي تحوي معلومات عن المنافذ التسلسلية؟ ما هو اسم هذه النافذة؟
- ما هي عناوين الدخل/الخرج و IRQ التي يَستخدمها المنفذ التسلسلي الأول؟
- ما هي عناوين الدخل/الخرج و IRQ التي يَستخدمها المنفذ التفرعي الأول؟

19. تمرين عملي: (إقلاع النظام)

أعد إقلاع الجهاز وأجب عن الأسئلة التالية:

- انزع وصلة لوحة المفاتيح وأعد الإقلاع، ما هو الفرق في عملية الإقلاع؟
- أعد وصل لوحة المفاتيح، وانزع وصلة الشاشة وأعد الإقلاع، ثم أعد توصيل الشاشة بعد انتهاء الإقلاع، هل لاحظ الحاسب أن الشاشة غير موصولة؟
- ضع قرص مرن لا يحوي برنامج الإقلاع في السواقة المرنة، وسجل ما يحدث أثناء إقلاع النظام، هل أقلع الحاسب بالشكل الطبيعي؟

20. التمارين:

1. يحتاج نظام التشغيل إلى حفظ معلومات معينة عن العتاد، أو عن التطبيقات في النظام:
   A. صح
   B. خطأ

2. يقدم نظام التشغيل:
   A. واجهة المستخدم
   B. إدارة التطبيقات
   C. إدارة العتاد
   D. جميع الإجابات صحيحة

3. يتم الاتصال بين نظام التشغيل والعتاديات من خلال سواقات خاصة بالتجهيزات:
   A. صح
   B. خطأ

4. يمكن أن تكون لواجهة المستخدم أحد الأشكال التالية:
   A. واجهة القوائم
   B. واجهة الأيقونات
   C. واجهة الأوامر
   D. جميع الإجابات صحيحة
   E. جميع الإجابات خاطئة

5. يستخدم نظام windows عدد من نظم الملفات أهمها FAT و NTFS:
   A. صح
   B. خطأ

6. يستطيع التطبيق أن يقوم بتحميل نفسه في الذاكرة من أجل التنفيذ:
   A. صح
   B. خطأ

7. الـBIOS:

A. ذاكرة موضوعة على اللوحة الأم مباشرةً

B. نظام ملفات

C. نظام تطبيقات

D. واجهة المستخدم

8. من موارد النظام:

A. عناوين الذاكرة

B. أرقام طلبات المقاطعة

C. قناة الوصول المباشر للذاكرة

D. جميع الإجابات صحيحة

9. يستخدم النظام عناوين الدخل/الخرج للوصول إلى التجهيزات، بطريقة مغايرة لاستخدام عناوين الذاكرة للوصول إلى الذاكرة:

A. صح

B. خطأ

رقم التمرين	الإجابة الصحيحة
.1	(A)
.2	(D)
.3	(A)
.4	(D)
.5	(A)
.6	(B)
.7	(A)
.8	(D)
.9	(B)

مسالك التنفيذ

الكلمات المفتاحية:

مسلك التنفيذ، مسلك على مستوى النواة، مسلك على مستوى المستخدم، النموذج كثير إلى واحد، النموذج كثير إلى كثير، النموذج واحد إلى واحد، إيقاف مسلك التنفيذ، المسلك المستهدف، مجمَّع المسالك، المكتبة Pthreads.

ملخص:

يركز هذا الفصل على مفهوم مسالك التنفيذ وفوائدها، وأنواع مسالك التنفيذ، مقارنة بينها وبين الإجراءات، بالإضافة إلى نماذج مسالك التنفيذ، ومجمَّع مسالك التنفيذ.

أهداف تعليمية:

يهدف هذا الفصل إلى:

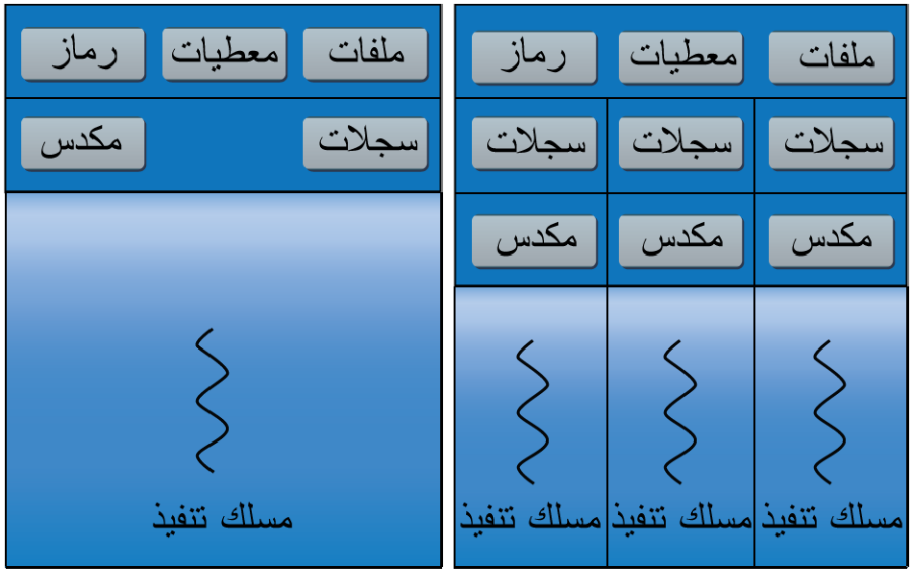
- التعرف على مسالك التنفيذ.
- فوائد مسالك التنفيذ.
- مقارنة بين الإجراءات ومسالك التنفيذ.
- أنواع مسالك التنفيذ (مسالك على مستوى النواة، ومسالك على مستوى المستخدم).
- نماذج تعدد مسالك التنفيذ (كثير إلى واحد، واحد إلى واحد، كثير إلى كثير).
- مجمَّع مسالك التنفيذ.
- مثال عن استخدام المكتبة Pthreads في إنشاء مسالك التنفيذ.

1. مسالك التنفيذ

يُعرَّف مسلك التنفيذ بأنه الوحدة الأساسية في استخدام وحدة المعالجة، ويتألف من محدد هوية (ID)، وعدّاد برنامج، بالإضافة إلى مجموعة سجلات ومكدس.

تتشارك مسالك التنفيذ المنتمية إلى نفس الإجراء، في مقطع الرماز ومقطع المعطيات الخاصين بها، بالإضافة إلى موارد نظام التشغيل كالملفات.

يملك الإجراء التقليدي مسلك تنفيذ واحد، ولكن عندما يحوي الإجراء على عدة مسالك تنفيذ، فإنه يستطيع القيام بأكثر من مهمة في وقت واحد.



إجراء مع عدة مسالك تنفيذ　　إجراء مع مسلك تنفيذ وحيد

2. فوائد مسالك التنفيذ

يمكن تقسيم فوائد البرمجة باستخدام مسالك التنفيذ إلى أربع فوائد أساسية:

1. سرعة الإستجابة: يستمر التطبيق متعدد المسالك في التنفيذ، حتى لو كان جزءاً منه ينتظر أو يؤدي عملية طويلة، وهذا يرفع سرعة الاستجابة.
2. التشارك في الموارد: تتشارك مسالك التنفيذ تلقائياً في الذاكرة، وفي الموارد الخاصة بالإجراء الذي تنتمي إليه، وهذا ما يسمح للتطبيق بإمتلاك عدة مسالك تقع كلها ضمن فضاء عنونة واحد.
3. الاقتصاد: إن حجز الذاكرة والموارد من أجل إنشاء الإجراءات، هو عملية مكلفة، وبما أن مسالك التنفيذ تتشارك في موارد الإجراء الذي تنتمي إليه، يُعتبر إنشاء المسالك وتبديل سياقها، بدلاً من إنشاء الإجراءات، أمراً اقتصادياً أكثر.

4. استخدام بنية متعددة المعالجات: تتضاعف فوائد تعدد مسالك التنفيذ في بنية متعددة المعالجات، حيث يمكن تنفيذ كل مسلك على التوازي على معالج مختلف. بينما لا يمكن تنفيذ إجراء أحادي المسلك إلا على وحدة معالجة واحدة.

3. مقارنة بين الإجراءات ومسالك التنفيذ

الإجراءات	مسالك التنفيذ
يحتاج توليد إجراء جديد إلى موارد كثيرة.	لا يحتاج توليد مسلك تنفيذ جديد إلى موارد كثيرة.
تبديل السياق بين الإجراءات مكلف.	تبديل السياق بين المسالك أقل تكلفة منه في حالة الإجراءات.
الاتصال بين الإجراءات صعب.	الاتصال بين مسالك التنفيذ سهل.
يوجد حماية للإجراءات وعزل فيما بينها.	لا يوجد حماية للمسالك أو عزل فيما بينها.

4. مسالك على مستوى النواة

هنالك نوعان لمسالك التنفيذ: مسالك على مستوى النواة، ومسالك على مستوى المستخدم.

المسالك على مستوى النواة:

تكون بنية معطيات المسالك معرفة ضمن منطقة الذاكرة الخاصة بنواة نظام التشغيل، وتكون عملية إنشاء وإدارة المسالك من مهام نظام التشغيل (استدعاءات نظام). يحصل كل مسلك على فترة تنفيذ خاصة به. وإذا توقف أحد المسالك بسبب استدعاء نظام ما، فإن ذلك لا يؤثر على بقية المسالك.

5. مسالك على مستوى المستخدم

هنالك نوعان لمسالك التنفيذ: مسالك على مستوى النواة، ومسالك على مستوى المستخدم.

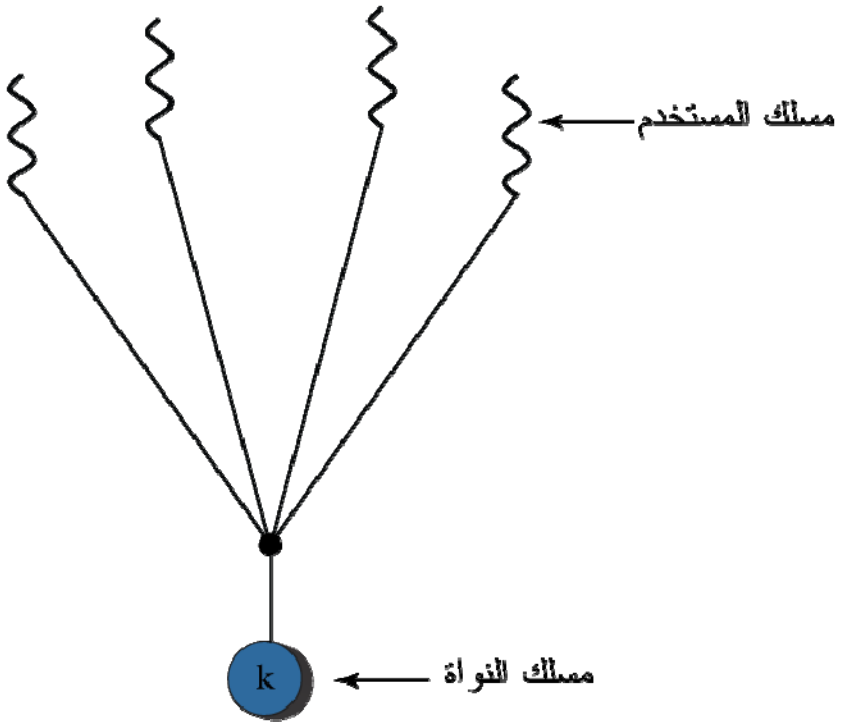
المسالك على مستوى المستخدم:

- تُعرَّف بنية المعطيات الخاصة بالمسالك ضمن منطقة الذاكرة الخاصة بإجراء المستخدم.
- تكون عملية إنشاء وإدارة المسالك من مهام الإجراء نفسه (باستخدام مكتبة من الإجراءات العادية) بدون استخدام استدعاءات نظام، وبالتالي يتم ذلك بسرعة أكبر.
- تتقاسم المسالك فيما بينها الفترة الزمنية المخصصة لتنفيذ الإجراء.
- إذا توقف أحد المسالك بسبب استدعاء نظام، فإن الإجراء بالكامل يتوقف، وتتوقف بالتالي جميع المسالك.

6. نماذج تعدد مسالك التنفيذ (النموذج كثير إلى واحد)

النموذج الأول من نماذج تعدد المسالك، هو النموذج كثير إلى واحد:

تجري في هذا النموذج مقابلة أكثر من مسلك على مستوى المستخدم، بمسلك واحد على مستوى النواة، حيث تجري إدارة المسالك في فضاء المستخدم ولذلك فهي فعالة، إلا أن الإجراء سيتوقف كله إذا قام مسلك واحد بطلب أحد استدعاءات النظام التي تسبب الانتظار. كما يمكن لمسلك واحد فقط النفاذ إلى النواة في وقت واحد، وبذلك لا تستطيع عدة مسالك أن تنفذ على التوازي حتى ولو تعددت المعالجات.

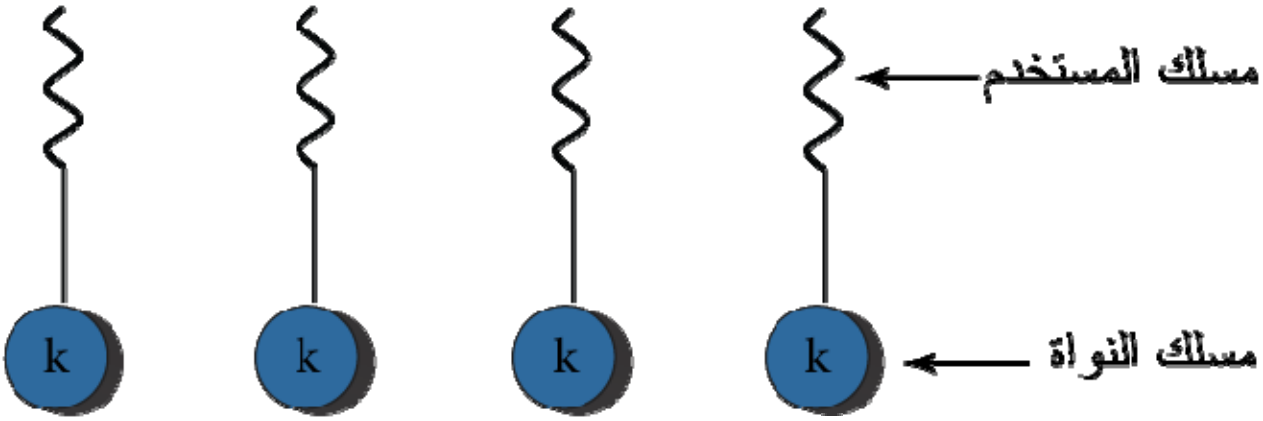


7. نماذج تعدد مسالك التنفيذ (النموذج واحد إلى واحد)

في النموذج واحد إلى واحد، تجري مقابلة كل مسلك مستخدم بمسلك نواة.

يوفر هذا النموذج قدرة أكبر على التوازي من النموذج كثير إلى واحد، وذلك من خلال السماح بتنفيذ مسلك ثاني حين يستدعي مسلك ما أحد استدعاءات النظام التي تسبب الانتظار. كما يسمح بتنفيذ عدة مسالك على التوازي في حال وجود عدة معالجات.

تكمن سيئة هذا النموذج في أن إنشاء مسلك مستخدم جديد يتطلب إنشاء مسلك النواة الموافق له، مما يؤدي إلى عبء إضافي على التطبيق ويؤثر سلباً على أدائه.



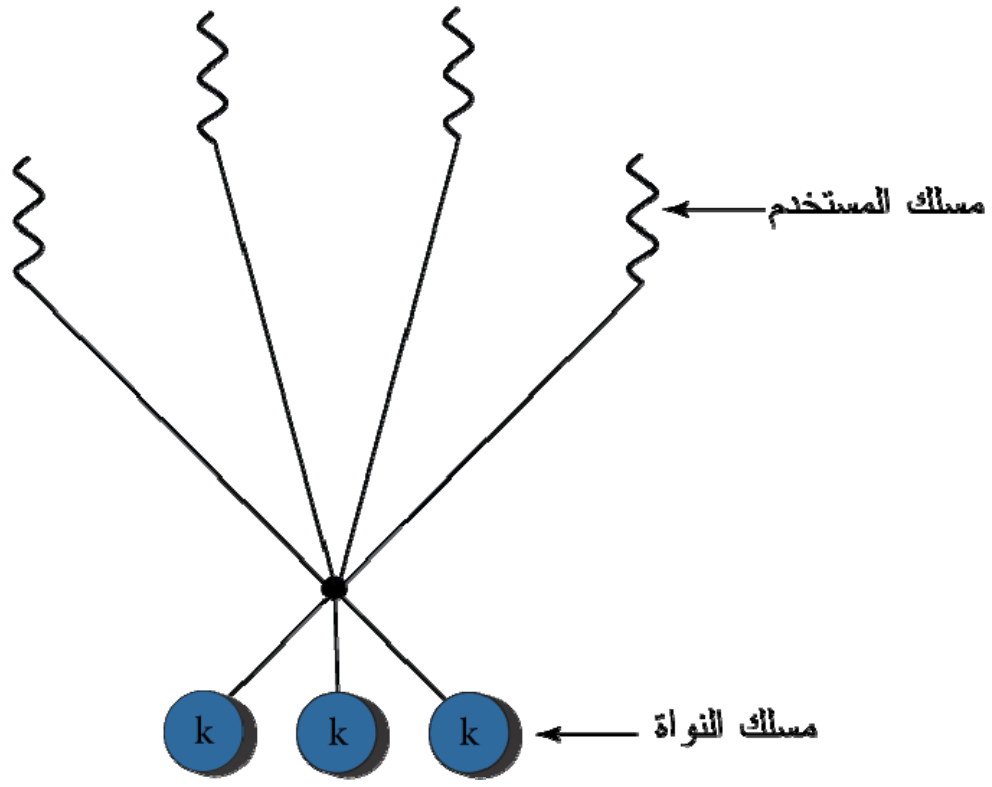
8. نماذج تعدد مسالك التنفيذ (النموذج كثير إلى كثير)

في النموذج كثير إلى كثير تجري مقابلة عدة مسالك على مستوى المستخدم مع عدة مسالك على مستوى النواة، بحيث يكون عدد مسلك النواة أقل أو يساوي عدد مسلك المستخدم.

يتيح النموذج "كثير إلى واحد للمطوِّر إنشاء العدد الذي يرغب به من مسالك المستخدم، لكنه لا يسمح له بتحقيق توازي حقيقي في التنفيذ، لأن النواة لا تجدول إلا مسلك واحد في وقت واحد.

ويتيح النموذج واحد إلى واحد توازي أكبر، لكن يتطلب من المطوِّر انتباهاً أكبر على عدد المسالك التي يجري إنشاؤها.

أما نموذج "كثير إلى كثير" فإنه لا يعاني من أي من هذه المساوئ: إذ يستطيع المطورون إنشاء العدد الضروري من المسالك، بحيث يمكن تنفيذ مسالك النواة الموافقة لها، على التوازي عند وجود عدة معالجات. كذلك فعند قيام مسلك ما باستدعاء أحد استدعاءات النظام التي تسبب الانتظار، فإن النواة تستطيع جدولة مسلك آخر للتنفيذ.



9. إيقاف مسالك التنفيذ

يمكن إيقاف مسلك التنفيذ، أي إنهاء مهمته قبل أن يُنهي تنفيذه، فعلى سبيل المثال، حين تبحث عدة مسالك على التوازي في قاعدة معطيات، ويعيد أحد هذه المسالك نتيجة ما، يمكن إيقاف المسالك الأخرى. كذلك يجري إيقاف مسلك التحميل عندما يضغط المستخدم على زر في متصفح الوِب من أجل إيقاف تحميل الصفحة (حيث يجري تحميل صفحة الوِب غالباً بواسطة مسلك تنفيذ مستقل).

يمكن إيقاف مسلك التنفيذ في حالتين:

إيقاف لا متزامن: يقوم أحد المسالك بإنهاء المسلك المستهدف مباشرةً.

إيقاف مؤجل: يستطيع المسلك المستهدف أن يتحقق دورياً، هل يجب عليه أن ينتهي أم لا؟ وبذلك يمكن أن يُنهي تنفيذه نهاية نظامية.

10. مجمَّع مسالك التنفيذ

تتلخص الفكرة العامة لمجمَّع مسالك التنفيذ في إنشاء عدة مسالك تنفيذ عند بدء الإجرائية، ومن ثم وضع هذه المسالك في مجمَّع، وإبقائها جاهزة للعمل.

عندما يستلم النظام طلب ما، يُسنِد مهمة الطلب إلى أحد المسالك المتاحة في المجمَّع، وحين ينتهي مسلك التنفيذ من مهمته، يعود إلى المجمَّع لانتظار مهمات جديدة لحين إسنادها إليه.

في حال عدم احتواء المجمَّع على مسالك متاحة، ينتظر النظام إنهاء أحد المسالك للمهمة الموكلة إليه.

أهم فوائد مجمَّع مسالك التنفيذ:

- خدمة أسرع للطلبات من حالة مسلك تنفيذ واحد، ومن حالة انتظار مسلك تنفيذ جديد.
- تحديد عدد المسالك الموجودة في وقت واحد، وهي ناحية مهمة جداً خصوصاً في النظم التي لا تدعم إلا عدد محدد من مسالك التنفيذ.

11. مكتبة التعامل مع مسالك التنفيذ Pthreads

- تعتبر Pthreads مكتبة برمجية تساعد على إنشاء مسالك التنفيذ ومزامنتها، وهي عبارة عن مكتبة على مستوى المستخدم، وجميع المسالك التي يتم تعريفها من خلال هذه المكتبة، هي عبارة عن مسالك على مستوى المستخدم.
- تتضمن جميع برامج Pthreads ملف الترويسة pthread.h، كما يمثل المتحول "pthread_t tid" محدد الهوية الخاصة بالمسلك المراد إنشاؤه، كما يمثل "pthread_attr_t attr" التصريح عن مجموعة واصفات المسلك (حجم المكدس، ومعلومات الجدولة).

```
#include <pthread.h>
#include <stdio.h>

int sum; /* this data is shared by the thread(s) */
void *runner(void *param); / the thread */

main (int argc, char *argv[])
{
    pthread_t tid; /* the thread identifier */
    pthread_attr_t attr; /* set of thread attributes */
    if (argc != 2) {
        fprintf(stderr, "usage: a.out <integer value>\n");
        exit() ;
    }
    if (atoi(argv[1]) < 0) {
        fprintf(stderr, "%d must be >=0\n",atoi(argv[1]));
        exit() ;
    }
    /* get the default attributes */
    pthread_attr_init(&attr);
    /* create the thread */
    pthread_create(&tid,&attr,runner,argv[1]);
    /* now wait for the thread to exit */
    pthread_join(tid,NULL);
    fprintf("sum = %d\n",sum);
}
```

مثال عن استخدام هذه المكتبة:

يمكن كتابة برنامج يقوم بإنشاء مسلك مستقل مهمته حساب ناتج جمع الأعداد من 1 وحتى عدد صحيح معين غير سالب.

يبدأ البرنامج بالتنفيذ من خلال مسلك التنفيذ الأساسي main، ومن ثم يُنشئ main مسلك ثاني يقوم بوظيفة الحساب وهو المسلك runner.

تتضمن جميع برامج Pthreads ملف الترويسة pthread.h، كما يمثل المتحول tid محدد الهوية للمسلك المراد إنشاؤه، كما يمثل attr التصريح عن مجموعة واصفات المسلك (حجم المكدس، ومعلومات الجدولة).

```
/* the thread will begin control in this function */
void *runner(void *param)
{
    int upper = atoi(param);
    int i;
    sum = 0;
    if (upper > 0) {
        for (i = 1; i <= upper; i++)
            sum += i;
    }
    pthread_exit(0);
}
```

12. التمارين

1. يُعرَّف مسلك التنفيذ بأنه الوحدة الأساسية في استخدام وحدة المعالجة، ويتألف من محدد هوية (ID)، وعدّاد برنامج، بالإضافة إلى مجموعة سجلات ومكدس:
   A. صح
   B. خطأ

2. من فوائد البرمجة باستخدام مسالك التنفيذ:
   A. التشارك في الموارد
   B. سرعة الاستجابة
   C. الاقتصاد
   D. جميع الإجابات صحيحة

3. يحتاج توليد إجراء جديد إلى موارد كثيرة على عكس توليد مسلك تنفيذ جديد:
   A. صح
   B. خطأ

4. يوجد حماية للمسالك أو عزل فيما بينها على عكس الإجراءات:
   A. صح
   B. خطأ

5. تتقاسم المسالك فيما بينها الفترة الزمنية المخصصة لتنفيذ الإجراء في المسالك على مستوى النواة:
   A. صح
   B. خطأ

6. تجري في هذا النموذج مقابلة أكثر من مسلك على مستوى المستخدم، بمسلك واحد على مستوى النواة:
   A. النموذج كثير إلى واحد
   B. النموذج واحد إلى واحد
   C. النموذج كثير إلى كثير
   D. جميع الإجابات خاطئة

7. يتيح النموذج "كثير إلى واحد" للمطوِّر إنشاء العدد الذي يرغب به من مسالك المستخدم، لكنه لا يسمح له بتحقيق توازي حقيقي في التنفيذ:

A. صح

B. خطأ

8. يقوم أحد المسالك بإنهاء المسلك المستهدف مباشرةً في الإيقاف المؤجل:

A. صح

B. خطأ

9. أهم فوائد مجمَّع مسالك التنفيذ: خدمة أسرع للطلبات من حالة مسلك تنفيذ واحد:

A. صح

B. خطأ

10. تعتبر Pthreads مكتبة برمجية تساعد على إنشاء مسالك التنفيذ ومزامنتها:

A. صح

B. خطأ

رقم التمرين	الإجابة الصحيحة
.1	(A)
.2	(D)
.3	(A)
.4	(B)
.5	(B)
.6	(A)
.7	(A)
.8	(B)
.9	(A)
.10	(A)

إدارة العتاديات

الكلمات المفتاحية:

المودیم، منفذ تسلسلي، إشارة رقمية، إشارة جيبية، الحاسب المضيف، الوسائط المتعددة، وحدة التغذية، اللوحة الأم، الحواسيب المحمولة، الحواسيب المسطحة، قلم رقمي، المساعد الرقمي الشخصي، تطبيق برمجي.

ملخص:

يركز هذا الفصل على إدارة العتاديات (المودیم وكرت الشبكة والطابعة)، واختيار البرمجيات والتجهيزات المناسبة للحاسب، بالإضافة إلى عرض سريع لبعض أنواع الحواسيب المحمولة.

أهداف تعليمية:

يهدف هذا الفصل إلى:

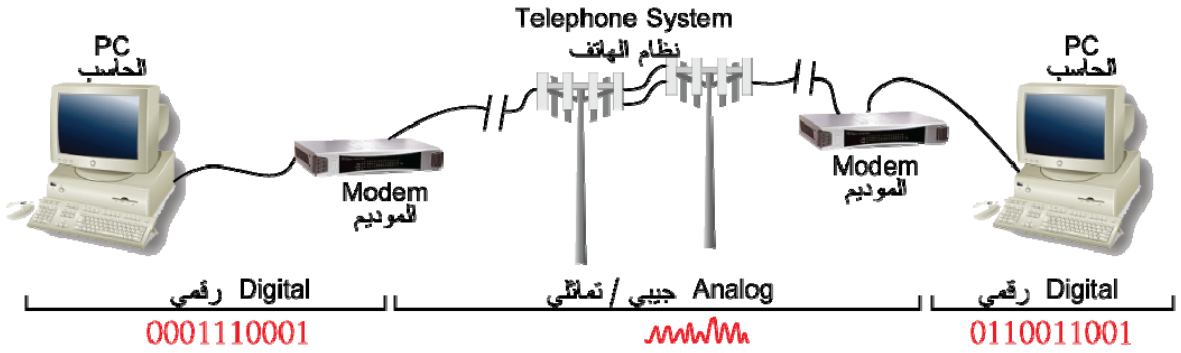
- التعرف على المودیم، سرعته، توصيله، بالإضافة إلى عملية تنصيبه.
- تنصيب كرت الشبكة، ومشاركة الملفات عبر الشبكة.
- تنصيب الطابعة، ومشاركتها على الشبكة.
- اختيار العتاديات المناسبة للحاسب.
- اختيار البرمجيات المناسبة للحاسب.
- عرض سريع لبعض أنواع الحواسيب المحمولة.

1. المودیم

يُستخدم المودیم للاتصال عبر خط الهاتف، حيث يمكن أن يكون:

- خارجي: يتم وصله بالحاسب من خلال منفذ تسلسلي serial port، أو منفذ USB
- داخلي: كرت موديم موصول بمأخذ ISA أو مأخذ PCI

بما أن المعطيات ضمن الحاسب مخزنة رقمياً، والإشارة التي تنتقل عبر الهاتف هي عبارة عن إشارة جيبية، لذلك يقوم المودیم بتحويل الإشارة الجيبية المنتقلة عبر خطوط الهاتف، إلى إشارة رقمية يفهمها الحاسب، وبالعكس.

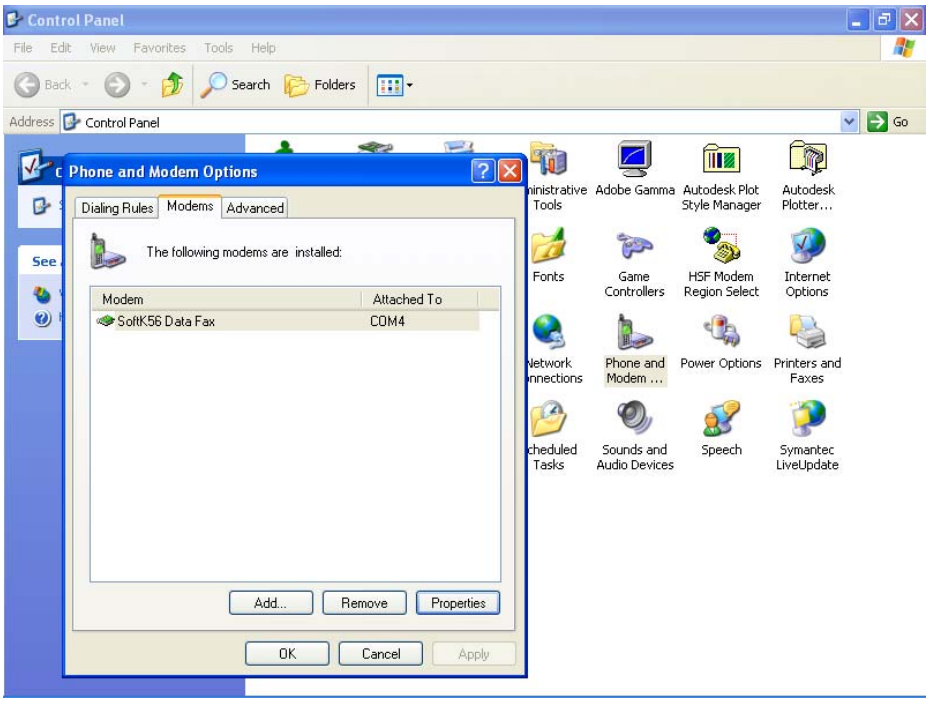


ملحق الرسم	
Telephone System	نظام الهاتف
Modem	المودیم
Analog	جيبي/تماثلي
Digital	رقمي
PC	الحاسب

2. سرعة المودیم

تُعبّر سرعة المودیم عن سرعة تمریر المعطیات عبر خط الهاتف، حیث تقدر بعدد البتات المنقولة في الثانیة (bps).

لمعرفة سرعة المودیم ومواصفاته الأخرى في نظام Windows، اختر لوحة التحكم control panel، ثم خیارات المودیم Modem and phone options.

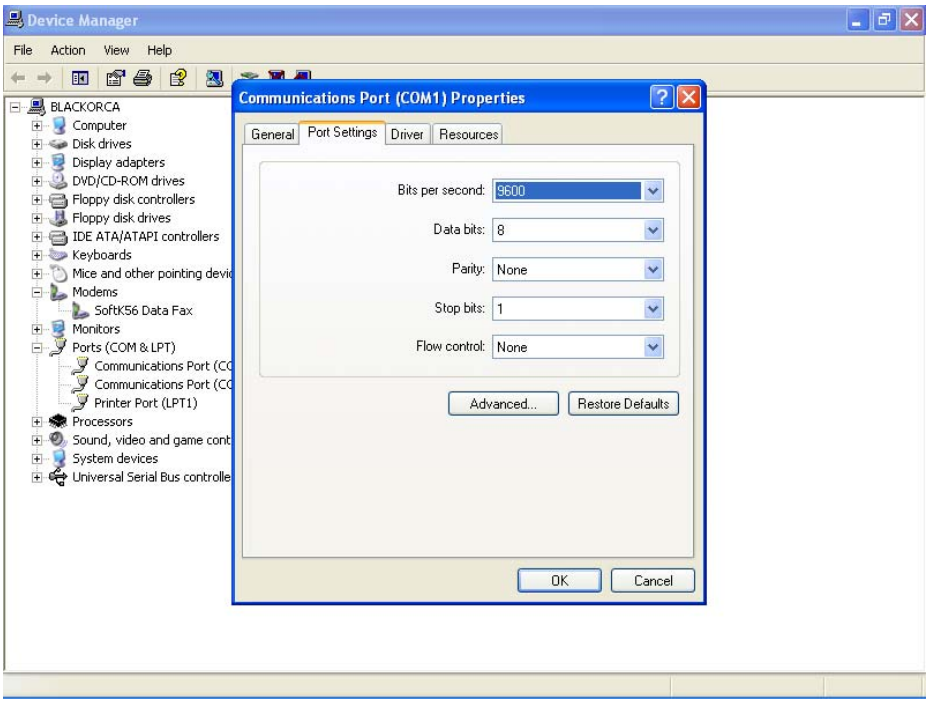


3. إعدادات منفذ الاتصال

يمكن الوصول إلى إعدادات منفذ الاتصال[1]، من خلال إدارة التجهيزة Device manager، ثم ضغط يميني على الموديم.

إعدادات المنفذ:

- عدد بتات المعطيات: عدد البتات المستخدمة لنقل محرف واحد من المعطيات (7 أو 8 بت).
- الزوجية: من أجل كشف الأخطاء (زوجي أو فردي أو لا شيء).
- بتات التوقف: الرمز الذي يعبر عن بداية أو نهاية محرف (1 أو 1.5 أو 2 بت).
- التحكم بالتدفق: كيفية التحكم بتدفق المعطيات (برمجياً Xon/Xoff، عتادياً RTS/CTS).



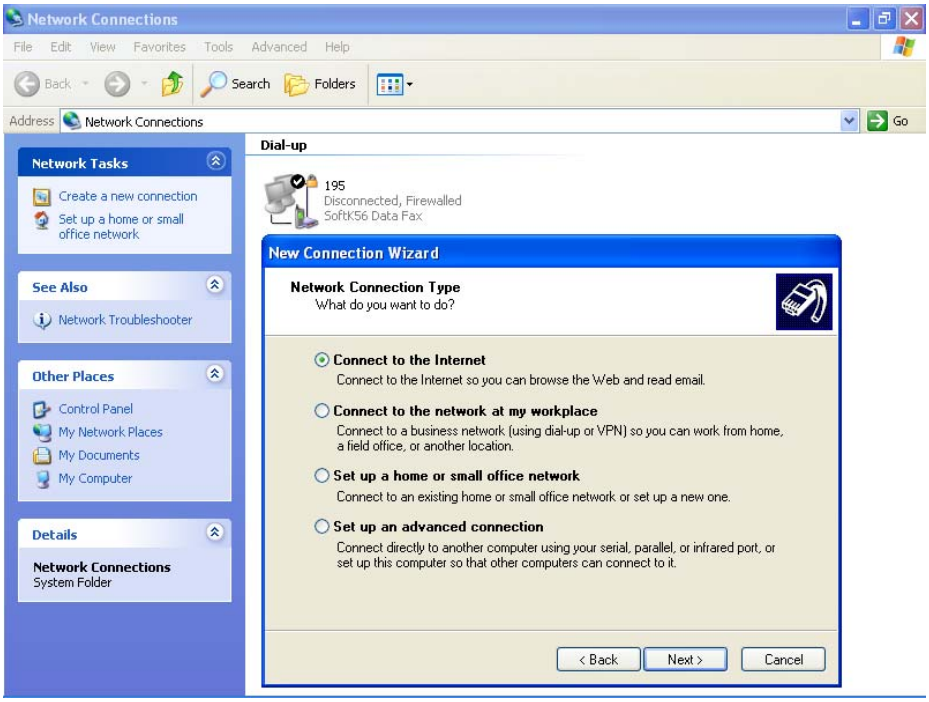
إعدادات المنفذ	الوصف	القيمة
عدد بتات المعطيات	عدد البتات المستخدمة لنقل محرف واحد من المعطيات	7 أو 8 بت
الزوجية	من أجل كشف الأخطاء	زوجي أو فردي أو لا شيء
بتات التوقف	الرمز الذي يعبر عن بداية أو نهاية محرف	1 أو 1.5 أو 2 بت
التحكم بالتدفق	كيفية التحكم بتدفق المعطيات	برمجياً (Xon/Xoff)، عتادياً (RTS/CTS)

1 منفذ الاتصال: المنفذ الذي يستخدمه الموديم.

4. تنصيب الموديم

بعد وضع الموديم في المأخذ المناسب، ثم إقلاع الحاسب:

- يقوم نظام Windows بملاحظة وجود تجهيزة جديدة، وفي معظم الحالات يقوم نظامWindows XP بالتعرف على الموديم وتنصيب البرمجيات اللازمة لعمله، وإذا لم يحدث ذلك عليك تنصيب البرمجيات من خلال الـCD المرافق للموديم.
- يتم التأكد أن الموديم قد تم تنصيبه بشكل صحيح، من خلال control panel ثم Phone and modem options، ثم خصائص الموديم.
- يجب وضع سرعة الموديم على السرعة القصوى.
- يتم بناء اتصال جديد، من خلال الضغط بالزر اليمين على My network places ثم خصائص ومن ثم بناء اتصال جديد، ثم تتبع الخطوات.



5. أدوات اختبار الموديم (HyperTerminal)

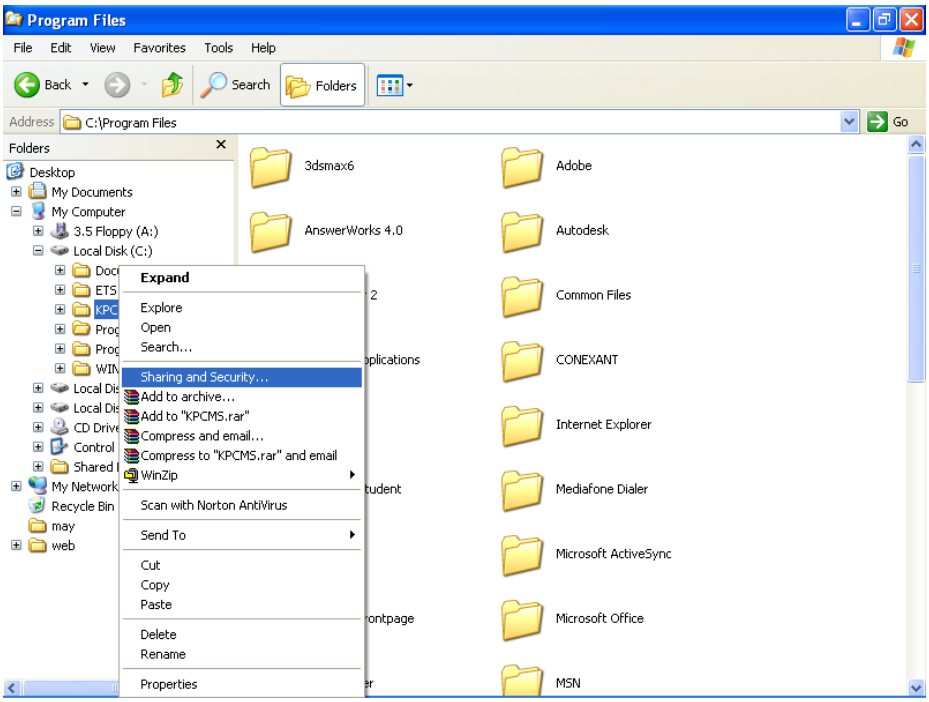
- أحد أفضل الأدوات المستخدمة لاختبار الموديم هي HyperTerminal، حيث يمكن من خلالها إجراء اتصالات هاتفية سهلة وسريعة عن طريق الحاسب.
- اضغط Start، All programs، Accessories، Communications، ثم HyperTerminal، وحاول إجراء اتصال هاتفي عن طريق هذه الأداة.

6. تنصيب كرت الشبكة

- يقوم نظام Windows بملاحظة وجود تجهيزة جديدة، وفي معظم الحالات يقوم نظامWindows XP بالتعرف على كرت الشبكة وتنصيب البرمجيات اللازمة لعمله، وإذا لم يحدث ذلك عليك تنصيب البرمجيات من خلال الـCD المرافق للكرت.
- تأكد أن كرت الشبكة قد تم تنصيبه بشكل صحيح، من خلال مدير التجهيزة Device manager، ثم ضغط يميني على الكرت للحصول على خصائصه.

7. مشاركة الملفات على الشبكة

- ضغط يميني على المجلد ثم اختيار مشاركة وأمن Sharing and Security.
- جرب هذا الخيار على مجلد ما، هل يمكنك وضع كلمة سر معينة لحماية الوصول إلى هذا المجلد؟



8. أسئلة 1

أجب عن الأسئلة التالية من خلال النظر إلى إعدادات الموديم على حاسبك:

- ما هو المنفذ الذي يستخدمه الموديم؟
- ما هو رقم المقاطعة التي يستخدمها الموديم؟
- ما هي السرعة القصوى للمنفذ؟
- ما هو عنوان المنفذ الذي يستخدمه الموديم؟

9. أسئلة 2

- ابحث في الإنترنت عن خصائص موديم خارجي يستخدم منفذ USB.
- ابحث في الإنترنت عن خصائص موديم خارجي يستخدم منفذ تسلسلي.
- ابحث في الإنترنت عن خصائص الموديم الداخلي.

10. تنصيب ومشاركة الطابعة

يتم وصل الطابعة إلى الحاسب عبر منفذ معين، ومن ثم يتم مشاركتها مع الأجهزة الأخرى عبر الشبكة. لكن يجب أن يحوي نظام التشغيل في كل حاسب، على السواقة الخاصة بالطابعة لكي يتم التعامل معها، فهذه السواقة عبارة عن طبقة الوصل بين التطبيقات (التي تستخدم آلية الطباعة)، وبين الطابعة.

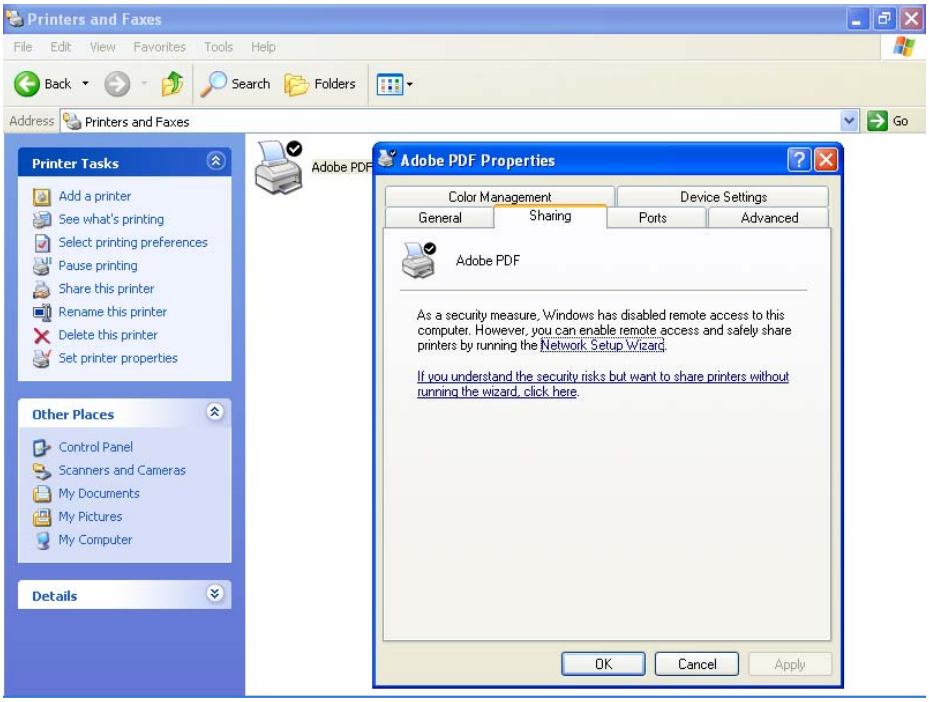
هنالك أيضاً أنواع من الطابعات يمكن وصلها مباشرة إلى الشبكة لكي يتم استخدامها من قبل جميع تجهيزات الشبكة.

11. تنصيب الطابعة

- أولاً يجب وصل الطابعة فيزيائياً إلى الحاسب عبر أحد المنافذ (منفذ تسلسلي، منفذ USB، منفذ SCSI، أو غيرها).
- يقوم نظام Windows بملاحظة وجود تجهيزة جديدة، وفي معظم الحالات يقوم نظامWindows XP بالتعرف على الطابعة وتنصيب البرمجيات اللازمة لعملها، وإذا لم يحدث ذلك عليك تنصيب البرمجيات من خلال الـCD المرافق للطابعة.
- يتم إضافة طابعة جديدة من خلال لوحة التحكم Control Panel، ثم Printers and Faxes، ثم إضافة طابعة جديدة Add Printer.
- يجب تجربة الطابعة للتأكد من أنها تعمل بشكل صحيح.

12. مشاركة الطابعة

- أولاً فتح نافذة الطابعة من خلال لوحة التحكم Control Panel، ثم Printers and Faxes، ثم ضغط يميني على الطابعة المراد مشاركتها، ومن ثم اختيار مشاركة sharing.
- إذا كان من المطلوب جعل السواقة الخاصة بالطابعة، متاحة للمستخدمين البعيدين، يجب الضغط على Additional Drivers.
- بعد أن تظهر نافذة Additional Drivers، يجب اختيار أنظمة التشغيل التي سيتمكن مستخدميها من الحصول على سواقة الطابعة المطلوب مشاركتها.



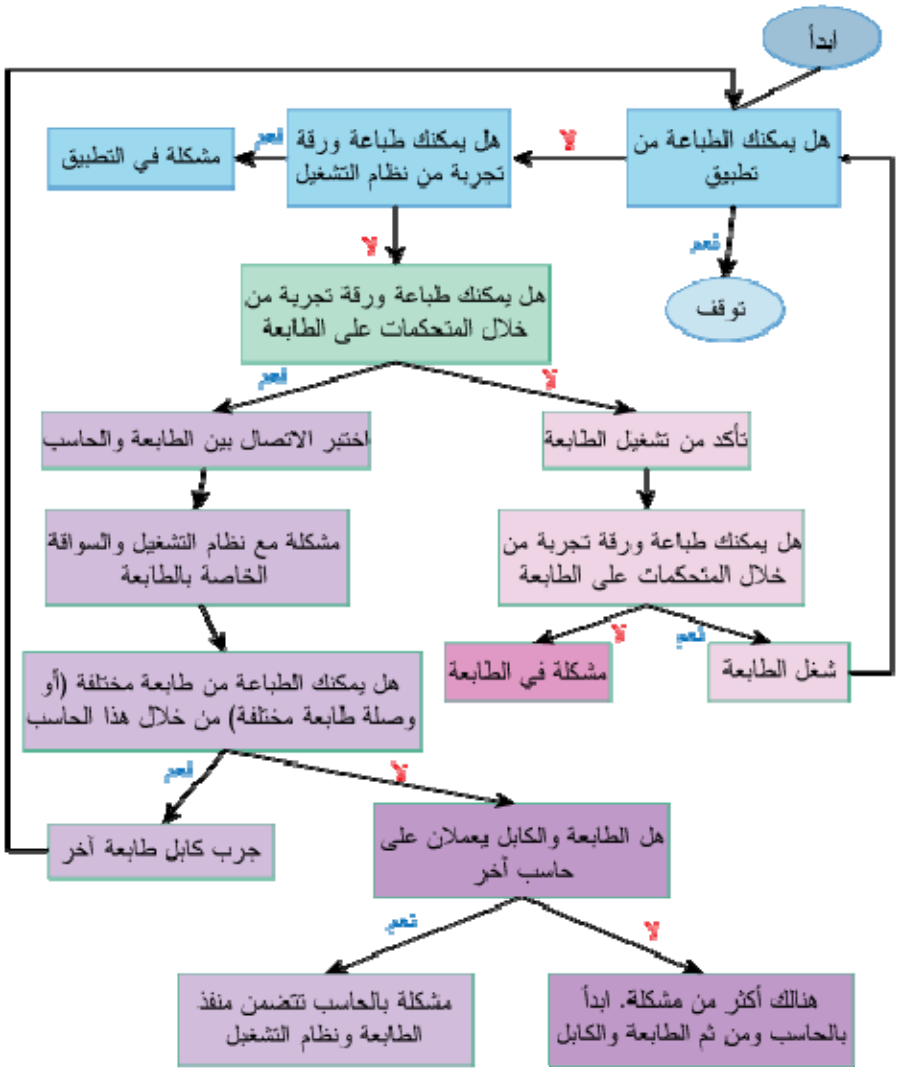
13. استخدام طابعة مشتركة على الشبكة

- أولاً فتح نافذة الطابعة من خلال لوحة التحكم Control Panel، ثم Printers and Faxes، ثم ضغط مزدوج على إضافة طابعة Add a printer.
- يجب اختيار Network printer، أو printer attached to another computer.
- ثم يجب ادخال اسم الحاسب المضيف واسم الطابعة.
- يقوم Windows XP بالبحث عن سواقة الطابعة على الحاسب المضيف، وفي حال عدم وجودها يجب اختيار Hard Disk من أجل تنصيب السواقة المناسبة من الـCD الخاص بالطابعة.
- ثم يتم اختيار هذه الطابعة على أنها الطابعة التلقائية أم لا (لكي يتم إرسال ملفات الطباعة إليها تلقائياً).

14. مشاكل الطابعة

يظهر المخطط التالي المشاكل العامة للطابعة:

- مشكلة في التطبيق.
- مشكلة في الطابعة.
- مشكلة في كابل الطباعة.
- مشكلة في نظام التشغيل وسواقات الطابعة.



15. تمرين 1

لديك طابعة تم تنصيبها على نظام Windows 98، وتريد أن تجعل هذه الطابعة متاحة لثمانية أجهزة Windows 98 أخرى على الشبكة، ما هي الطريقة الأفضل لذلك:

A. استخدام إضافة طابعة Add Printer من نافذة Printers في الأجهزة الثمانية.

B. استخدام جوار الشبكة Network Neighborhood من أجل تنصيب الطابعة على الأجهزة الثمانية.

C. استخدام الـCD الخاص بالطابعة على كل جهاز.

D. تنصيب الطابعة على الأجهزة الثمانية من خلال الحاسب المضيف، وذلك باستخدام جوار الشبكة Network Neighborhood للحاسب المضيف.

16. تمرين 2

لنفرض أنك حاولت طباعة ملف word على نظام Windows XP، وباستخدام طابعة معرفة على الشبكة وموصولة مباشرةً عليها، ولكن لم تتمكن من عملية الطباعة.

بالنظر إلى نافذة Printer and Faxes يظهر اسم الطابعة (\\SMITHWIN2K\HP LaserJet 8100).

ما هو برأيك سبب المشكلة:

A. الحاسب SMITHWIN2K لا يعمل.

B. الطابعة HP LaserJet 8100 غير موصولة.

C. الطابعة SMITHWIN2K غير موصولة.

D. الحاسب HP LaserJet 8100 لا ينتمي إلى مجموعة العمل.

E. مشكلة في حاسب Windows XP.

17. اختيار العتاديات المناسبة للحاسب

يختلف اختيار العتاديات المناسبة للحاسب، تبعاً للغرض الذي سيتم استخدام الحاسب لأجله:

- إذا كنت تريد استخدام الحاسب من أجل الألعاب وتطبيقات الوسائط المتعددة، تحتاج إلى وحدة معالجة بسرعة عالية، كرت فيديو بنوعية ممتازة، بالإضافة إلى ذاكرة كبيرة.
- إذا كنت تريد استخدام الحاسب بشكل كبير على شبكة، تحتاج إلى طاقة معالجة كبيرة.
- إذا كنت تريد الاتصال بالإنترنت عبر موديم، فيجب اختيار موديم بنوعية جيدة.
- يجب اختيار وحدة تغذية كافية لجميع مكونات الحاسب.

18. اختيار البرمجيات المناسبة للحاسب

يختلف اختيار البرمجيات المناسبة للحاسب، تبعاً للغرض الذي سيتم استخدام الحاسب لأجله، لذلك يجب الأخذ بعين الاعتبار الأسئلة التالية، أثناء اختيار البرمجيات:

- ما الذي تريده من التطبيق البرمجي؟
- هل التطبيق البرمجي متوافق مع التطبيقات أو المعطيات الأخرى في النظام؟
- هل ملفات الشرح المرافقة للتطبيق البرمجي، جيدة أم لا؟
- هل هنالك وسائل تدريب على هذا البرنامج، إذا كنت لا تجيد استخدامه؟
- ما هي سياسة شركة التطبيق من حيث تقديم النسخ المعدلة منه؟
- هل التطبيق معروف ومستخدم بكثرة أم لا؟

19. تمرين 3

لنفترض أنك تريد شراء حاسب جديد، أجب عن الأسئلة التالية لمساعدتك في اتخاذ القرار المناسب:

- ما هو الهدف أو الأهداف الأساسية للحاسب؟
- ما هي الوظائف التي يجب أن يحققها الحاسب من أجل كل هدف؟
- ما هي العتاديات والبرمجيات المناسبة لكل وظيفة؟
- من أجل كل مكون برمجي وعتادي، ما هو السؤال الذي تحتاج الإجابة عليه قبل اتخاذ قرارك النهائي؟

20. تمرين 4

لنفترض أنك تريد تجربة اللوحة الأم قبل استكمال عملية التنصيب، ما هي أقل التجهيزات التي تحتاجها لكي تستطيع إقلاع النظام والدخول إلى الـBIOS. اختر التجهيزات من القائمة التالية:

الماوس، وحدة التغذية، الشاشة، القرص الصلب، لوحة المفاتيح، القرص المرن، سواقة الـCD، اللوحة الأم، كرت الفيديو، الموديم، وحدة المعالجة، RAM.

21. الحواسيب المحمولة

يُصمّم هذا النوع من الحواسيب بهدف الاستخدام المتنقل، وهو يَستخدِم نفس تقنيات الحواسيب الشخصية ولكن مع بعض التعديلات، وذلك من أجل استخدام أقل للطاقة، شغل مساحة أقل، والعمل أثناء الحركة.

مميزات نظم تشغيل Windows 98/2000/XP على الحواسيب المحمولة:

- إمكانية استخدام اتصالي موديم بنفس الوقت، وذلك من أجل تسريع عملية نقل المعطيات أثناء الاتصال عن طريق خط الهاتف.
- إمكانات إضافية لإدارة الطاقة، مثل تشغيل الحاسب المحمول عن طريق تجهيزة معينة أو العكس، إطفاء الحاسب بشكل أتوماتيكي عندما لا يكون في حالة استخدام.
- إمكانية اختيار التجهيزات المراد تشغيلها أثناء الإقلاع، وذلك تبعاً لظروف استخدام الحاسب، مثل استخدام الحاسب داخل السيارة أثناء السفر، وهو استخدام مختلف عن استخدامه في المنزل.

22. الحواسيب المسطحة

الحواسيب المسطحة (Tablet PCs) هي عبارة عن نوع من الحواسيب المحمولة مصمم خصيصاً للمستخدمين المهتمين بالواجهات البيانية، بالإضافة إلى أنها أصغر حجماً، وأكثر محمولية، مع إمكانية استخدام قلم رقمي للتعامل مع الواجهات، أو حتى الكتابة بخط اليد.

23. المساعد الرقمي الشخصي (PAD)

إن الPAD عبارة عن حاسب جيب صغير، يحوي نظام تشغيل وبعض التطبيقات كدفتر العناوين، مفكرة، محرر نصوص، متصفح محدود لصفحات الوِب، بالإضافة إلى ارسال واستقبال البريد الالكتروني.

يمكن وصله إلى الحاسب من خلال وصلة تسلسلية أو وصلة USB، مع إمكانية نقل المعطيات عبر هذه الوصلة من وإلى الحاسب.

هنالك نظامي تشغيل أساسيين للPAD:

- الأول من شركة Microsoft وهو Windows Mobile
- الثاني من شركة PalmSource ويدعى Palm OS

24. التمارين:

1. يقوم الموديم بتحويل الإشارة الجيبية المنتقلة عبر خطوط الهاتف، إلى إشارة رقمية يفهمها الحاسب، وبالعكس:

A. صح

B. خطأ

2. تُعبّر سرعة الموديم عن سرعة تمرير المعطيات عبر خط الهاتف، حيث تقدر بعدد البتات المنقولة في الثانية (bps):

A. صح

B. خطأ

3. عدد البتات المستخدمة لنقل محرف واحد من المعطيات:

A. 4 أو 5 بت

B. 6 أو 7 بت

C. 7 أو 8 بت

D. 9 أو 10 بت

4. أحد أفضل الأدوات المستخدمة لاختبار الموديم هي HyperTerminal:

A. صح

B. خطأ

5. واحد مما يلي يعتبر من المشاكل العامة للطابعة:

A. مشكلة في الطابعة

B. مشكلة في كابل الطباعة

C. مشكلة في التطبيق

D. جميع الإجابات صحيحة

6. إذا كنت تريد استخدام الحاسب من أجل الألعاب وتطبيقات الوسائط المتعددة تحتاج إلى:
   A. طاقة معالجة كبيرة
   B. ذاكرة كبيرة
   C. وحدة معالجة بسرعة عالية
   D. الإجابتين 2 و3
   E. الإجابتين 1 و3

7. من مميزات نظم تشغيل Windows 98/2000/XP على الحواسيب المحمولة:
   A. إمكانية اختيار التجهيزات المراد تشغيلها أثناء الإقلاع
   B. إمكانية استخدام اتصالي موديم بنفس الوقت
   C. إمكانات إضافية لإدارة الطاقة
   D. جميع الإجابات صحيحة

8. من مميزات الحواسيب المسطحة:
   A. أصغر حجماً
   B. إمكانية استخدام قلم رقمي للتعامل مع الواجهات
   C. مصمم خصيصاً للمستخدمين المهتمين بالواجهات البيانية
   D. جميع الإجابات صحيحة

رقم التمرين	الإجابة الصحيحة
.1	(A)
.2	(A)
.3	(C)
.4	(A)
.5	(D)
.6	(D)
.7	(D)
.8	(D)

جدولة وحدة المعالجة

الكلمات المفتاحية:

جدولة وحدة المعالجة – دفعة وحدة معالجة – دفعة دخل/خرج – مجدول وحدة المعالجة – مسند المهمات – خوارزمية FCFS (القادم أولاً يخدم أولاً) – خوارزمية SJF (الأقصر عملاً أولاً) – خوارزمية الجدولة وفق الأولوية – خوارزمية الجدولة الدائرية – حصة زمنية – خوارزمية الجدولة وفق أرتال متعددة المستوى – خوارزمية الجدولة وفق أرتال متعددة المستوى بتغذية راجعة.

ملخص:

يركز هذا الفصل على التعرف على مفهوم جدولة وحدة المعالجة – معايير الجدولة – خوارزميات الجدولة.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- التعرف على مفهوم جدولة وحدة المعالجة.
- مجدول وحدة المعالجة – ومسند المهمات.
- معايير الجدولة.
- خوارزميات الجدولة (FCFS – SJF – الجدولة وفق الأولوية – الجدولة الدائرية – الجدولة وفق أرتال متعددة المستوى – الجدولة وفق أرتال متعددة المستوى بتغذية راجعة).

1. جدولة وحدة المعالجة

إن جدولة وحدة المعالجة هي أساس نظم التشغيل المتعدّدة البرمجة، حيث يستطيع نظام التشغيل من خلال تبديل التنفيذ بين الإجرائيات، من جعل الحاسوب أكثر إنتاجية.

إن الهدف الأساسي من تعدّد البرمجة هو أن يكون لدينا دوماً إجرائية تُنفذ، وذلك لرفع معدل استخدام وحدة المعالجة إلى أعلى حد ممكن، ففي نظام أحادي المعالج، لا يمكن وجود أكثر من إجرائية واحدة في حال التنفيذ، وإذا وجدت أكثر من إجرائية في النظام، فإن واحدة منها ستكون في حالة التنفيذ، وأما البقية فستنتظر إلى أن تحرر وحدة المعالجة وتعاد جدولتها.

2. دفعات وحدة المعالجة والدخل/الخرج

يعتمد نجاح جدولة وحدة المعالجة على الخاصة التي تلاحظ في الإجرائيات: وهي أن تنفيذ الإجرائية يمر في مراحل متتالية من التنفيذ في وحدة المعالجة، ومن انتظار الدخل/الخرج.

يبدأ تنفيذ كل إجرائية بدفعة وحدة المعالجة، تليها دفعة دخل/خرج، ثم وحدة معالجة وهكذا، حتى ينتهي التنفيذ بدفعة وحدة معالجة عند طلب النظام إنهاء تنفيذ الإجرائية.

إن البرنامج المقيد بوحدة المعالجة سوف يستهلك عدد قليل من دفعات وحدة المعالجة الطويلة جداً، بينما البرنامج المقيد بالدخل/الخرج سوف يستهلك دفعات طويلة من الدخل/الخرج.

3. مجدول وحدة المعالجة

عندما تصبح وحدة المعالجة في حالة راحة، يترتب على نظام التشغيل انتقاء إحدى الإجرائيات الموجودة في رتل الجاهزات من أجل تنفيذها.

تجري عملية انتقاء الإجرائيات بواسطة المجدول القصير الأمد، حيث ينتقي هذا المجدول واحداً من الإجرائيات الجاهزة للتنفيذ الموجودة في الذاكرة، ويخصص وحدة المعالجة لها.

يجري اتخاذ قرارات الجدولة في الظروف التالية:

- عندما تتبدل حالة إجرائية من حالة التنفيذ إلى حالة الانتظار (مثلاً طلب دخل/خرج، أو انتظار انتهاء إحدى الإجرائيات الأولاد)
- عندما تتبدل حالة إجرائية من حالة التنفيذ إلى الحالة الجاهزة (مثلاً عند حصول مقاطعة)
- عندما تتبدل حالة إجرائية من حالة الانتظار إلى الحالة الجاهزة (مثلاً عند انتهاء عملية دخل/خرج)
- عند انتهاء تنفيذ إجرائية

4. مُسند المهمات

إن مُسند المهمات هو أحد المكونات التي تساهم في جدولة وحدة المعالجة، حيث يسلم قيادة وحدة المعالجة إلى الإجرائية المختارة، بواسطة المجدول قصير الأمد، وهذا يتضمن:

- تبديل السياق
- الانتقال إلى نمط المستخدم
- القفز إلى الموقع المناسب في برنامج المستخدم من أجل التنفيذ

كما يجب أن يكون مسند المهمات سريعاً ما أمكن، إذ يحدث استدعاؤه مع كل تبديل إجرائية.

5. معايير الجدولة

تتمتع خوارزميات الجدولة المختلفة لوحدات المعالجة بخصائص مختلفة، لذلك يجب اختيار الخوارزمية الأنسب للاستخدام في حالة معينة.

المعايير المستخدمة للمقارنة بين الخوارزميات:

- معدّل استخدام وحدة المعالجة: يجب أن تكون وحدة المعالجة مشغولة قدر الإمكان.
- معدّل التدفق: عدد الإجرائيات التي ينتهي تنفيذها في واحدة الزمن.
- زمن تنفيذ المهمة: هو الوقت اللازم لتنفيذ الإجرائية، أي الزمن بين طلب تنفيذ الإجرائية وإنهائها (مجموع الفترات الزمنية في انتظار الدخول إلى الذاكرة، والانتظار في رتل الجاهزات، والتنفيذ في وحدة المعالجة، والقيام بالدخل/الخرج).
- زمن الانتظار: هو الوقت الذي تقضيه إجرائية منتظرة في رتل الجاهزات (يساوي مجموع فترات الانتظار في رتل الجاهزات).
- زمن الاستجابة: يعتبر الزمن بين إرسال طلب وحدوث أول استجابة (الفترة الزمنية قبل حدوث الاستجابة) معيار مهم في الأنظمة التفاعلية، حيث يكون الاهتمام بسرعة استجابة النظام.

6. خوارزمية الجدولة (القادم أولاً يُخدّم أولاً) "FCFS"

تعتبر أبسط خوارزميات جدولة وحدة المعالجة، فالإجرائية التي تطلب وحدة المعالجة أولاً، هي التي تحصل عليها أولاً.

من الناحية التقنية يمكن تحقيق هذه الخوارزمية من خلال رتل، فعند دخول إجرائية إلى رتل الجاهزات، تربط كتلة تحكم الإجرائية الخاصة بها إلى مؤخرة الرتل، وعند تحرّر وحدة المعالجة، يجري إسنادها إلى الإجرائية التي في مقدمة الرتل.

سيئات هذه الخوارزمية:

- ما إن تُعطى وحدة المعالجة إلى إجرائية ما حتى تحتكرها، ولا تحرّرها إلا عند انتهاء التنفيذ أو طلب عملية دخل/خرج (بينما من الضروري حصول الإجرائيات على وحدة المعالجة خلال فترات زمنية نظامية).
- قد يحدث انتظار مجموعة من الإجرائيات الصغيرة، خروج إجرائية كبيرة من وحدة المعالجة، وهذا ما يخفض من معدّل استخدام وحدة المعالجة والتجهيزات.

7. خوارزمية الجدولة (الأقصر عملاً أولاً) "SJF"

تربط هذه الخوارزمية بكل إجرائية دفعة الـCPU التالية المتعلقة بها، فعندما تتحرّر وحدة المعالجة تُعطى للإجرائية ذات دفعة التنفيذ التالية الأصغر زمناً، وإذا وجدت إجرائيتان لهما طول دفعة واحد يجري الاختيار حسب FCFS.

إن هذه الخوارزمية تحقّق أصغر زمن انتظار وسطي، لمجموعة من الإجرائيات المعطاة، فمن خلال تمرير إجرائية قبل إجرائية أخرى أطول منها، يقصر زمن انتظار الإجرائية القصيرة، بدلاً من تطويل زمن انتظار الإجرائية الطويلة.

لكن الصعوبة الحقيقية في هذه الخوارزمية هي معرفة مدة الطلب التالي لوحدة المعالجة، لذلك يجري توقع هذه القيمة بالاعتماد على أطوال دفعات التنفيذ السابقة (نتوقع أن تكون دفعة CPU مشابهة للدفعات السابقة).

كذلك يمكن أن تكون خوارزمية SJF في إحدى حالتين:

فإذا وردت إجرائية جديدة إلى رتل الجاهزات، وكانت دفعة الـCPU التالية لها أقصر مما بقي للإجرائية التي يجري تنفيذها، عندها إما أن يجري إيقاف الإجرائية التي تنفذ، ويتم اسناد وحدة المعالجة إلى الإجرائية الجديدة، أو تتابع الإجرائية تنفيذها دون توقف، حتى انتهاء الفترة المخصصة لها.

8. خوارزمية الجدولة وفق الأولوية

في خوارزمية الجدولة وفق الأولوية، تُحدّد أولوية بكل إجرائية، ويجري اختيار الإجرائية ذات الأولوية العليا.
نلاحظ أن خوارزمية SJF هي حالة خاصة من خوارزمية الجدولة وفق الأولوية مع اعتبار أولوية الإجرائية هي مقلوب زمن دفعة الـCPU التالية.
يمكن أن تُعرّف الأولويات إما داخلياً أو خارجياً، حيث تستخدم الأولويات المعرفة داخلياً مقادير قابلة للقياس مثل (الحدود الزمنية، متطلبات الذاكرة، عدد الملفات المفتوحة) من أجل حساب الأولوية، أما الأولويات الخارجية فيجري تعيينها بواسطة معايير خارجية على نظام التشغيل مثل (أهمية الإجرائية، المبلغ المدفوع).

المشكلة الأساسية لهذه الخوارزمية:

هي التوقف اللانهائي أو الحرمان (إذا انتظرت الإجرائية الجاهزة للتنفيذ فراغ وحدة المعالجة لفترة طويلة)، وذلك لأن أولويتها منخفضة بالنسبة إلى جميع الإجرائيات الأخرى في النظام.
هناك حل لمشكلة الحرمان، وذلك بزيادة أولوية الإجرائية منخفضة الأولوية تدريجياً، مع استمرار بقائها بانتظار التنفيذ.

9. خوارزمية الجدولة الدائرية

إن خوارزمية الجدولة الدائرية RR (round–robin)، مصممة من أجل نظم المشاركة بزمن المعالج، وهي مشابهة لجدولة FCFS ولكن مع تعريف حصة زمنية (وحدة صغيرة من الزمن عادةً بين 10 و 100 ملّي).
يتم التعامل في هذه الخوارزمية مع رتل الجاهزات بطريقة دوّارة، وتنتقل وحدة المعالجة بين إجرائيات الرتل، بحيث أن لكل إجرائية مدة زمنية لا تتجاوز حصة زمنية محدّدة مسبقاً.
تبدأ وحدة المعالجة بتنفيذ الإجرائية الأولى في رتل الجاهزات، ومن ثم إذا كان للإجرائية دفعة وحدة معالجة أقصر من حصة زمنية واحدة، عندها تتخلى عن وحدة المعالجة عند انتهاء دفعة التنفيذ، وتعود إلى مؤخرة رتل الجاهزات، أما إذا كانت دفعة وحدة المعالجة أطول من حصة زمنية واحدة، عندها مع اكتمال الحصة الزمنية، تجري مقاطعة الإجرائية، فيتوقف تنفيذها ويجري تبديل السياق مع الإجرائية الأولى في رتل الجاهزات، بينما تعود هذه الإجرائية إلى مؤخرة الرتل.
يعتمد أداء هذه الخوارزمية على حجم الحصة الزمنية، فإذا كانت كبيرة جداً تتحول إلى FCFS، وإذا كانت صغيرة جداً يُصبح هناك تشارك كامل بالمعالج.

10. الجدولة وفق أرتال متعددة المستوى

تُقسم الإجرائيات إلى مجموعات منفصلة حسب نوعها مثلاً (إجرائيات تفاعلية أو إجرائيات خلفية)، حيث يكون لكل نوع من هذه الإجرائيات متطلبات مختلفة.

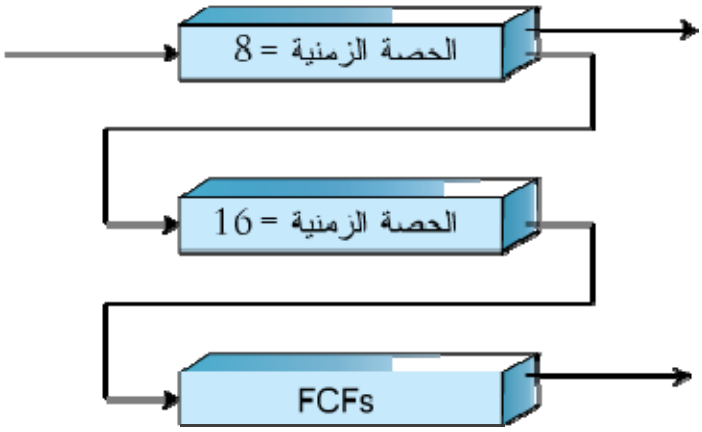
ففي هذه الخوارزمية يُجزأ رتل الجاهزات إلى عدة أرتال مستقلة، ويجري ربط الإجرائية برتل واحد دائماً حسب خواصها، كما يمكن أن يكون لكل رتل خوارزمية جدولة خاصة به.

يجري أيضاً تطبيق الجدولة بين الأرتال المختلفة، فمثلاً يتم اعطاء أولوية لكل رتل. كما يمكن اتباع طريقة تعتمد على تقسيم الزمن بين الأرتال، حيث يحصل كل رتل على حصة محددة من زمن وحدة المعالجة.

11. الجدولة وفق أرتال متعدّدة المستوى بتغذية راجعة

في خوارزمية الجدولة وفق أرتال متعدّدة المستوى، توضع الإجرائيات عند دخولها النظام في رتل محدد، ولا تنتقل منه إلى أي رتل آخر. بينما في خوارزمية الجدولة وفق أرتال متعدّدة المستوى بتغذية راجعة، فإنه يمكن للإجرائيات الانتقال بين الأرتال.

تقوم فكرة الخوارزمية على فصل الإجرائيات المميزة بدفعات وحدة المعالجة، فمثلاً إذا كانت الإجرائية تستخدم وقت كبير من زمن وحدة المعالجة فإنها تنتقل إلى رتل ذو أولوية منخفضة، وبذلك تبقى الإجرائيات المقيدة بالدخل/الخرج والإجرائيات التفاعلية في أرتال ذات أولوية عليا، ولحل مشكلة الحرمان، تنتقل الإجرائية إلى رتل ذو أولوية أعلى، إذا بقيت فترة طويلة في رتل ذو أولوية دنيا.



تُعرّف هذه الخوارزمية عموماً بالمعاملات التالية:

- عدد الأرتال.
- خوارزمية الجدولة المطبقة على الرتل.
- الطريقة المستخدمة لتحديد متى يجري نقل الإجرائية إلى رتل ذو أولوية أعلى (رفع الأولوية).
- الطريقة المستخدمة لتحديد متى يجري نقل الإجرائية إلى رتل ذو أولوية أدنى (خفض الأولوية).
- الطريقة المستخدمة لتحديد الرتل الذي يجب إدخال الإجرائية فيه، عندما تطلب تلك الإجرائية خدمة معينة.

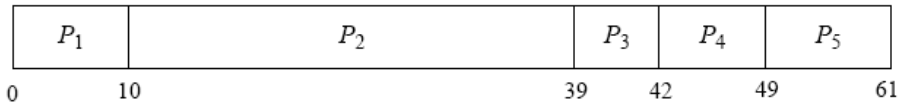
12. تمرين

لنأخذ خوارزمية FCFS (القادم أولاً يخدم أولاً)، SJF (الأقصر عملاً أولاً)، RR (الجدولة الدائرية)، وإذا افترضنا أن الحصة الزمنية تساوي 10 ملّي ثانية، ما هي الخوارزمية التي سوف تعطي أصغر زمن انتظار وسطي؟ علماً أن جميع الإجرائيات قدمت إلى النظام في اللحظة 0.

الإجرائية	زمن الرشقة
P1	10
P2	29
P3	3
P4	7
P5	12

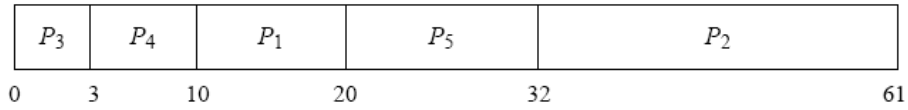
13. حل التمرين (الخوارزمية FCFS)

- في حال خوارزمية FCFS، تُنفذ الإجرائيات وفق الترتيب الآتي: P1 تليها P2 تليها P3 تليها P4 ثم P5.
- زمن الانتظار يساوي 0 ملّي ثانية للإجرائية P1، و10 ملّي ثانية للإجرائية P2، و39 ملّي ثانية للإجرائية P3، و42 ملّي ثانية للإجرائية P4، و49 ملّي ثانية للإجرائية P5، لذا يساوي زمن الانتظار الوسطي: (0 + 10 + 39 + 42 + 49) ÷ 5 = 28 ملّي ثانية.



14. حل التمرين (الخوارزمية SJF)

- في حال خوارزمية SJF، تُنفذ الإجرائيات وفق الترتيب الآتي: P3 تليها P4 تليها P1 تليها P5 ثم P2.
- زمن الانتظار يساوي 10 ملّي ثانية للإجرائية P1، و32 ملّي ثانية للإجرائية P2، و0 ملّي ثانية للإجرائية P3، و3 ملّي ثانية للإجرائية P4، و20 ملّي ثانية للإجرائية P5، لذا يساوي زمن الانتظار الوسطي: (10 + 32 + 0 + 3 + 20) ÷ 5 = 13 ملّي ثانية.



15. حل التمرين (الخوارزمية RR)

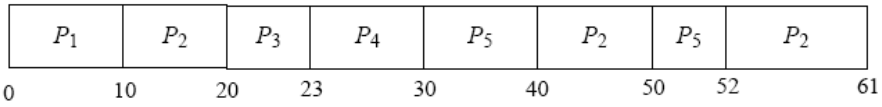
في حال خوارزمية RR، تُنفذ الإجرائيات وفق الترتيب الآتي: P1 تليها P2 تليها P3 تليها P4 تليها P5 تليها P2 تليها P5 ثم P2.

حيث تبدأ الإجرائية P2 بالتنفيذ، ثم توقف بعد 10 ملّي ثانية (مقدار الحصة الزمنية) من بدئها وتعاد مرة أخرى إلى رتل الانتظار.

زمن الانتظار يساوي 0 ملّي ثانية للإجرائية P1، و32 ملّي ثانية للإجرائية P2، و20 ملّي ثانية للإجرائية P3، و23 ملّي ثانية للإجرائية P4، و40 ملّي ثانية للإجرائية P5. لذا يساوي زمن الانتظار الوسطي:

(0 + 32 + 20 + 23 + 40) ÷ 5 = 23 ملّي ثانية.

نلاحظ بعد حساب الزمن الوسطي في حالة الخوارزميات الثلاث، أن الزمن الوسطي في حال خوارزمية SJF أقل من نصف زمن الانتظار الوسطي الذي تعطيه الجدولة FCFS، على حين تعطينا خوارزمية RR قيمة وسط بين القيمتين السابقتين.



16. التمارين:

1. يعتمد نجاح جدولة وحدة المعالجة على الخاصة التي تلاحظ في الإجرائيات: وهي أن تنفيذ الإجرائية يمر في مراحل متتالية من التنفيذ في وحدة المعالجة، ومن انتظار الدخل/الخرج:
   A. صح
   B. خطأ

2. يجري اتخاذ قرارات الجدولة في إحدى الظروف التالية:
   A. عندما تتبدل حالة إجرائية من حالة الانتظار إلى الحالة الجاهزة (مثلاً عند انتهاء عملية دخل/خرج).
   B. عندما تتبدل حالة إجرائية من حالة التنفيذ إلى حالة الانتظار
   C. عندما تتبدل حالة إجرائية من حالة التنفيذ إلى الحالة الجاهزة
   D. جميع الإجابات صحيحة

3. يجري اتخاذ قرارات الجدولة في إحدى الظروف التالية:
   A. عندما تتبدل حالة إجرائية من حالة الانتظار إلى الحالة الجاهزة
   B. عند انتهاء تنفيذ إجرائية
   C. عندما تتبدل حالة إجرائية من حالة التنفيذ إلى حالة الانتظار
   D. جميع الإجابات صحيحة

4. إن مُسند المهمات هو أحد المكونات التي تساهم في جدولة وحدة المعالجة:
   A. صح
   B. خطأ

5. المعايير المستخدمة للمقارنة بين الخوارزميات:
   A. معدّل استخدام وحدة المعالجة
   B. زمن الاستجابة
   C. معدّل التدفق
   D. جميع الإجابات صحيحة

6. هو الوقت الذي تقضيه إجرائية منتظرة في رتل الجاهزات:
   A. زمن الاستجابة
   B. زمن الانتظار
   C. زمن التدفق
   D. زمن تنفيذ المهمة

7. تعتبر أبسط خوارزميات جدولة وحدة المعالجة:
   A. القادم أولاً يُخدّم أولاً
   B. الأقصر عملاً أولاً
   C. الجدولة الدائرية
   D. الجدولة وفق الأولوية

8. هذه الخوارزمية تحقّق أصغر زمن انتظار وسطي، لمجموعة من الإجرائيات المعطاة:
   A. القادم أولاً يُخدّم أولاً
   B. الأقصر عملاً أولاً
   C. الجدولة الدائرية
   D. الجدولة وفق الأولوية

9. المشكلة الأساسية لهذه الخوارزمية هي التوقف اللانهائي أو الحرمان (إذا انتظرت الإجرائية الجاهزة للتنفيذ فراغ وحدة المعالجة لفترة طويلة):
   A. القادم أولاً يُخدّم أولاً
   B. الأقصر عملاً أولاً
   C. الجدولة الدائرية
   D. الجدولة وفق الأولوية

10. هذه الخوارزمية مصممة من أجل نظم المشاركة بزمن المعالج:
   A. القادم أولاً يُخدّم أولاً
   B. الأقصر عملاً أولاً
   C. الجدولة الدائرية
   D. الجدولة وفق الأولوية

11. يعتمد أداء هذه الخوارزمية على حجم الحصة الزمنية:

A. القادم أولاً يُخدّم أولاً

B. الأقصر عملاً أولاً

C. الجدولة الدائرية

D. الجدولة وفق الأولوية

12. في هذه الخوارزمية يُجزأ رتل الجاهزات إلى عدة أرتال مستقلة، ويجري ربط الإجرائية برتل واحد دائماً حسب خواصها، كما يمكن أن يكون لكل رتل خوارزمية جدولة خاصة به:

A. القادم أولاً يُخدّم أولاً

B. الأقصر عملاً أولاً

C. الجدولة وفق أرتال متعددة المستوى

D. الجدولة وفق الأولوية

13. تقوم فكرة الخوارزمية على فصل الإجرائيات المميزة بدفعات وحدة المعالجة:

A. القادم أولاً يُخدّم أولاً

B. الجدولة وفق أرتال متعددة المستوى

C. الجدولة وفق أرتال متعددة المستوى بتغذية راجعة

D. الجدولة وفق الأولوية

رقم التمرين	الإجابة الصحيحة
.1	(A)
.2	(D)
.3	(D)
.4	(A)
.5	(D)
.6	(B)
.7	(A)
.8	(B)
.9	(D)
.10	(C)
.11	(C)
.12	(C)
.13	(C)

إدارة الإجراءات

الكلمات المفتاحية:

الإجراء – حالة الإجراء – كتلة تحكم الإجراء – رتل الأعمال – رتل الجاهزات – رتل التجهيزة – المجدول طويل الأمد – المجدول قصير الأمد – الإجراء المُقيد بالدخل/الخرج – الإجراء المقيد بوحدة المعالجة – الإجراءات المتعاونة.

ملخص:

يركّز هذا الفصل على التعرف على مفهوم الإجراء وحالاته، وعلى تعريف كتلة تحكم الإجراء، وعلى أساليب جدولة الإجراءات وتبديل السياق فيما بينها، بالإضافة إلى العمليات على الإجراءات من إنشاء وحذف، وتعاون واتصال.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- التعرف على مفهوم الإجراء.
- حالة الإجراء.
- كتلة تحكم الإجراء.
- جدولة الإجراءات، ومجدولات نظام التشغيل.
- تبديل السياق بين الإجراءات.
- العمليات على الإجراءات (إنشاء وحذف).
- الإجراءات المستقلة والإجراءات المتعاونة.
- الاتصال بين الإجراءات (مباشر أو لا مباشر).
- التزامن بين الإجراءات المتصلة.

1. مفهوم الإجراء

سمحت النظم الحاسوبية الأولى بتنفيذ برنامج واحد فقط في وقت معين، حيث كان البرنامج يتحكّم كلياً بالنظام، وكان بوسعه النفاذ إلى كل موارده، أما النظم الحاسوبية اليوم، فتسمح بتحميل عدة برامج في الذاكرة وتنفيذها بالتساير، ومن هنا ظهر مفهوم الإجراء، والذي أصبح وحدة العمل الأساسية في هذه النظم.

إذن يتألف النظام من مجموعة إجراءات:

إجراءات نظام التشغيل، وإجراءات المستخدم، وكل هذه الإجراءات قادرة على التنفيذ بالتساير، حيث يجري تبديل وحدة المعالجة فيما بينها.

2. الإجراء

الإجراء هو أي برنامج أثناء التنفيذ، فالبرنامج بحد ذاته هو كيان غير ناشط، بينما الإجراء هو كيان ناشط تُنفذ تسلسلياً، يحدّد عدّاد البرنامج فيه التعليمة التالية التي يجب تنفيذها.

يتألف الإجراء من رمّاز البرنامج، الفعالية الحالية وتُمثل بقيمة عدّاد البرنامج ومحتويات سجلات المعالج، كما يتضمن الإجراء عموماً مكدس (stack) يحوي معطيات مؤقتة (معاملات الطرائق، عناوين العودة، متغيرات محلية)، بالإضافة إلى مقطع معطيات يتضمن متغيرات عامة.

يمكن أن يرتبط إجراءان ببرنامج واحد، لكنهما يُنفذان تنفيذاً مستقلاً. مثلاً عندما يقوم المستخدم باستدعاء عدة نسخ من محرّر النصوص، فإن كل واحدة من هذه النسخ هي عبارة عن إجراء مستقل، على الرغم من أن رمّاز البرنامج هو نفسه، ولكن مقاطع المعطيات متباينة.

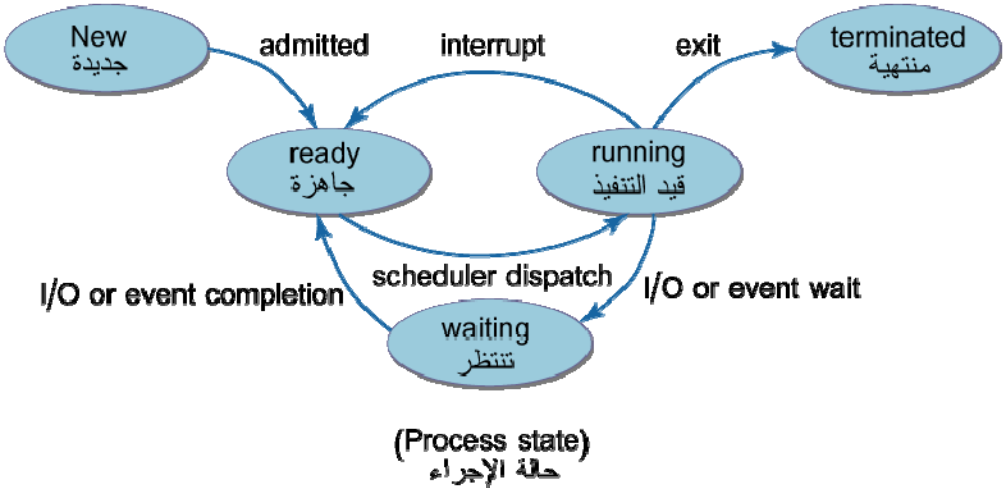
3. حالة الإجراء

تتغير حالة الإجراء أثناء تنفيذه، وتُعرف بالفاعلية الحالية لذلك الإجراء.

يمكن أن يكون الإجراء في إحدى الحالات التالية:

- جديدة: أُنشئ الإجراء.
- قيد التنفيذ: يجري تنفيذ تعليمات الإجراء.
- تنتظر: ينتظر الإجراء وقوع حدث ما.
- جاهزة: ينتظر الإجراء المعالجة.
- منتهية: أنهى الإجراء التنفيذ.

في لحظة معينة، يوجد إجراء واحد فقط أثناء التنفيذ في أي معالج، بينما يمكن أن توجد عدة إجراءات جاهزة وتنتظر.



4. كتلة تحكم الإجراء

يُمثّل كل إجراء في نظام التشغيل بكتلة تحكم الإجراء (PCB)، حيث تحتوي على عدة معلومات تتعلق بالإجراء:

- حالة الإجراء: وهي واحدة من الحالات: جديدة، جاهزة، قيد التنفيذ، تنتظر، منتهية، ...
- عدّاد البرنامج: يدل العدّاد على عنوان التعليمة التالية التي يجب تنفيذها.
- سجلات وحدة المعالجة: تتضمن المراكمات، سجلات الفهرس، مؤشرات المكدس، والسجلات ذات الاستخدام العام.
- معلومات جدولة وحدة المعالجة: تتضمن أولوية الإجراء، مؤشرات إلى أرتال الجدولة، ومعلومات أخرى (سنراها لاحقاً).
- معلومات إدارة الذاكرة: تتضمن معلومات عن الذاكرة مثل قيمة سجلي القاعدة والحدّ.
- معلومات المحاسبة: تتضمن معدل استخدام وحدة المعالجة، الزمن الحقيقي المستخدم، ...
- معلومات حالة الدخل/الخرج: تتضمن لائحة من تجهيزات الدخل/الخرج اللازمة للإجراء.

5. جدولة الإجراءات

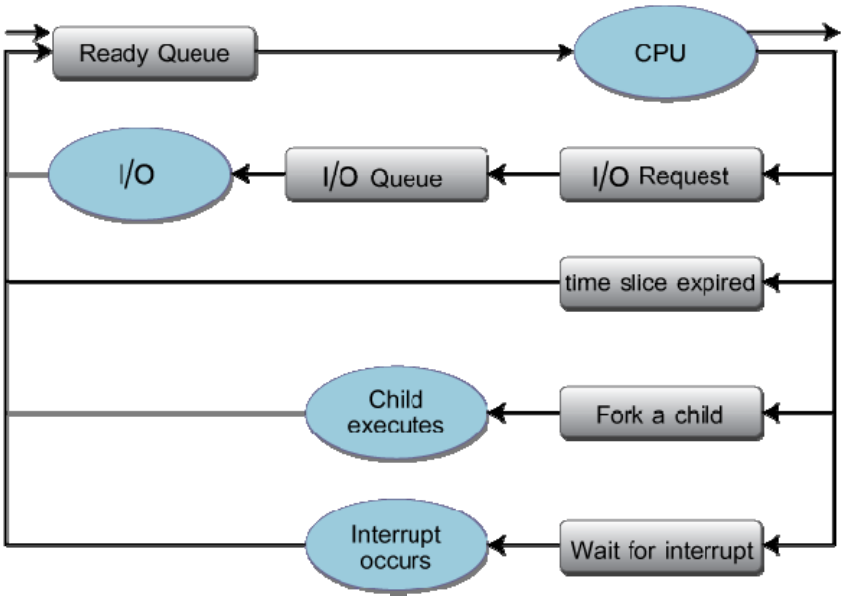
الغرض من تعدّد الإجراءات هو زيادة استخدام وحدة المعالجة، وأن يكون هنالك إجراء يجري تنفيذه في كل وقت.
من أجل جدولة الإجراءات ضمن نظام التشغيل يجري تعريف مجموعة من الأرتال:

- رتل الأعمال: يتألف من مجموعة الإجراءات الموجودة في النظام.
- رتل الجاهزات: يتألف من الإجراءات الموجودة في الذاكرة الرئيسية (RAM) الجاهزة للتنفيذ وبحالة انتظار.
- رتل التجهيزات: لكل تجهيزة دخل/خرج هنالك رتل يحوي الإجراءات التي تنتظر الحصول على هذه التجهيزة.

توضع الإجراءات عند دخولها إلى النظام في رتل الأعمال بانتظار تنفيذها، عندما تُخصص وحدة المعالجة عملها لإجراء معين، فإنها تنفذه لفترة معينة، ثم تتوقف أو تُقاطع أو تنتظر وقوع حدث معين.

عملية دخل/خرج مثلاً:

- يطلب الإجراء عملية دخل/خرج، عندها تنتقل إلى رتل التجهيزة بانتظار الحصول عليها، وبعد انتهاء عملية الدخل/الخرج، ينتقل الإجراء مجدّداً إلى رتل الجاهزات بانتظار إكمال تنفيذها.
- يُنشئ الإجراء إجراءاً فرعياً جديداً، ثم ينتظر انتهاء الإجراء الفرعي.
- يُقاطَع الإجراء فيجري إيقاف تنفيذه ويُعاد ثانيةً إلى رتل الجاهزات.

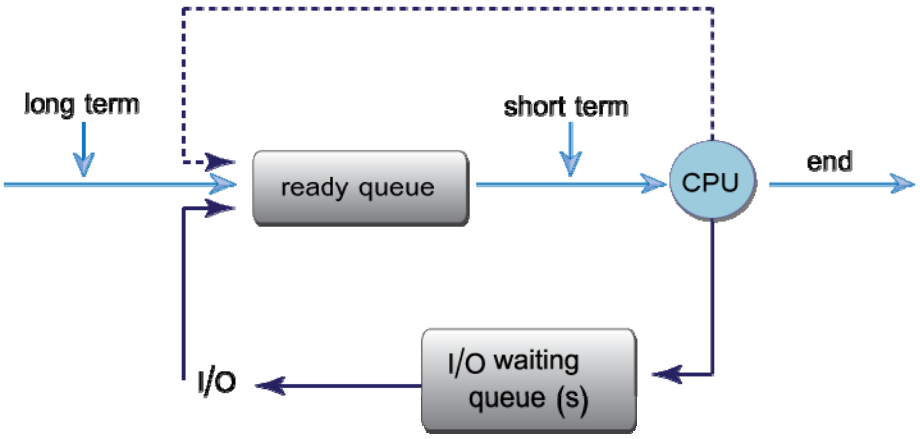


6. المجدولات

تنتقل الإجراءات بين أرتال الجدولة طوال فترة وجودها، ويجب أن يختار نظام التشغيل الإجراءات من الأرتال بطريقة ما، وهذا ما يتم إجرائه من خلال المجدول.

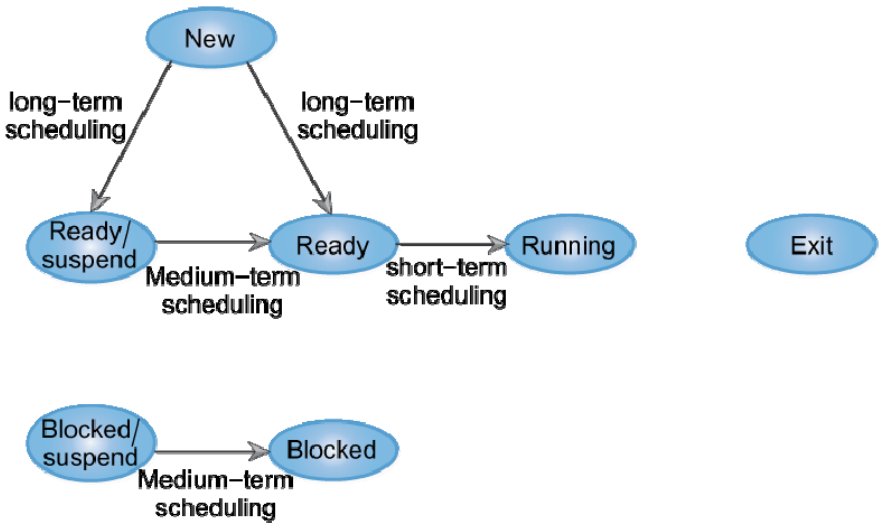
هنالك نوعان من المجدولات:

- المجدول طويل الأمد (مجدول الأعمال): يختار من بين مجموعة الإجراءات المراد تنفيذها، والموجودة على القرص الصلب، وذلك من أجل تحميلها في الذاكرة لبدء تنفيذها.
- المجدول قصير الأمد (مجدول وحدة المعالجة): يختار إجراءاً من بين الإجراءات الجاهزة للتنفيذ، وذلك لكي يسند إليه وحدة المعالجة.



الفرق الأساسي بين المجدولين هو في تواتر تنفيذهما، فالمجدول قصير الأمد ينتخب إجراءاً جديداً لوحدة المعالجة بتواتر كبير (يُستخدم مرة على الأقل كل 100 ميلي ثانية)، بينما ينفذ المجدول طويل الأمد بتواتر أقل بكثير (من مرتبة دقائق)، فهو غالباً يُستدعى عندما يُغادر إجراء ما النظام.

يضاف في بعض نظم التشغيل (خاصة نظم المشاركة بزمن المعالج)، مستوى متوسط من الجدولة (المجدول متوسط الأمد)، حيث يعتمد على إزالة/إعادة الإجراءات من/إلى الذاكرة أي التبديل بين الإجراءات المنفذة والمراد تنفيذها، من أجل تحسين مزيج الإجراءات التي يجري تنفيذها.


New
long-term scheduling
long-term scheduling
Ready/ suspend
Ready
Running
Exit
Medium-term scheduling
short-term scheduling
Blocked/ suspend
Blocked
Medium-term scheduling

7. تبديل السياق

يمكن وصف معظم الإجراءات على أنها مقيدة بالدخل/الخرج أو مقيدة بوحدة المعالجة.

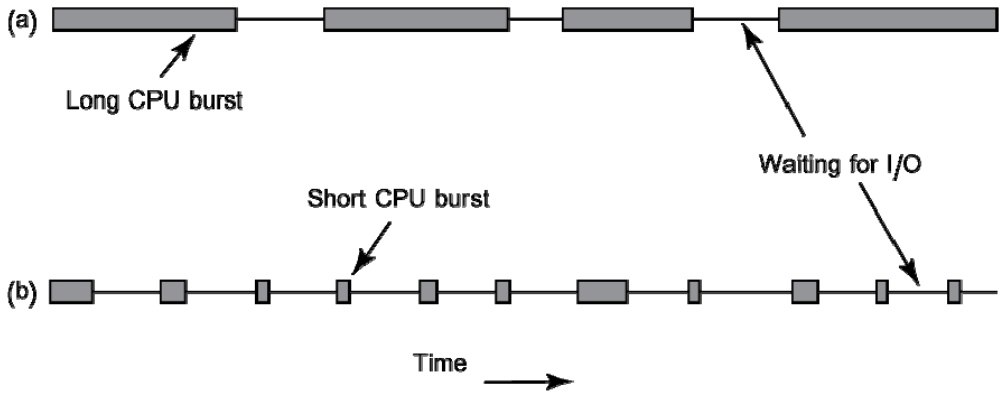
الإجراء المقيد بالدخل/الخرج: هو الإجراء الذي يمضي وقتاً أكثر في عمليات الدخل/الخرج، من الوقت الذي يمضيه في الحسابات.

الإجراء المقيد بوحدة المعالجة: هو الإجراء الذي يستغل معظم وقته في الحسابات.

لكي يجري انتقال وحدة المعالجة من إجراء إلى إجراء آخر، نحتاج إلى حفظ حالة الإجراء الأول، وتحميل الحالة المخزنة للإجراء الثاني، وهذا ما يُسمّى تبديل السياق.

يُمثّل سياق الإجراء بكتلة تحكم الإجراء (PCB)، ويتضمن قيمة سجلات وحدة المعالجة، وحالة الإجراء، ومعلومات إدارة الذاكرة.

كلما كان نظام التشغيل معقداً، كلما كانت كمية العمل اللازمة خلال تبديل السياق أكبر.



(a) A CPU-bound process
الإجراء المقيد بوحدة المعالجة

(b) An I/O-bound process
الإجراء المقيد بالدخل/الخرج

8. إنشاء الإجراءات

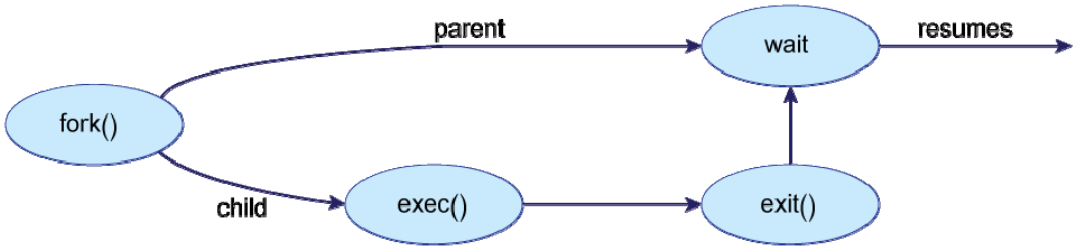
تستطيع الإجراءات أن تنشئ أثناء تنفيذها إجراءات جديدة بواسطة استدعاء النظام (create process)، حيث يُسمى الإجراء الذي يقوم بعملية الإنشاء بالإجراء الأب، بينما تسمى الإجراءات الجديدة بالأبناء.

يمكن للإجراءات الأبناء أن تنشئ إجراءات جديدة أيضاً، وبذلك تتكون شجرة من الإجراءات.

يحتاج الإجراء لإتمام مهمته إلى موارد محددة، لذلك عندما يقوم إجراء بإنشاء إجراء فرعي، يمكن أن يحصل الإجراء الفرعي على موارده مباشرةً من نظام التشغيل، أو يمكن أن يكون مقيداً بمجموعة جزئية من الموارد المتاحة للإجراء الأب، كما يمكن أن يلجأ الإجراء الأب إلى تجزئة أو مشاركة موارده بين الإجراءات الأبناء.

عندما يُنشئ الإجراء إجراءاً جديداً، يكون لدينا احتمالين لتنفيذهما:

- يتابع الإجراء الأب تنفيذه على التساير مع الإجراءات الأبناء الموافقة لها
- ينتظر الإجراء الأب انتهاء بعض أو كل الإجراءات الأبناء التابع له
- هنالك احتمالان بخصوص فضاء عنونة الإجراءات الجديدة
- الإجراء الابن وهو نسخة طبق الأصل عن الإجراء الأب
- الإجراء الابن يملك برنامجاً مختلفاً عن الإجراء الأب



Process Creation

9. إنهاء الإجراءات

ينتهي الإجراء عندما يُنهي تنفيذ آخر عبارة لديه، ويُطلب من نظام التشغيل أن يحذفه بواسطة استدعاء النظام (exit)، عند ذلك يمكن أن يعيد الإجراء معطيات إلى الإجراء الأب.

هنالك ظروف إضافية يمكن أن تسبب الانتهاء، حيث يمكن لإجراء أب أن يوقف أحد أبنائه عن طريق استدعاء النظام (abort)، وذلك لأحد الأسباب التالية:

- تخطي الإجراء الإبن الاستخدام المسموح به لبعض الموارد المخصصة له.
- لم تعد المهمة المسندة إلى الإجراء الإبن مطلوبة.
- إنهاء الإجراء الأب لتنفيذه، ونظام التشغيل لا يسمح للأبناء بالاستمرار بعد انتهاء تنفيذ الإجراء الأب.

10. الإجراءات المتعاونة

إن الإجراءات التي تنفذ في نظام التشغيل هي إما إجراءات مستقلة أو إجراءات متعاونة.

تكون الإجراءات مستقلة إذا كانت لا تؤثر أو تتأثر بالإجراءات الأخرى التي تنفذ في النظام، بينما تكون الإجراءات متعاونة إذا كانت تستطيع التأثير أو التأثر في الإجراءات الأخرى التي تنفذ في النظام.

هنالك عدة أسباب لتوفير بيئة تسمح للإجراءات بالتعاون فيما بينها:

- التشارك في المعلومات: عدة مستخدمين مهتمين بمعلومات مشتركة.
- تسريع الحساب: من أجل تسريع مهمة معينة، يجب أن نجزئها إلى مهمات جزئية تُنفذ على التوازي.
- الاجتزائية: بناء النظام اجتزائياً من خلال تقسيم وظائف النظام إلى إجراءات.
- الملاءمة: وجود عدة إجراءات يريدها المستخدم أن تعمل في وقت واحد.

11. الاتصال بين الإجراءات

يمكن للإجراءات المتعاونة الاتصال في ما بينها في بيئة ذات ذاكرة مشتركة، وذلك من خلال صِوان (buffer) مشترك. كذلك يمكن لنظام التشغيل تحقيق الاتصال بين الإجراءات من خلال عدة طرق، وأفضل هذه الطرق هو نظام التراسل، الذي يسمح للإجراءات بالاتصال فيما بينها دون الحاجة إلى معطيات مشتركة، حيث يوفر عمليتين أساسيتين للإجراء إرسال واستقبال (send, receive)، ولكن لكي يجري التراسل بين الإجراءين يجب تعريف وصلة اتصال بينهما.

وهنالك عدة أنواع لوصلة الاتصال:

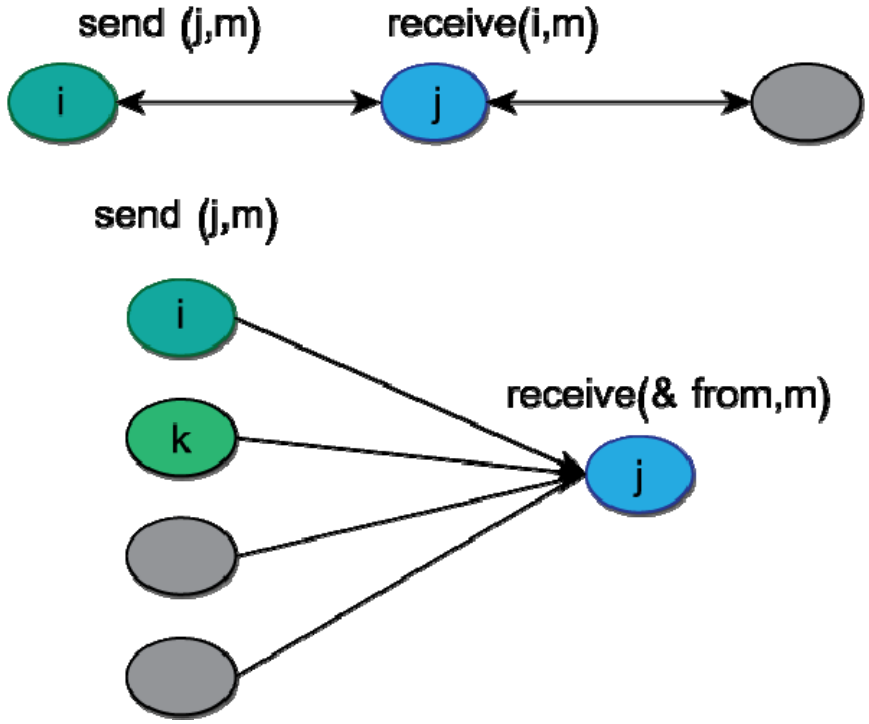
- اتصال مباشر أو لا مباشر.
- اتصال متناظر أو لا متناظر.
- صِوان ضمني أو صريح.
- إرسال نسخة أو مرجع.
- طول الرسائل متغيير أو ثابت.

12. الاتصال المباشر بين الإجراءات

يتعين على كل إجراء راغب في الاتصال أن يحدّد اسم المستقبل أو المرسل صراحةً.

وتكون توابع الإرسال والاستقبال بين الإجراءات:

- Send(P, message) إرسال الرسالة message إلى الإجراء P.
- Receive(id, message) استقبال الرسالة message من الإجراء id (في نمط الاتصال المتناظر يجب تحديد id الإجراء المرسل، بينما لا حاجة لتحديده في النمط اللامتناظر).



خصائص وصلة الاتصال:

تتمتع وصلة الاتصال في هذا النمط بالخصائص التالية:

- يحتاج كل إجراء إلى معرفة هوية الإجراء الآخر.
- تربط الوصلة بين إجراءين.
- هنالك وصلة واحدة بين كل زوج من الإجراءات.

يتميز هذا النمط من الاتصال بتناظر العنونة، إذ يجب أن يسمي إجراء المُرسِل وإجراء المستقبِل كلٌ منهما الطرف الآخر، كما يوجد بديل لهذا الأسلوب يستخدم عنونة لا متناظرة، حيث يسمي المرسِل فقط المرسَل إليه، بينما لا يقوم المستقبِل بتسمية المرسِل.

إن سيئة هذا النوع من الاتصال هو في حال تغيير اسم أحد الإجراءات، حيث نحتاج إلى تغيير هذا الاسم في جميع الإجراءات الأخرى التي تتصل معه.

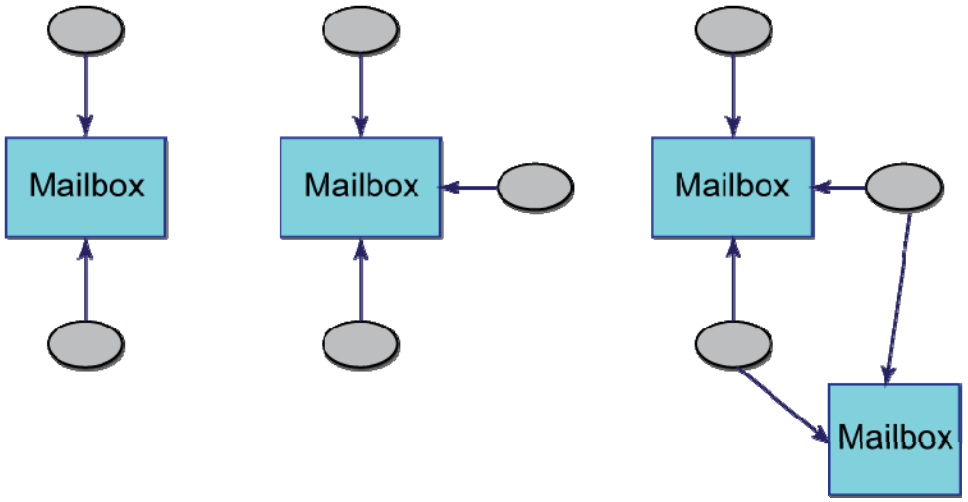
13. الاتصال اللامباشر بين الإجراءات

يجري إرسال واستقبال الرسائل من صناديق البريد أو من البوابات، حيث يعامل صندوق البريد على أنه غرض، توضع فيه الرسائل وتؤخذ بواسطة الإجراءات.

لكل صندوق بريد هوية مميزة، ويمكن لإجراءين الاتصال فيما بينهما عند وجود صندوق بريد مشترك بينهما.

وتكون توابع الإرسال والاستقبال بين الإجراءات:

- Send(A, message) إرسال الرسالة message إلى صندوق البريد A.
- Receive(A, message) استقبال الرسالة message من صندوق البريد A.



خصائص وصلة الاتصال:

- تنشأ وصلة بين كل زوج من الإجراءات، شريطة أن يكون بينهما صندوق بريد مشترك.
- يمكن أن ترتبط الوصلة بإجراءين أو أكثر.
- يمكن أن يوجد عدد من الوصلات المختلفة بين كل زوج من الإجراءات المتصلة، بحيث تقابل كل وصلة منها صندوق بريد واحد.

سؤال: في حال تشارك عدة إجراءات في صندوق البريد نفسه، فأي من هذه الإجراءات سوف يستلم الرسالة القادمة إلى الصندوق؟

إن هذا يعتمد على الحل المختار:

- السماح بأن ترتبط وصلة ما بإجراءين على الأكثر.
- السماح بإجراء واحد على الأكثر بتنفيذ عملية receive في وقت واحد.
- السماح للنظام بأن ينتقي عشوائياً أحد الإجراءات.

14. التزامن بين الإجراءات المتصلة

يمكن أن يكون الاتصال بين الإجراءات من خلال عمليتي الإرسال والاستقبال، بعدة أشكال (معيق أو غير معيق، متزامن أو غير متزامن):

1. إرسال معيق: يُعاق الإجراء المرسِل إلى أن يتسلم الإجراء المُستقبل الرسالة.
2. إرسال غير معيق: يستطيع الإجراء المرسِل متابعة عمله بعد إرسال الرسالة.
3. استقبال معيق: يُعاق الإجراء المستقبِل حتى تصبح الرسالة متوفرة.
4. تسلم غير معيق: يقوم الإجراء المستقبل بعملية الاستقبال سواء كانت الرسالة موجودة أو لا.

15. رتل الانتظار

بين الإجراءات مباشر أو غير مباشر، فإن الرسائل المتبادلة تمر برتل مؤقت (buffer).

يوجد ثلاث أنواع لهذا الرتل:

1. سعة صفرية: سعة الرتل صفرية، أي لا يمكن لأي رسالة الانتظار في الرتل، ولذلك يجب على المرسل انتظار استقبال الرسالة من قبل المستقبل.
2. سعة محدودة: سعة منتهية للرتل (n)، وبذلك يمكن لـn رسالة على الأكثر أن تبقى في الرتل.
3. سعة لامحدودة: سعة غير محدودة للرتل، وبذلك يمكن لأي عدد من الرسائل أن ينتظر في الرتل.

التمارين:

1. الإجراء هو أي برنامج أثناء التنفيذ:
   A. صح
   B. خطأ

2. يتألف الإجراء من:
   A. رماز البرنامج
   B. مكدس
   C. مقطع معطيات
   D. جميع الإجابات صحيحة

3. يمكن أن يكون الإجراء في إحدى الحالات التالية:
   A. جديدة
   B. تنتظر
   C. جاهزة
   D. منتهية
   E. جميع الإجابات صحيحة

4. يُمثّل كل إجراء في نظام التشغيل بكتلة تحكم الإجراء (PCB)، حيث تحتوي على عدة معلومات تتعلق بالإجراء:
   A. عداد البرنامج
   B. معلومات المحاسبة
   C. معلومات جدولة وحدة المعالجة
   D. جميع الإجابات صحيحة

5. الغرض من تعدّد الإجراءات هو زيادة استخدام وحدة المعالجة، وأن يكون هنالك إجراء يجري تنفيذه في كل وقت:
   A. صح
   B. خطأ

6. من أجل جدولة الإجراءات ضمن نظام التشغيل يجري تعريف مجموعة من الأرتال:
   A. رتل التجهيزات
   B. رتل الجاهزات
   C. رتل الأعمال
   D. جميع الإجابات صحيحة

7. يختار إجراءاً من بين الإجراءات الجاهزة للتنفيذ:
   A. المجدول قصير الأمد
   B. المجدول طويل الأمد
   C. رتل الأعمال
   D. جميع الإجابات خاطئة

8. يمكن لإجراء أب أن يوقف أحد أبنائه عن طريق استدعاء النظام (abort)، وذلك لأحد الأسباب التالية:
   A. تخطي الإجراء الإبن الاستخدام المسموح به لبعض الموارد المخصصة له.
   B. لم تعد المهمة المسندة إلى الإجراء الإبن مطلوبة.
   C. إنهاء الإجراء الأب لتنفيذه، ونظام التشغيل لا يسمح للأبناء بالاستمرار بعد انتهاء تنفيذ الإجراء الأب.
   D. جميع الإجابات صحيحة

9. هنالك عدة أسباب لتوفير بيئة تسمح للإجراءات بالتعاون فيما بينها:
   A. التشارك في المعلومات
   B. الاجتزائية
   C. الملاءمة
   D. جميع الإجابات صحيحة

10. يمكن للإجراءات المتعاونة الاتصال في ما بينها في بيئة ذات ذاكرة مشتركة، وذلك من خلال صِوان (buffer) مشترك:
   A. صح
   B. خطأ

11. هنالك عدة أنواع لوصلة الاتصال:

A. إرسال نسخة أو مرجع.

B. اتصال متناظر أو لا متناظر.

C. صِوان ضمني أو صريح.

D. جميع الإجابات صحيحة

12. يتميز هذا النمط من الاتصال بتناظر العنونة:

A. الاتصال غير المباشر بين الاجراءات

B. الاتصال المباشر

C. الاتصال الضمني

D. الاتصال الصريح

13. يمكن أن يكون الاتصال بين الإجراءات من خلال عمليتي الإرسال والاستقبال، بعدة أشكال (معيق أو غير معيق، متزامن أو غير متزامن):

A. صح

B. خطأ

رقم التمرين	الإجابة الصحيحة
.1	(A)
.2	(D)
.3	(E)
.4	(D)
.5	(A)
.6	(D)
.7	(A)
.8	(D)
.9	(D)
.10	(A)
.11	(D)
.12	(B)
.13	(A)

أنظمة الملفات

الكلمات المفتاحية:

نظام الملفات، جدول تخصيص الملفات، مدخل، قائمة الواصفات، مجلد، كتلة معطيات، عناقيد المعطيات، مؤشر إلى ملف، مؤشر إلى مجلد، منطقة جدول الملفات الرئيسي، جدول الملفات الرئيسي، الملفات المترفعة، قطاع معطل، المجلد الجذر، i-node.

ملخص:

يركز هذا الفصل على التعرف إلى أنظمة الملفات، وعلى بنيتها وأنواعها.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- التعرف على نظام الملفات (وظيفته، مستوياته، أشهر أنواعه).
- نظام ملفات ISO 9660.
- نظام الملفات CP/M.
- نظام ملفات MS-DOS.
- جدول تخصيص الملفات FAT.
- نظام ملفات Windows 98.
- نظام ملفات UNIX V7.
- نظام ملفات Windows 2000.
- نظام الملفات NTFS، بنية جدول الملفات الرئيسي، آلية حماية وضغط وتشفير الملفات.
- اختيار نظام الملفات.
- ميزات NTFS.
- ميزات FAT.
- التحويل بين أنظمة الملفات.

1. أقراص التخزين

يجب أن تحقق أقراص التخزين الخاصتين التاليتين:

- إمكانية تخزين الملفات على القرص، وإعادة كتابتها في الموقع نفسه، بالإضافة إلى إمكانية قراءة كتلة من القرص، تعديلها، وكتابتها مجدداً في نفس المكان.
- الوصول المباشر إلى أي كتلة من المعطيات على القرص، وهذا ما يعطي سهولة في الوصول إلى معطيات الملف، إما بشكل تسلسلي أو بشكل عشوائي.

2. نظام الملفات

يَستخدِم نظام التشغيل ما يسمى نظام الملفات، من أجل تأمين وصول ملائم وفعّال إلى القرص، والسماح بتخزين المعطيات، وتحديد مكانها، بالإضافة إلى استرجاعها بسهولة.

هنالك مشكلتان أساسيتان لنظام الملفات:

- كيفية ظهور نظام الملفات بالنسبة للمستخدم، وهذا يتضمن تعريف الملف، تعريف واصفاته، وتحديد العمليات المتاحة عليه، بالإضافة إلى توصيف بنية المجلدات، وتوصيف طريقة تنظيم الملفات.
- الخوارزميات وبنى المعطيات التي يجب خلقها، لتحقيق التقابل بين نظام الملفات المنطقي، وأجهزة التخزين الفيزيائية.

3. مستويات نظام الملفات

هنالك خمس مستويات لنظام الملفات:

- التحكم بالدخل/الخرج وتجهيزات التخزين: عبارة عن الطبقة الدنيا لنظام الملفات، تهتم بعملية نقل الملفات من نظام القرص إلى الذاكرة.
- نظام الملفات الأساسي: يقوم بإعطاء تعليمات عامة إلى قرص التخزين المناسب، من أجل قراءة وكتابة كتل فيزيائية على القرص.
- جزء تنظيم الملف: يهتم بالملف وكتله المنطقية والفيزيائية.
- نظام الملفات المنطقي: يستخدم بنية المجلد، من أجل تزويد جزء تنظيم الملف، بالمعلومات التي يحتاجها، كما أنه مسؤول عن حماية وأمن الملفات.

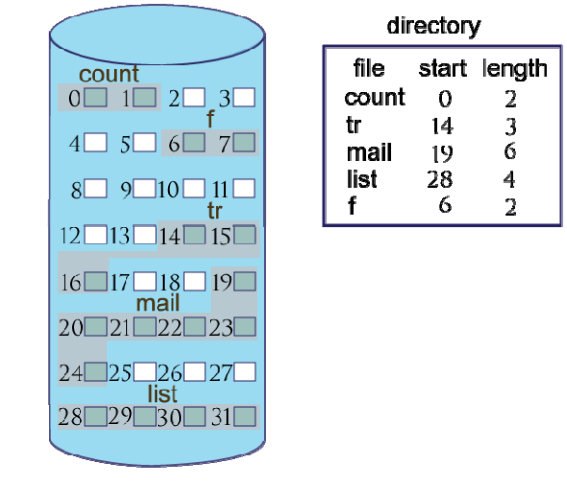
4. طرق حجز كتل الملفات

هنالك ثلاث طرق لحجز كتل المعطيات في نظام الملفات:

1. حجز مستمر:

يحتاج كل ملف إلى حجز مجموعة متتالية من كتل المعطيات من أجل تخزينه. وبالتالي فإن عملية القراءة من القرص تتم بشكل تسلسلي أيضاً.

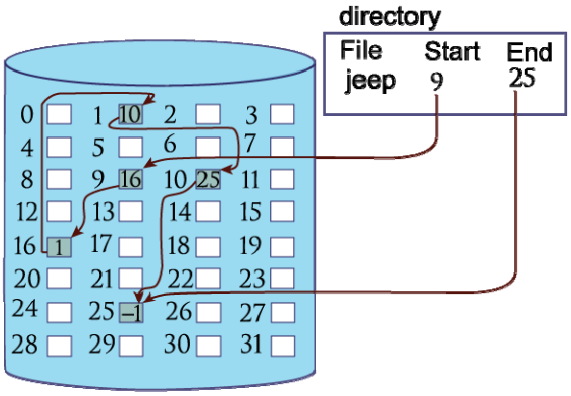
المشكلة الأساسية في هذه الطريقة هي إيجاد عدد كافي من الكتل المتتالية لتخزين الملف.



Contiguous allocation
حجز مستمر

2. حجز مترابط:

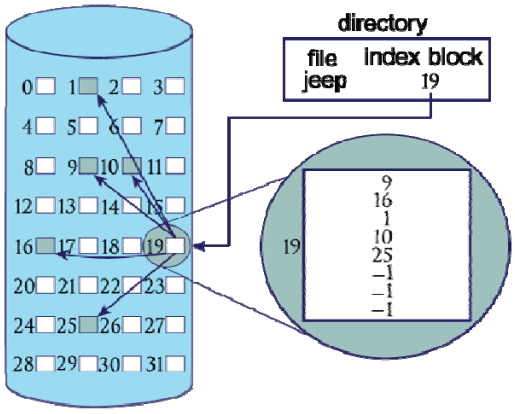
كل ملف عبارة عن مجموعة من الكتل المترابطة، حيث يحتوي المجلد على مؤشر إلى أول وآخر كتلة في الملف، كما أن كل كتلة تشير إلى الكتلة التي تليها.



Linked allocation
حجز مترابط

3. حجز مفهرس:

يوجد لكل ملف كتلة فهرس تحوي على عناوين الكتل الخاصة بهذا الملف، وبالتالي فإن الوصول إلى أي كتلة، يتم بشكل مباشر من خلال كتلة الفهرس، وهذا يسرع عملية الوصول.



Indexed allocation
حجز مفهرس

5. أنظمة الملفات الشهيرة

أشهر أنظمة الملفات:

- نظام الملفات ISO 9660 المستخدم لتنظيم الـCD-ROM.
- نظام الملفات CP/M (Control Program for Microcomputer) الخاص بالحواسب الشخصية.
- نظام الملفات الخاص بنظام التشغيل MS-DOS ونظام Windows 95 وWindows 98، حيث يعتمد نظام الملفات على جدول تخصيص الملفات FAT (File Allocation Table).
- نظام الملفات UNIX V7 الخاص بنظام التشغيل UNIX.
- نظام الملفات NTFS (NT File System) أي نظام الملفات الشبكي، تم تطويره من أجل الأقراص الصلبة التي تتعامل مع نظام التشغيل Windows NT، ولاحقاً مع نظم التشغيل Windows 2000 وWindows XP.

6. نظام ملفات الـCD-ROM

إن بنية الـCD-ROM مختلفة عن بنية القرص الصلب، فالمسارات الموجودة في القرص الصلب غير موجودة في الـCD، إنما بنية الـCD عبارة عن حلزون واحد مستمر، ومُقسّم إلى كتل منطقية تدعى قطاعات مؤلفة من 2352 بايت.

كما أنه لا يوجد حاجة لترك فراغات بين الكتل لأنه لا يمكن حذف أو إضافة كتل جديدة إلى الـCD بعد الكتابة عليه.

أشهر نظام ملفات مستخدم للـCD-ROM هو النظام ISO 9660.

يتم تنظيم الحجم ضمن الـCD كما يلي:

- يبدأ كل CD بـ16 كتلة محجوزة، لا تستخدم من قبل نظام الملفات ISO 9660، إنما تستخدم فقط من قبل مُصنِّع الـCD وذلك لتخزين برنامج إقلاع (لجعل الحاسب قادر على الإقلاع من الـCD)، أو من أجل أغراض أخرى.
- يأتي بعد الـ16 كتلة، كتلة واحدة (تدعى واصف الحجم الرئيسي) تحوي على معلومات عامة عن الـCD، من ضمنها مميز النظام (32 بايت)، مميز الحجم (32 بايت)، مميز الناشر (128 بايت)، بالإضافة إلى مميز تحضير المعطيات (128 بايت).
- يحتوي واصف الحجم الرئيسي على: اسم ثلاث ملفات يمكن أن تحوي معلومات عن الـCD، حجم الكتلة المنطقية، عدد كتل المعطيات، زمن انتهاء صلاحية الـCD، بالإضافة إلى مدخل يؤشر إلى المجلد الجذر ضمن الـCD.
- من خلال المجلد الجذر يمكن الوصول إلى بقية الملفات على الـCD.

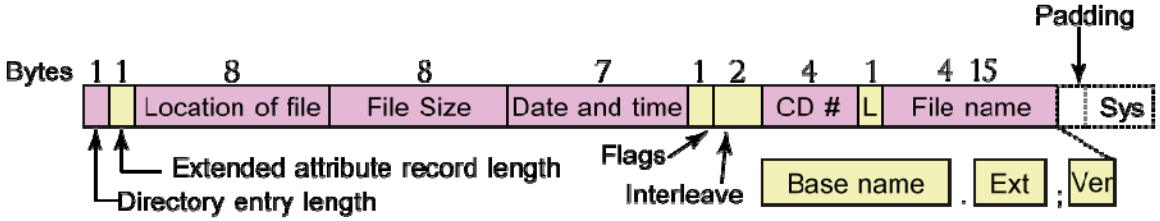
7. بنية المجلد في نظام ISO 9660

يتألف المجلد الجذر وجميع المجلدات الأخرى من عدد متغير من المداخل، حيث يحتوي المدخل الأخير منها على بت للدلالة على نهاية هذه المداخل. كما أن مداخل المجلد نفسه متغيرة الطول، فكل مدخل يتألف من 10 إلى 12 حقل، حيث يشير البايت الأول من المدخل إلى طول هذا المدخل.

تحتوي حقول المدخل على:

- الكتلة الأولى من الملف، وبما أن الملفات مخزنة بشكل تسلسلي على الـCD، فإن الكتلة الأولى من الملف تعبر عن مسار الملف بشكل كامل.
- الوقت والتاريخ.
- حقل للمؤشرات، للتمييز بين مدخل ملف ومدخل مجلد، للدلالة على المدخل الأخير، بالإضافة إلى مؤشرات أخرى.
- حقل للدلالة إلى الـCD الذي يحتوي الملف، حيث يمكن أن يحوي CD ما على مجلد يؤشر إلى ملف موجود في CD آخر.
- حقل لحجم الملف.
- حقل لاسم الملف، حيث يتألف من الاسم الأساسي، ثم (.)، ومن ثم لاحقة الملف، ثم (;)، يليها رقم ثنائي.

بالنسبة لمداخل المجلد فيتم ترتيبها ترتيب أبجدي، ماعدا المدخلين الأول والثاني، فالمدخل الأول يشير للمجلد نفسه، والمدخل الثاني يشير للمجلد الأب له.



ملحق الرسم	
File location	موضع الملف
File size	حجم الملف
Date and time	الوقت والزمن
CD#	رقم الـCD
L	حجم الملف
File name	اسم الملف
Base name	الاسم الأساسي
Ext	اللاحقة
Directory entry length	طول مدخل المجلد

8. نظام الملفات CP/M

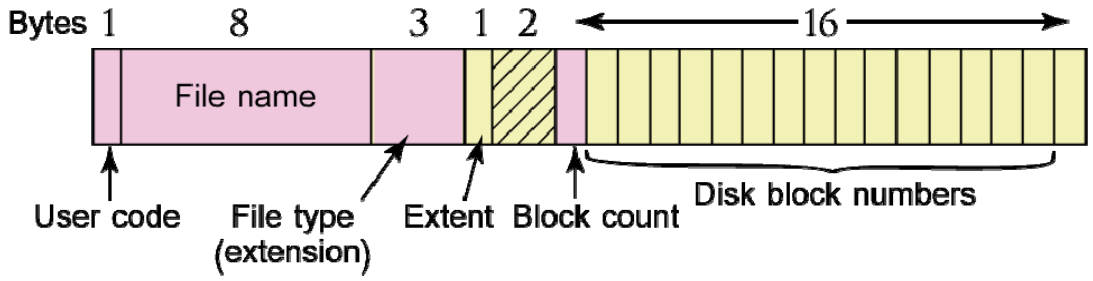
نظام الملفات الخاص بالحواسب الشخصية (CP/M Control Program for Microcomputers).

يتألف نظام الملفات من مجلد واحد ثابت الحجم، يحتوي على مداخل ثابتة الحجم (32 بايت)، حيث أن جميع الملفات في النظام موجودة ضمن هذا المجلد.

يقوم نظام الملفات CP/M عند طلب الوصول إلى ملف معين، بالبحث ضمن المجلد عن مدخل هذا الملف، وبعد إيجاده، يقوم بأخذ أرقام كتل الملف منه، وذلك ليتمكن من الوصول إليها.

يتألف المدخل من الحقول التالية:

رمز المستخدم والذي يعبر عن مالك الملف، اسم الملف، لاحقة الملف، عدد كتل الملف.



ملحق الرسم	
User code	رمز المستخدم
File name	اسم الملف
Extension	اللاحقة
Block count	عدد الكتل
Disk block numbers	أرقام كتل القرص

9. نظام ملفات MS-DOS

يشبه نظام ملفات الـMS-DOS نظام ملفات CP/M، فالنسخ الأولى منه كانت تحوي على مجلد واحد يضم جميع الملفات. أما مع النسخ الجديدة ومع توسع وظائف نظام الملفات، أصبحت بنية النظام هرمية، مع إمكانية تداخل المجلدات ضمن البنية. أي أنه يمكن للمجلد الجذر أن يحوي مجلدات جزئية، وكل مجلد جزئي يمكن أن يحوي مجلدات أخرى وهكذا. وبالتالي أصبحت البنية عبارة عن شجرة تبدأ بالمجلد الجذر.

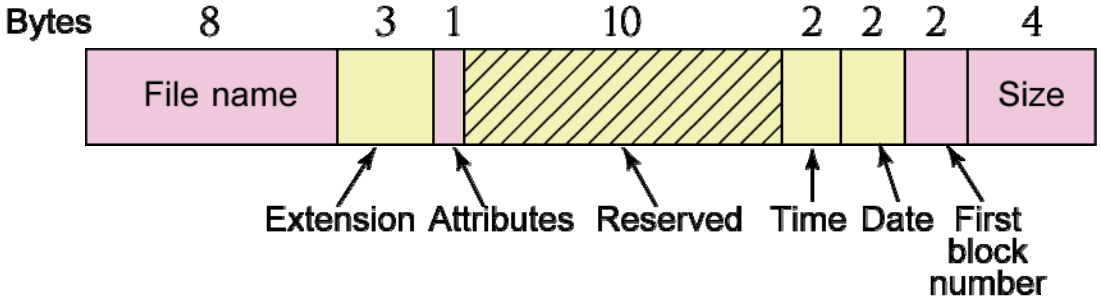
يتم البحث ضمن نظام الملفات عن ملف أو مجلد معين انطلاقاً من مساره، حيث يتم تحليل المسار جزء بجزء حتى الوصول إلى المجلد أو الملف المطلوب.

على الرغم من أن حجم المجلدات ضمن النظام غير ثابت، إلا أن حجم مداخل المجلد ثابت (32 بايت).

يتألف مدخل المجلد من الحقول التالية:

اسم الملف، واصفات الملف (يحتوي على بتات للدلالة أن الملف للقراءة فقط، مؤرشف، ملف مخفي، أو ملف نظام)، زمن وتوقيت البناء، كتلة البداية، بالإضافة إلى حجم الملف.

هنالك أيضاً فرق آخر بين نظام ملفات MS-DOS وCP/M، حيث لا يخزن MS-DOS عناوين الكتل في مداخل المجلد، إنما يتم حفظها في جدول تخصيص الملفات (FAT) ضمن الذاكرة الرئيسية. ويتم الوصول إلى العنصر المحدد من هذا الجدول (الذي يمثل الملف المطلوب)، من خلال رقم الكتلة الأولى من الملف، حيث يخزن هذا الرقم في مدخل المجلد.



ملحق الرسم	
File name	اسم الملف
Extension	اللاحقة
Attributes	الواصفات
Reserved	منطقة محجوزة
Time and date	الوقت والتاريخ
First block number	رقم الكتلة الأولى
Size	الحجم

10. جدول تخصيص الملفات في نظام ملفات MS-DOS

يأتي جدول تخصيص الملفات بثلاث نسخ من أجل نظام MS-DOS:

1. FAT-12
2. FAT-16
3. FAT-32

حيث تعتمد هذه النسخ على عدد البتات المستخدمة في عناوين القرص:

- FAT-12 يستخدم12 بت للعنوان
- FAT-16 يستخدم 16 بت للعنوان
- FAT-32 يستخدم فقط 28 بت للعنوان

كما أن حجم كتل المعطيات من أجل كل جداول FAT هي من مضاعفات 512 بايت.

- فمن أجل نظام FAT-12 وحجم كتلة 512 bytes، يكون الحجم الأعظمي لجزء القرص الصلب الذي يمكن عنونته (212 * 512 bytes) أي حوالي 2MB.
- ومن أجل نظام FAT-16 وحجم كتلة 32 KB، فإن الحجم الأعظمي لجزء القرص الذي يمكن عنونته 2GB. ومن أجل نظام FAT-32 فإن الحجم الأعظمي هو 2TB.
- يبين الجدول التالي الحجم الأعظمي الذي يمكن عنونته، تبعاً لجدول تخصيص الملفات وحجم الكتلة:

حجم الكتلة	FAT-12	FAT-16	FAT-32
0.5KB	2MB		
1KB	4MB		
2KB	8MB	128MB	
4KB	16MB	256MB	1TB
8KB		512MB	2TB
16KB		1024MB	2TB
32KB		2048MB	2TB

11. بنية جدول تخصيص الملفات

تبدأ كل نسخة من جدول تخصيص الملفات ببعض البتات (والتي لها دلالات معينة)، ومن ثم تأتي عناوين عناقيد المعطيات والمخزنة ضمن الجدول أيضاً.

للوصول إلى عنقود المعطيات التالي، تتم قراءة عنوانه من جدول تخصيص الملفات بانزياح معين عن العنقود الحالي.

هنالك بعض القيم الخاصة لبتات الدلالة، مثلاً 0 يعني أن هذا العنقود لا يُستخدم، وقيم عليا تعبر عن نهاية سلسلة العناقيد وعن القطاعات المعطلة. أما بقية القيم الأخرى فهي تشير إلى عناقيد معطيات، وبما أنه لا يمكن أن ينتمي العنقود إلى أكثر من ملف أو مجلد، فإن القيم غير الخاصة للبتات هي قيم فريدة على مستوى جدول تخصيص الملفات.

12. نظام ملفات Windows 98

تستخدم النسخ الأولى من نظام التشغيل windows 95 نظم الملفات FAT-12 و FAT-16، ولكن في نظام windows 98 والنسخ المطورة من نظام windows 95، أصبح نظام الملفات المستخدم هو نظام FAT-32 وذلك لإتاحة الإمكانية للتعامل مع أقراص صلبة حجم أجزاءها أكبر من 2GB.

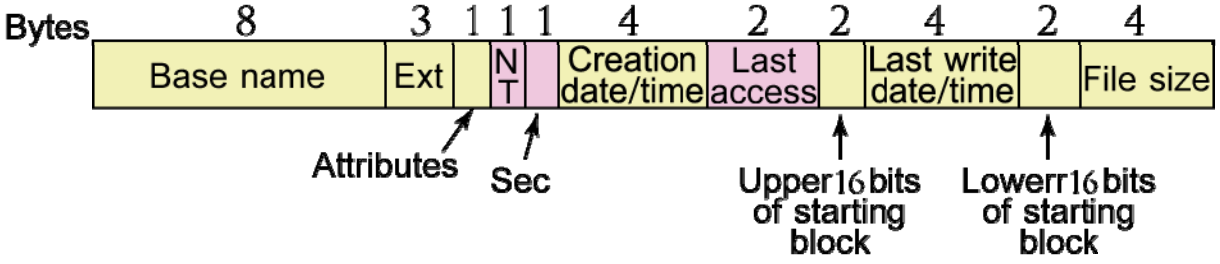
إن بنية مدخل المجلد المستخدمة في نظام windows 98 هي نفسها بنية المدخل في MS-DOS ولكن مع بعض الحقول الإضافية:

الحقل NT: لإظهار أن هذا الملف متوافق مع نظام windows.

الحقل Sec: يقدم هذا الحقل بتات إضافية لحل مشكلة تخزين توقيت اليوم.

الحقل Last Access: يخزن هذا الحقل تاريخ (دون توقيت) آخر تعديل على الملف.

وبما أن رقم الكتل أصبح يمثل على 32 بت في FAT-32، لذلك نحتاج إلى 16 بت إضافية لتخزين الـ16 بت العليا من رقم الكتلة الأولى من الملف.



ملحق الرسم	
Base name	الاسم الأساسي
Ext	اللاحقة
Attributes	الواصفات
Creation date/time	تاريخ/توقيت البناء
Last access	آخر وصول
Upper 16 bits of starting block	الـ16 بت العليا من الكتلة الأولى
Lower 16 bits of starting block	الـ16 بت الدنيا من الكتلة الأولى
Last write date/time	تاريخ/توقيت آخر كتابة
File size	حجم الملف

13. أسماء الملفات في windows 98

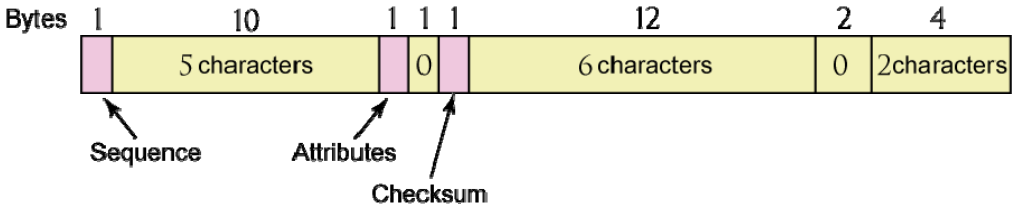
يعطي نظام windows 98 إمكانية استخدام أسماء طويلة للملف، فكيف يحقق هذه التقنية مع المحافظة على التوافق مع طول الأسماء في MS-DOS؟

الحل هو بإعطاء اسمين للملف اسم طويل للتوافق مع windows 98 وآخر للتوافق مع MS-DOS، فمن أجل windows 98 يتم استخدام الاسم الطويل كما هو، ومن أجل MS-DOS يتم أخذ المحارف الستة الأولى من الاسم، تحويلها إلى محارف كبيرة، ومن ثم إضافة تتمة للاسم 1~، وفي حال وجود ملف بهذا الاسم يتم وضع التتمة 2~ وهكذا.

رأينا سابقاً طريقة تخزين اسم الملف في نظام MS-DOS، أما في نظام Windows 98 فلكل مدخل لاسم طويل يستخدم 13 محرف للتعبير عنه، حيث يتم تخزينها بطريقة معكوسة، تبدأ بالاسم الأساسي (6 محارف كما في MS-DOS)، ومن ثم مجموعة من المقاطع:

- المقطع Attributes: يستخدم للتمييز بين اسم ملف طويل، واسم ملف في نظام MS-DOS (حيث يحوي هذا الحقل القيمة 0x0F في حال الاسم الطويل، ويتم تجاهل هذه القيمة من قبل نظام MS-DOS).
- المقطع Sequence: لتحديد رقم المقطع، في حال استمرار الاسم على أكثر من مقطع.
- المقطع Checksum: يتم استخدام هذا المقطع لتفادي بعض المشاكل التي تحدث في التسمية، والتحقق من الاسم الموجود.

يستخدم في نظام Windows 98 لكل مدخل اسم طويل، 13 محرف للتعبير عنه، حيث يتم تخزينها بطريقة معكوسة، تبدأ بالاسم الأساسي، ومن ثم مجموعة من المقاطع كما في الشكل التالي:



ملحق الرسم	
Characters	محارف
Sequence	تتالي
Attributes	الواصفات
Checksum	التحقق من المجموع

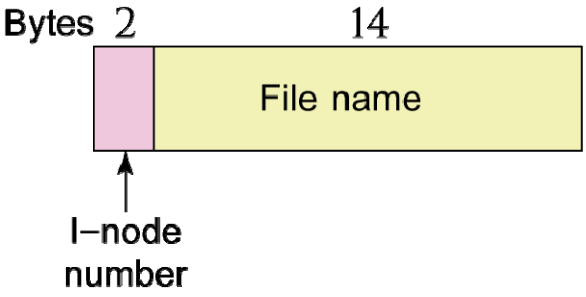
14. نظام ملفات UNIX V7

نظام الملفات عبارة عن شجرة تبدأ بالمجلد الجذر، بالإضافة إلى وصلات إضافية تشكل بيان موجه.

يحوي مدخل المجلد في نظام UNIX، مدخل واحد لكل ملف ضمن المجلد، كما أن المدخل بسيط البنية، حيث يستخدم نظام UNIX بنية i–node.

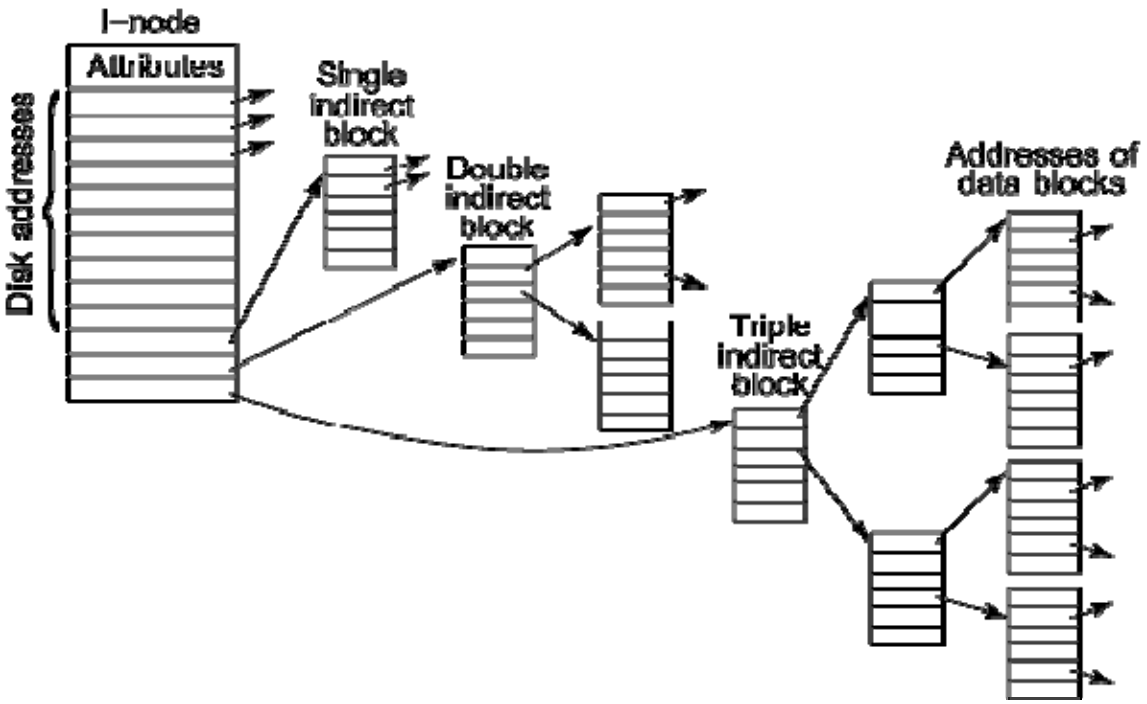
يحوي مدخل المجلد على حقلين: اسم الملف، وعدد العقد i–node لهذا الملف (2 بايت).

تحوي i–node في نظام UNIX على عدد من الواصفات: حجم الملف، ثلاث أزمنة (البناء، آخر تعديل، آخر وصول)، المالك، المجموعة، معلومات الحماية، عدد مداخل المجلدات التي تشير إلى هذه الـi–node.



من أجل الوصول إلى كتل الملف المخزنة على القرص، يتم تخزين الـ10 عناوين الأولى ضمن العقدة i–node نفسها، وبذلك فإذا كان الملف صغير فإن جميع عناوين الكتل التابعة له ستُخزن ضمن العقدة نفسها. أما بالنسبة للملفات الكبيرة فإن أحد العناوين ضمن العقدة، يشير إلى كتلة تدعى "كتلة غير مباشرة وحيدة"، حيث تحوي هذه العقدة بدورها على عناوين كتل على القرص، وبنفس الطريقة يمكن أن تشير إلى كتلة تدعى "كتلة غير مباشرة مزدوجة"، ونفس الأمر يتكرر من أجل هذه الكتلة.

يستخدم نظام UNIX بنية i–node لتنظيم الملفات، كما هو موضح في الشكل:



ملحق الرسم	
File name	اسم الملف
i-node number	رقم العقدة
Attributes	واصفات
Disk addresses	عناوين القرص
Single indirect block	كتلة غير مباشرة وحيدة
Double indirect block	كتلة غير مباشرة مزدوجة
Triple indirect block	كتلة غير مباشرة ثلاثية
Addresses of data blocks	عناوين كتل المعطيات

15. نظام ملفات Windows 2000

يدعم نظام Windows 2000 مجموعة من أنظمة الملفات أهمها FAT-16 وFAT-32 وNTFS، وسنشرح فيما يلي بنية نظام الملفات NTFS. يحتوي كل جزء من القرص يستخدم نظام الملفات NTFS، على ملفات، مجلدات، بالإضافة إلى بنى معطيات أخرى.

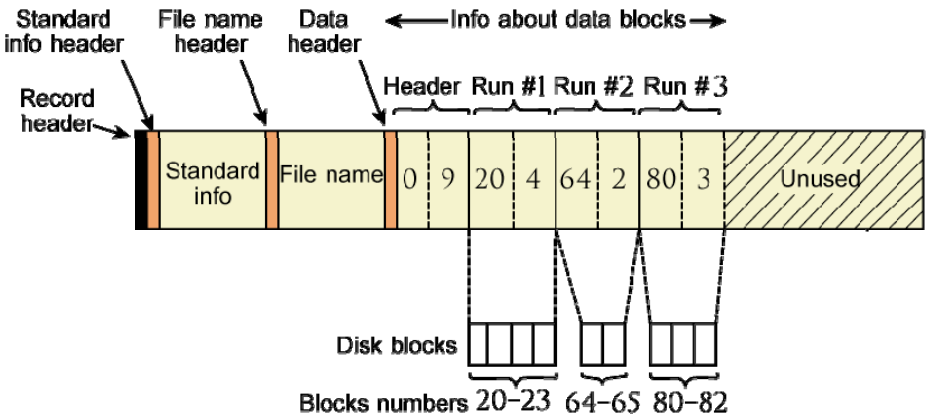
ينظم كل حجم على شكل تتالي من الكتل (عناقيد) ثابتة الحجم (بين 512 بايت و64 كيلوبايت)، حيث يشار إلى الكتلة من خلال انزياحها عن بداية الحجم من خلال عدد بـ64 بت.

بنية المعطيات الأساسية في كل حجم هي جدول الملف الأساسي (MFT)، وهو عبارة عن تتالي خطي من تسجيلات ثابتة الحجم (1 كيلوبايت)، حيث تعبر كل تسجيلة عن ملف أو مجلد.

تحوي كل تسجيلة على واصفات الملف: اسم الملف، الختم الزمني، قائمة عناوين الكتل المؤلفة له. يمكن أن يستخدم أكثر من تسجيلة لتخزين عناوين الكتل في حال كون الملف كبير، وعندها تدعى التسجيلة الأولى بالتسجيلة الأساسية.

إن جدول الملف الأساسي هو عبارة عن ملف أيضاً، ولذلك يمكن أن يخزن في أي مكان ضمن الحجم.

تحجز التسجيلات الـ16 الأولى من MFT، لتخزين الملفات المترفعة لنظام NTFS والتي سنشرحها لاحقاً.



ملحق الرسم	
MTF record	تسجيلة MTF
Record header	ترويسة التسجيلة
Standard info header	ترويسة المعلومات القياسية
Standard info	معلومات قياسية
File name header	ترويسة اسم الملف
File name	اسم الملف
Data header	ترويسة المعطيات
Info about data blocks	معلومات عن كتل المعطيات
Disk blocks	كتل القرص
Blocks numbers	أرقام الكتل

16. البنية الفيزيائية لنظام الملفات NTFS

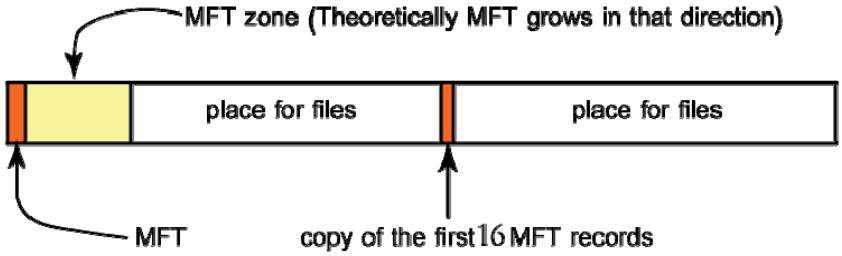
لا يوجد نظرياً حجم أعظمي للأجزاء المنطقية التي يمكن أن يتعامل معها نظام الملفات NTFS، ولكن عملياً فإن حجم الأجزاء متعلق بحجم القرص الصلب.

يَقسِم نظام الملفات NTFS القرص إلى قسمين:

- منطقة جدول الملفات الرئيسي MFT zone: وهي عبارة عن الـ12% الأولى من القرص، وهي منطقة فارغة لا يمكن الكتابة فيها، إنما تبقى فارغة تحسباً لتزايد حجم جدول الملفات الرئيسي (MFT)، وذلك منعاً لتقسيمه.
- منطقة تخزين الملفات: المساحة المتبقية من القرص 88% والمقسمة إلى كتل (عناقيد)، والمخصصة لتخزين الملفات.

يتم استخدام MFT zone بالشكل التالي:

عندما يزداد حجم ملف MFT ولا توجد مساحة كافية لتخزينه، يتم إنقاص حجم MFT zone وإضافة هذه المساحة المقطوعة إلى MFT، وبذلك لا يتم تقسيم MFT.



ملحق الرسم	
منطقة جدول الملفات الرئيسي	MFT zone
جدول الملفات الرئيسي	MFT
مساحة مخصصة للملفات	Place of files

17. بنية جدول الملفات الرئيسي MFT

يعتبر جدول الملفات الرئيسي MFT الملف الأهم في نظام الملفات NTFS، وهو أول الملفات المترفعة في النظام. حيث يمثل المجلد المركزي لجميع الملفات في النظام، وهو مقسم إلى تسجيلات بحجوم ثابتة (عادةً 1 كيلوبايت)، وكل تسجيلة تتبع إلى ملف معين، تحوي كل تسجيلة على جميع المعلومات الخاصة بالملف (الاسم، الحجم، مواقع الأجزاء)، ما عدا المعطيات المخزنة في الملف. في حال عدم كفاية تسجيلة واحدة للملف يتم حجز أكثر من تسجيلة له.

18. أهم واصفات الملف

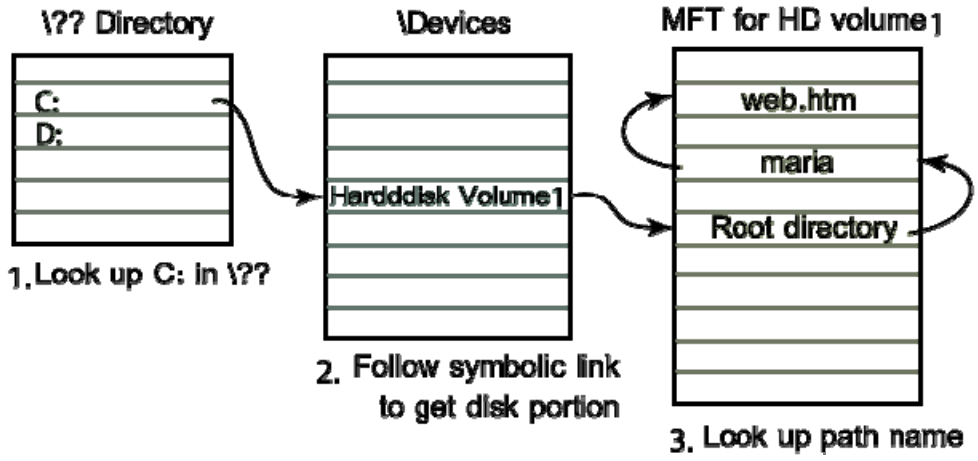
أهم واصفات الملف في نظام الملفات NTFS:

1. معلومات قياسية: مثل عدد الوصلات والختم الزمني.
2. قائمة الواصفات: قائمة بمسارات تسجيلات الواصفات (في حال امتداد الملف على أكثر من تسجيلة).
3. اسم الملف.
4. معلومات الأمان: مالك الملف والأشخاص المخولين بالوصول إليه.
5. معطيات الملف.
6. رقم مميز للغرض: رقم مميز للملف ضمن حجم التخزين.
7. فهرس الجذر: يستخدم لتحقيق المجلدات والفهارس.
8. فهرس الحجز: يستخدم لتحقيق المجلدات والفهارس.

نوع الواصفة	الوصف
معلومات قياسية	مثل عدد الوصلات والختم الزمني
قائمة الواصفات	قائمة بمسارات تسجيلات الواصفات (في حال امتداد الملف على أكثر من تسجيلة)
اسم الملف	اسم الملف
معلومات الأمان	مالك الملف والأشخاص المخولين بالوصول إليه
المعطيات	معطيات الملف
رقم مميز للغرض	رقم مميز للملف ضمن حجم التخزين
فهرس الجذر	يستخدم لتحقيق المجلدات والفهارس
فهرس الحجز	يستخدم لتحقيق المجلدات والفهارس

19. البحث عن ملف في نظام NTFS

- لنفرض أنه مطلوب فتح الملف التالي "C:\maria\web.htm" فكيف سيتم البحث عن هذا الملف:
- البدء بالمجلد الجذر، وذلك لإيجاد الحجم المناسب الذي سنبدأ منه البحث وفي مثالنا نبحث عن المدخل (C).
- بعد إيجاد الحجم المناسب نبحث في جدول الملف الأساسي MFT الخاص به، عن المجلد الذي يحمل الاسم "maria"، وهذه العملية تبدأ انطلاقاً من المجلد الجذر ضمن هذا الحجم.
- بعد إيجاد موقع المجلد المطلوب ضمن MFT، يتم البحث ضمن هذا المجلد عن الملف المطلوب "web.htm".
-



ملحق الرسم	
Directory	مجلد
Look up	البحث
Hard disk volume	حجم القرص الصلب
Device	التجهيزة
Root directory	المجلد الجذر
Path name	مسار الاسم

20. الملفات المترفعة في نظام NTFS

- إن الملفات الـ16 المترفعة هي عبارة عن ملفات نظام، كل منها مسؤول عن جزء معين من عمل النظام.
- توضع الملفات المترفعة في المجلد الجذر لقرص الـNTFS، وكل ملف يبدأ بإشارة $.
- قائمة الملفات المترفعة:

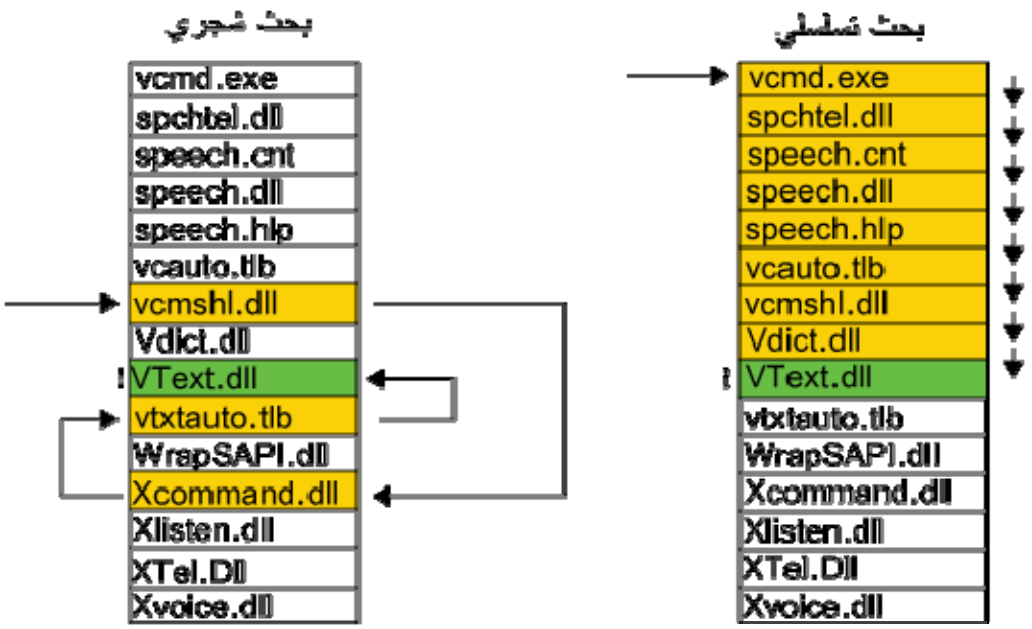
الملف	الوصف
$MFT	جدول الملفات الرئيسي
$MFTmirr	نسخة من الـ16 تسجيلة، مخزنة في منتصف القرص
$LogFile	ملف دعم
$Volume	معلومات عن نظام الملفات
$AtrrDef	قائمة بواصفات الملفات القياسية
$.	المجلد الجذر
$Bitmap	المساحة الفارغة
$Boot	قطاع الإقلاع
$Quota	معلومات عن صلاحيات المستخدم
$Upcase	جدول التوافق بين المحارف الكبيرة والصغيرة

21. المجلدات

المجلدات في نظام NTFS عبارة عن ملف خاص يحوي مؤشرات إلى ملفات ومجلدات أخرى.

يُقسَم ملف المجلد إلى كتل، كل منها يحوي اسم الملف، الواصفات الأساسية، بالإضافة إلى مؤشر إلى تسجيلة الملف في MFT، والتي تحوي على المعلومات الكاملة عن هذا الملف.

البنية الداخلية للمجلد عبارة عن شجرة ثنائية، وهذا ما يسرع من عملية البحث عن الملفات، مقارنة مع نظم الملفات الأخرى مثل FAT.



22. آلية حماية المعطيات من الضياع

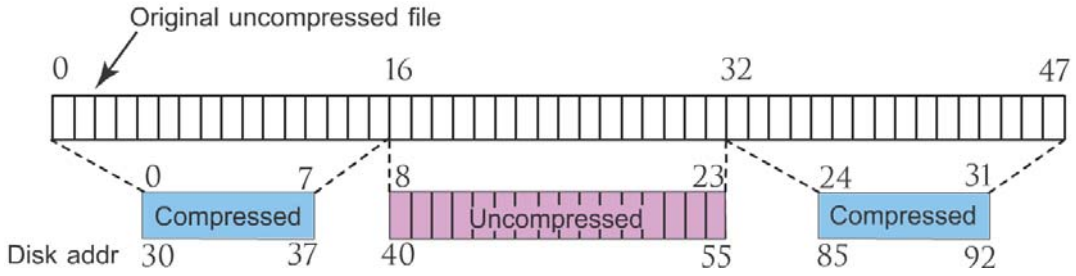
يحتوي نظام الملفات NTFS على نسختين متشابهتين من جدول الملفات الرئيسي (MFT)، فإذا تشوهت النسخة الأصلية منه نتيجة لتعطل قطاع ما، فإن النظام عند التشغيل التالي للجهاز، يستخدم النسخة الأخرى من MFT و ينشئ تلقائياً نسخة جديدة، مع الأخذ بعين الاعتبار وجود القطاع المعطل، وبهذا فإن النظام يضمن حفظ البيانات من الضياع.

23. ضغط الملفات في نظام NTFS

- يقدم نظام الملفات NTFS آلية ضغط شفافة بالنسبة للمستخدمين، حيث يمكن بناء الملف بنمط الضغط. وهذا يعني أن نظام الملفات يحاول ضغط كتل هذا الملف عند كتابتها على القرص، ويقوم بفك الضغط بشكل أوتوماتيكي عند قراءتها منه.
- تتم عملية الضغط كما يلي:

1. عندما يقوم نظام الملفات بكتابة ملف (بنمط الضغط)، يفحص الـ16 كتلة الأولى من الملف، ويطبّق خوارزمية الضغط عليها.
2. إذا كان بالإمكان تخزين نتيجة الضغط على عدد من الكتل أقل من 16، عندها يتم كتابة الكتل مضغوطة على القرص.
3. أما إذا كانت نتيجة الضغط ما تزال بحاجة إلى 16 كتلة، عندها يتم كتابة الكتل من دون ضغط.
4. يتم بعد ذلك تطبيق نفس العملية على الكتل الـ16 التالية وهكذا.

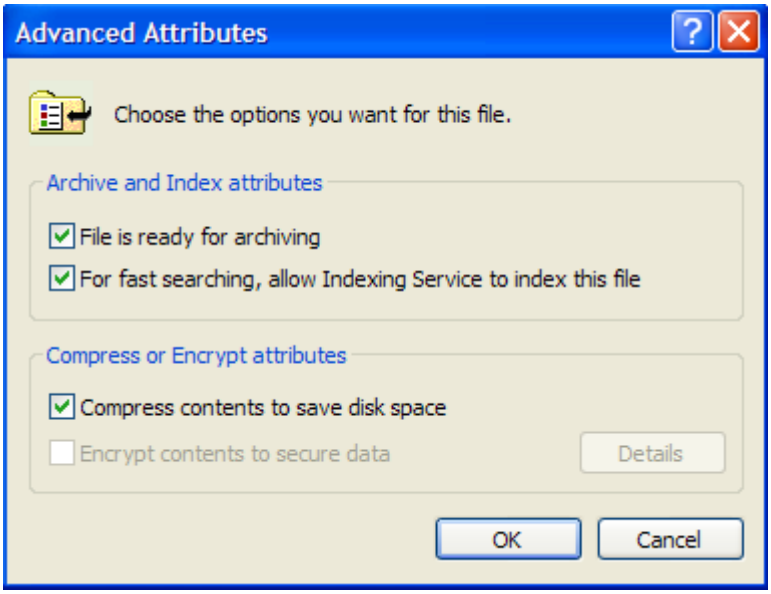
- نلاحظ من المثال التالي أن الكتل الـ16 الأولى قد تم ضغطها، أما الكتل الـ16 الثانية فبقيت من دون ضغط:



24. استخدام ضغط الملفات

يمكن استخدام نظام ضغط الملفات في نظام تشغيل Windows XP كما يلي:

ضغط يميني على الملف المراد ضغطه، اختيار خصائص، ثم خيارات متقدمة Advanced، ثم خيار ضغط المحتوى لحفظه على القرص:



25. نظام تشفير الملفات EFS

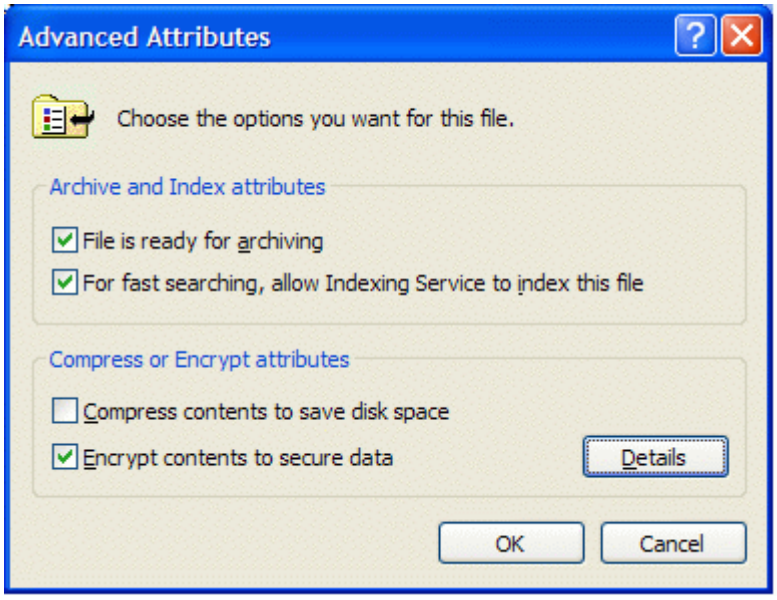
يُقدِم نظام تشفير الملفات (EFS)، نواة تقنية التشفير المستخدمة لتخزين الملفات المشفرة في نظام الملفات NTFS. يحمي هذا النظام الملفات من دخول غير المخولين إليها. حيث يستقبل المستخدمون غير المخولون بالدخول إلى ملف أو مجلد مشفر، رسالة "Access Denied" في حال محاولة الدخول.

لا يختلف التعامل مع الملفات المشفرة عن نظيرتها غير المشفرة، فعملية التشفير شفافة بالنسبة للمستخدم، حيث يتم فك التشفير بشكل أوتوماتيكي من قبل النظام، وذلك أثناء الدخول إلى ملف أو مجلد مشفر.

26. استخدام نظام تشفير الملفات

يمكن استخدام نظام تشفير الملفات في نظام تشغيل Windows XP كما يلي:

ضغط يميني على الملف المراد تشفيره، اختيار خصائص، ثم خيارات متقدمة Advanced، ثم خيار تشفير الملف:



قبل حفظ الإعدادات الجديدة، يقوم Windows بسؤال المستخدم إذا كان يرغب بتشفير الملف وحده أم الملف والمجلد الأب له:

27. ميزات نظام NTFS

ميزات نظام NTFS:

1. نظام ملفات قابل للاسترجاع: يحتفظ بنسخ من ملفات النظام المهمة، ويقوم باسترجاعها في حال حدوث مشكلة في النظام.
2. يدعم نظام NTFS إمكانية تشفير الملفات في نظام تشغيل windows 2000/XP.
3. يدعم إمكانية ضغط الملفات.
4. يؤمن ميزات أمن إضافية في حالة الإقلاع من القرص المرن (منع إمكانية الدخول إلى نظام الملفات أو طلب معلومات حساب المدير).
5. يدعم إمكانية حفظ نسختين من المعطيات على قرصين مختلفين.
6. يستخدم حجوم عناقيد أصغر من نظام الملفات FAT، مما يعطي فعالية أكبر في استخدام مساحة القرص الصلب.
7. يدعم حجوم تخزين كبيرة للأقراص الصلبة.

28. ميزات نظام FAT

ميزات نظام الملفات FAT:

1. أفضل بالنسبة للأقراص الصلبة ذات الحجوم الأقل من 500MB، فهو أخف حملاً على النظام من نظام الملفات NTFS.
2. متوافق مع نظام التشغيل DOS ومع نظام Windows 9x.
3. يمكن الإقلاع من قرص إقلاع windows 9x أو DOS لحل مشاكل النظام.

29. اختيار نظام الملفات

- نلاحظ دائما قبل تنصيب نظام تشغيل ما، أنه يجب تحديد نظام الملفات الذي سيتم استخدامه، ويتم الاختيار عادةً، تبعاً لحجم القرص، وطبيعة العمل على الحاسب، حيث إن نظام الملفات هو الأسلوب الذي يتم بموجبه تخزين المعطيات على القرص.
- تعتمد أنظمة التشغيل الحديثة وهي Windows 2000 و Windows XP نظام الملفات NTFS أو يمكن استخدام أحد أنظمة ملفات جدول تخصيص الملفات الأخرى FAT أو FAT32. ولكن يعتبر نظام الملفات NTFS هو النظام الموصى باستخدامه، حيث تتوفر فيه كافة قدرات نظام الملفات FAT الأساسية، بالإضافة إلى ميزات أخرى.
-

نظام التشغيل / نظام الملفات	DOS	Win95	Win98	WinNT	Win2000	WinXP
FAT16	X	X	X	X	X	X
FAT32			X (for OSR)		X	X
NTFS				X	X	X

30. التحويل بين أنظمة الملفات

التحويل من نظام الملفات FAT إلى نظام الملفات NTFS ممكن وبدون فقد للمعطيات، ويتم وفق الخطوات التالية:

قائمة ابدأ Start، ثم All programs، Accessories، Command prompt، ومن ثم كتابة التعليمة التالية:

convert drive_letter: /fs:ntfs.

مثلاً كتابة التعليمة التالية convert D:/fs:ntfs سوف تقوم بتهيئة السواقة D بتهيئة NTFS.

التحويل العكسي من NTFS إلى FAT غير ممكن، ويؤدي إلى ضياع في المعطيات، لذلك يجب استخدام برامج خاصة لذلك.

31. التمارين:

1. من مستويات لنظام الملفات:
   A. نظام الملفات الأساسي
   B. جزء تنظيم الملف
   C. نظام الملفات المنطقي
   D. جميع الإجابات صحيحة

2. أشهر أنظمة الملفات:
   A. UNIX V7
   B. ISO 9660
   C. NTFS
   D. جميع الإجابات صحيحة

3. يتألف نظام الملفات CP/M من مجلد واحد ثابت الحجم، يحتوي على مداخل ثابتة الحجم (32 بايت):
   A. صح
   B. خطأ

4. لا يخزن MS-DOS عناوين الكتل في مداخل المجلد، إنما يتم حفظها في جدول تخصيص الملفات (FAT) ضمن الذاكرة الرئيسية:
   A. صح
   B. خطأ

5. حجم كتل المعطيات من أجل كل جداول FAT هي من مضاعفات 512 بايت:
   A. صح
   B. خطأ

6. يعتبر جدول الملفات الرئيسي MFT الملف الأهم في نظام الملفات NTFS، وهو أول الملفات المترفعة في النظام:
   A. صح
   B. خطأ

7. أهم واصفات الملف في نظام الملفات NTFS:

A. معلومات الأمان

B. فهرس الجذر

C. فهرس الحجز

D. جميع الإجابات صحيحة

8. توضع الملفات المترفعة في المجلد الجذر لقرص الـNTFS، وكل ملف يبدأ بإشارة $:

A. صح

B. خطأ

9. ميزات نظام NTFS:

A. يدعم إمكانية ضغط الملفات

B. نظام ملفات قابل للاسترجاع

C. يستخدم حجوم عناقيد أصغر

D. جميع الإجابات صحيحة

رقم التمرين	الإجابة الصحيحة
.1	(D)
.2	(D)
.3	(A)
.4	(A)
.5	(A)
.6	(A)
.7	(D)
.8	(A)
.9	(D)

إدارة الذاكرة

الكلمات المفتاحية:

الخبيئة، سجل الحد، سجل القاعدة، التبديل، التبديل للخارج، التبديل للداخل، مدير الذاكرة، مُسند المهمات، الأطر، الذاكرة الافتراضية، عنوان منطقي، عنوان فيزيائي، جدول الصفحات، جدول المقاطع، التقطيع، نظام طلب الصفحات.

ملخص:

يركز هذا الفصل على التعرف على المفاهيم التالية: إدارة الذاكرة، التبديل، صفحات الذاكرة، التقطيع، خوارزميات التبديل.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- التعرف على بنية وهرمية الخزن.
- إدارة الذاكرة الرئيسية والخزن الثانوي.
- عناوين الذاكرة الفيزيائية والمنطقية وأقسام الذاكرة.
- التبديل.
- صفحات الذاكرة وجدول الصفحات وآلية حماية الصفحات.
- التقطيع.
- الذاكرة الافتراضية.
- نظام طلب الصفحات.
- إدارة مساحة التبديل وحجز أطر الذاكرة.
- خوارزميات تبديل الصفحات.

1. بنية الخزن

إن الذاكرة الرئيسية (RAM) والسجلات داخل المعالج نفسه، هما نوعا الخزن الوحيدين اللذين يمكن لوحدة المعالجة النفاذ إليهما مباشرة، لذلك فإن أي معطيات تحتاجها وحدة المعالجة يجب نقلها للذاكرة الرئيسية (RAM) قبل المعالجة.

يتضمن كل متحكم دخل/خرج سجلات من أجل الاحتفاظ بالأوامر والمعطيات، حيث تسمح تعليمات دخل/خرج خاصة بنقل المعطيات بين هذه السجلات وذاكرة النظام، ومن أجل توفير نفاذ أكثر ملائمة لمتحكمات الدخل/الخرج يتوفر في العديد من البنى دخل/خرج مقابل بالذاكرة، حيث يوضع مجال من عناوين الذاكرة جانباً ويُقابل بسجلات التجهيزة.

توفر الأقراص المغناطيسية معظم الخزن الثانوي لنظم الحواسيب، ويقسم سطح قرص التخزين منطقياً إلى مسارات دائرية والمسارات إلى قطاعات، وتسمى مجموعة المسارات الدائرية ذات البعد الواحد من مركز القرص بالاسطوانة.

2. هرمية التخزين

التخبئة (caching) هي مبدأ هام في النظم الحاسوبية، إذ تحفظ المعلومات عادةً في نظام خزن (الذاكرة الرئيسية) وعندما تستخدم هذه المعلومات، تنسخ إلى نظام خزن أسرع (الذاكرة الخبيئة).

عندما نحتاج إلى معلومة معينة ننظر أولاً إذا كانت موجودة في الخبيئة (نحصل عليها مباشرة)، أما في الحالة المعاكسة فإننا نحصل عليها من نظام الخزن الرئيسي ونضع في الوقت نفسه نسخة منها في الخبيئة، إذ أن هنالك احتمال كبير أن نحتاج إلى هذه المعلومات مرة ثانية قريباً.

يمكن اعتبار الذاكرة الرئيسية خبيئة سريعة للخزن الثانوي (أقراص صلبة أو مدمجة مثلاً)، ذلك أن المعطيات في الخزن الثانوي يجب أن تنسخ في الذاكرة الرئيسية لكي تستخدم.

يمكن أن تكون حركة المعلومات بين مستويات هرمية الخزن إما صريحة، أو ضمنية، وفقاً لتصميم عتاديات الجهاز وبرمجيات نظام التشغيل، فعلى سبيل المثال تكون عملية نقل المعطيات من الخبيئة إلى سجلات وحدة المعالجة عادةً وظيفة عتادية، بينما يكون نقل المعطيات من القرص إلى الذاكرة عملية يتحكم بها نظام التشغيل.

3. اتساق الخبيئة

في بنية خزن هرمية، يمكن أن تظهر نفس المعطيات في مستويات مختلفة من نظام الخزن. مثلاً لنفترض أن العدد الصحيح A موجود في الملف B ويجب زيادة قيمته بمقدار 1، وأن الملف B موجود في القرص المغناطيسي. تنفذ الزيادة بإجراء دخل/خرج أولاً، لنسخ كتلة القرص التي تحوي A إلى الذاكرة الرئيسية، يلي هذه العملية نسخ A إلى الخبيئة وإلى سجل داخلي. وبعد إجراء الزيادة في السجل الداخلي، تأخذ A قيماً مختلفة في نظم الخزن المختلفة. ولا تصبح قيمة A وحيدة إلا بعد نقل القيمة الجديدة لـA من السجل الداخلي إلى القرص المغناطيسي.

في بيئة حاسوبية تنفذ فيها إجراء واحد فقط في وقت واحد، لا يسبب هذا الترتيب أية صعوبات، ولكن في بيئة متعددة المهمات حيث يتم تبديل وحدة المعالجة بين الإجراءات يجب التأكد من حصول جميع الإجراءات على أحدث قيمة لـA.

تصبح الحالة أكثر تعقيداً في البيئة المتعددة المعالجات، حيث تحوي كل وحدة معالجة إضافة إلى السجلات الداخلية، خبيئة محلية أيضاً.

يدعى هذا الوضع اتساق الخبيئة وهو عادة مسألة عتادية تعالج في مستوى دون مستوى نظام التشغيل.

4. إدارة الذاكرة الرئيسية

الذاكرة الرئيسية عبارة عن مخزن معطيات مشترك بين وحدة المعالجة وتجهيزات الدخل/الخرج، وهي تجهيزة الخزن الكبيرة الوحيدة التي يمكن لوحدة المعالجة أن تعنونها وتنفذ إليها مباشرة.

نظام التشغيل مسؤول عن الفعاليات التالية فيما يخص إدارة الذاكرة:

- تعقب أجزاء الذاكرة المستخدمة حالياً، وهوية مستخدمها
- اختيار الإجراء الذي يجب تحميله في الذاكرة، حين يصبح فضاء الذاكرة متاحاً
- تقسيم فضاء الذاكرة إلى حصص، وإعادة تقسيمه بحسب الحاجة

كما أن نظام التشغيل مسؤول عن الحماية، من خلال التحكم في نفاذ البرامج أو الإجراءات أو المستخدمين إلى الموارد التي يحددها.

5. إدارة الخزن الثانوي

يجب أن تكون البرامج والمعطيات التي يتعامل معها النظام، ضمن الذاكرة الرئيسية من أجل تنفيذها. وبسبب كون الذاكرة الرئيسية أصغر من اللازم لتكفي كل المعطيات والبرامج، وبسبب فقدان المعطيات ضمنها بمجرد توقف التغذية، يجب توفير مخزن معطيات أكبر (ولا يفقد المعطيات)، من أجل التخزين والتعامل مع المعطيات والبرامج.

تَستخدم معظم نظم الحواسب الأقراص المغناطيسية كخزن ثانوي، ويهتم نظام التشغيل بالعمليات الخاصة على هذا الخزن من إدارة وتحصيص لمساحة التخزين، بالإضافة على جدولة القرص.

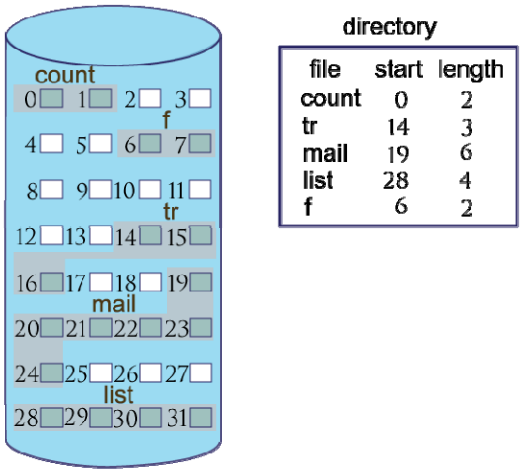
البنية الأساسية للتخزين على القرص هي الكتلة، وبالتالي جميع الملفات التي تخزن على القرص تكون عبارة عن مجموعة من الكتل.

هنالك عدة طرق لحجز كتل الملفات:

1. حجز مستمر:

يحتاج كل ملف إلى حجز مجموعة متتالية من كتل المعطيات من أجل تخزينه. وبالتالي فإن عملية القراءة من القرص تتم بشكل تسلسلي أيضاً.

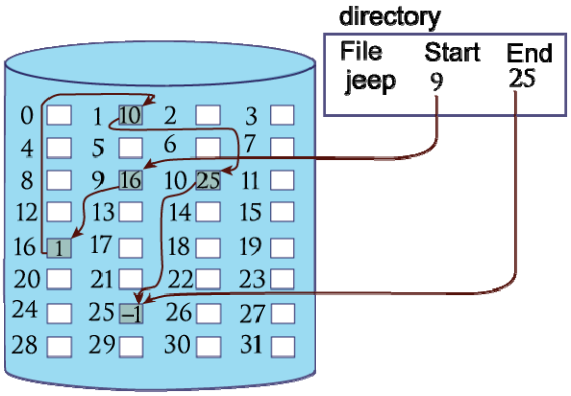
المشكلة الأساسية في هذه الطريقة هي إيجاد عدد كافي من الكتل المتتالية لتخزين الملف.



Contiguous allocation
حجز مستمر

2. حجز مترابط:

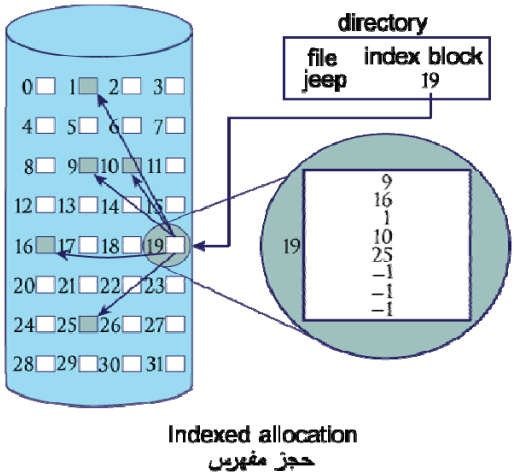
كل ملف عبارة عن مجموعة من الكتل المترابطة، حيث يحتوي المجلد على مؤشر إلى أول وآخر كتلة في الملف، كما أن كل كتلة تشير إلى الكتلة التي تليها.



Linked allocation
حجز مترابط

3. حجز مفهرس:

يوجد لكل ملف كتلة فهرس تحوي على عناوين الكتل الخاصة بهذا الملف، وبالتالي فإن الوصول إلى أي كتلة، يتم بشكل مباشر من خلال كتلة الفهرس، وهذا يسرع عملية الوصول.



Indexed allocation
حجز مفهرس

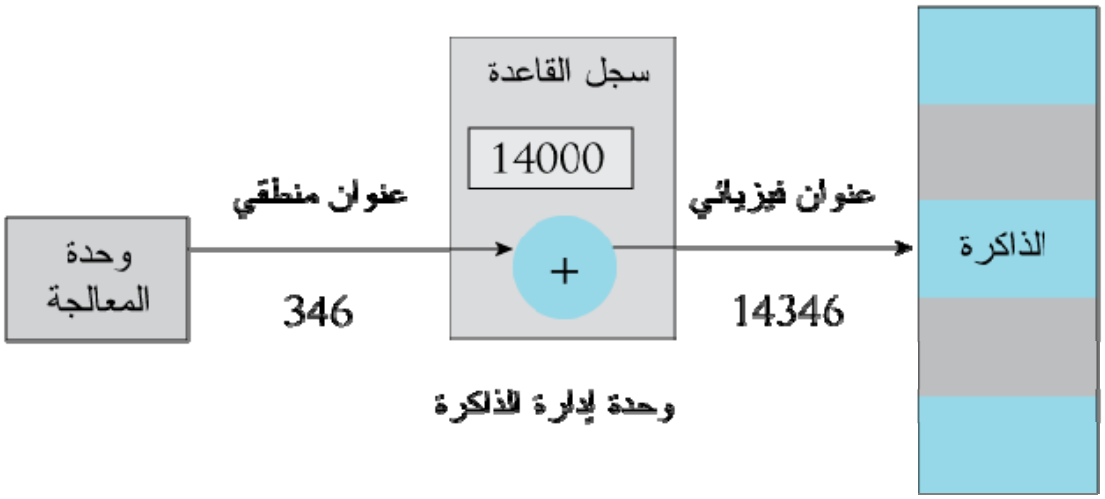
6. عناوين الذاكرة الفيزيائية والمنطقية

يُشار عادةً إلى العنوان الذي يتم توليده من قبل المعالج، على أنه عنوان منطقي. بينما يُشار إلى العنوان الذي تتعامل معه وحدة الذاكرة، على أنه عنوان فيزيائي.

تقوم وحدة إدارة الذاكرة بإجراء التقابل بين العناوين المنطقية والعناوين الفيزيائية. وهنالك عدة طرق لتحقيق هذا التقابل، وأحد هذه الطرق بالاعتماد على سجل القاعدة، حيث يتم إضافة أي عنوان منطقي يُولّد من قبل الإجراء، إلى القيمة الموجودة في السجل، للحصول على القيمة الفيزيائية المقابلة لذلك العنوان.

نلاحظ أن الإجراء لا يتعامل أبداً مع العناوين الفيزيائية المباشرة.

مثال: إذا كانت القيمة الموجودة في سجل القاعدة تساوي 14,000، والعنوان المنطقي المولد هو 0، فإن العنوان الفيزيائي يكون 14,000.



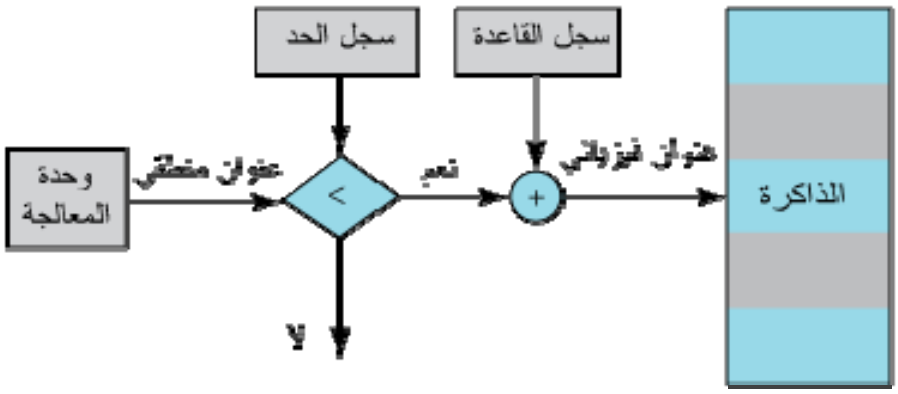
7. أقسام الذاكرة

يجب أن تتسع الذاكرة إلى نظام التشغيل وإجراءات المستخدمين، حيث تُقسم عادةً إلى قسمين: الأول لنظام التشغيل، والثاني لإجراءات المستخدمين.

ويكون عادةً القسم الخاص بنظام التشغيل ضمن العناوين الدنيا من الذاكرة، والقسم الخاص بإجراءات المستخدمين ضمن العناوين العليا.

بما أن نظام التشغيل والإجراءات تُحمّل في الذاكرة، لذلك نحتاج إلى حماية نظام التشغيل من إجراءات المستخدم، كما يجب حماية إجراءات المستخدمين فيما بينها. ولذلك الغرض نستخدم سجل القاعدة أو سجل الموضع، وسجل الحد. حيث يحوي سجل القاعدة على أصغر عنوان فيزيائي ضمن الجزء، بينما يحوي سجل الحد على مجال العناوين المنطقية ضمن الجزء. ويجب أن يكون كل عنوان منطقي أصغر من سجل الحد.

يقوم مُسند المهمات بتحميل القيم الصحيحة للسجلين السابقين، أثناء تحميل الإجراء إلى الذاكرة من أجل التنفيذ.



8. التبديل (Swapping)

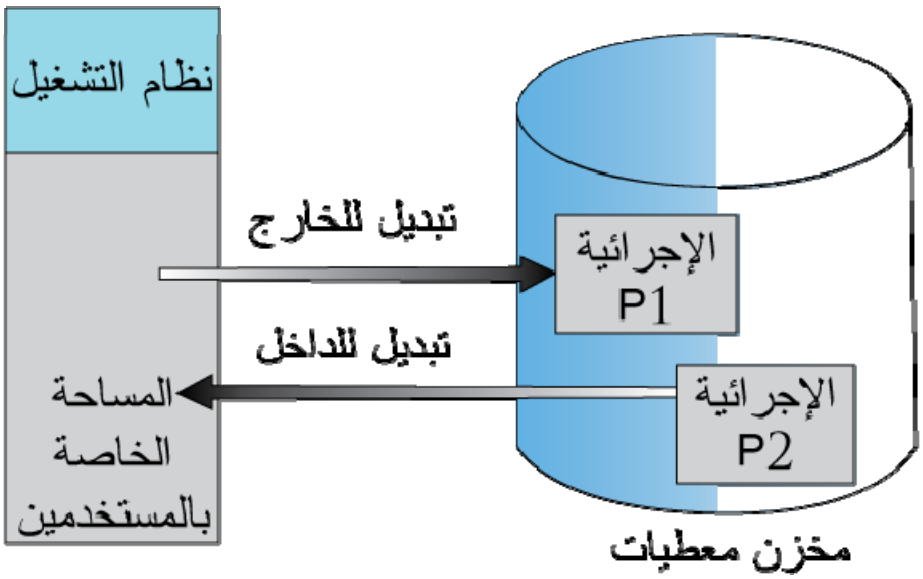
كما رأينا في الدروس السابقة، يحتاج الإجراء أن يكون في الذاكرة الرئيسية من أجل التنفيذ، لكن يمكن للإجراء أن ينتقل من الذاكرة الرئيسية (مكان التنفيذ)، إلى مكان تخزين للمعطيات (القرص الصلب)، ومن ثم يعود ثانية إلى الذاكرة، وهذه العملية تدعى التبديل (swapping).

في المعالج متعدد المهام يتم التبديل بين الإجراءات المنفذة بشكل مستمر، حيث يتم استخدام عمليتي التبديل للخارج، والتبديل للداخل بين الإجراءات المنفذة.

يقوم مدير الذاكرة بإدارة عملية التبديل بين الإجراءات بسرعة، بحيث يكون دائماً أحد الإجراءات قيد التنفيذ.

هنالك عدة سياسات لعملية التبديل، ففي خوارزمية الجدولة المعتمدة على الأولوية، عندما يصل إجراء ذو أولوية عليا ويحتاج وحدة المعالجة، يقوم مدير الذاكرة بالتبديل بين هذا الإجراء والإجراء ذو الأولوية الدنيا الذي يجري تنفيذه، وبعد انتهاء تنفيذ الإجراء ذو الأولوية العليا، يعود الإجراء ذو الأولوية الدنيا للتنفيذ.

تحتاج عملية التبديل إلى مخزن معطيات سريع وكبير كفاية.



عندما يقرر مجدول وحدة المعالجة تنفيذ إجراء ما، يقوم باستدعاء مسند المهمات، الذي يفحص بدوره إذا كان الإجراء الذي سيجري تنفيذه موجود في الذاكرة أم لا، وهنالك ثلاث حالات:

1. إذا كان الإجراء موجود في الذاكرة، فإنه يبدأ عملية التنفيذ مباشرةً.
2. إذا لم يكن الإجراء موجود في الذاكرة وهنالك مساحة كافية في الذاكرة، يقوم مسند المهمات بتحميل الإجراء ومن ثم يبدأ التنفيذ.
3. أما إذا لم يكن هنالك مساحة كافية، فإن مسند المهمات يقوم بعملية التبديل بين الإجراء الموجود في الذاكرة، والإجراء المطلوب تنفيذه.

9. صفحات الذاكرة

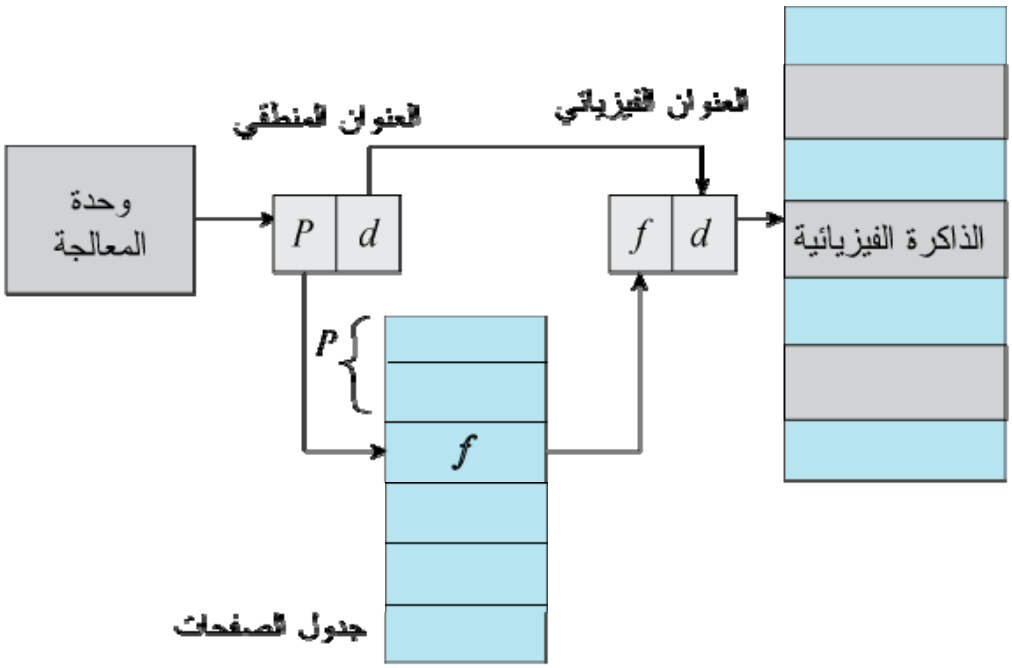
تقسم الذاكرة الفيزيائية إلى كتل ثابتة الحجم تدعى أطر، كما أن الذاكرة المنطقية مقسمة إلى كتل من نفس الحجم تدعى صفحات.

عندما يُطلب تنفيذ إجراء معين، يتم تحميل صفحاته من مخزن المعطيات إلى أطر الذاكرة الرئيسية المتوفرة، حيث أن أطر الذاكرة الرئيسية مساوية في الحجم لأطر مخزن المعطيات.

من أجل التعامل مع صفحات الذاكرة، يتم تقسيم عناوين الذاكرة التي يطلبها المعالج، إلى رقم الصفحة (p) وانزياح الصفحة (d).

يُستخدَم رقم الصفحة (p) كمدخل لجدول الصفحات، الذي يحوي على العنوان الأساسي للصفحة ضمن الذاكرة الفيزيائية (f)، بينما يستخدم الانزياح (d) مع عنوان الذاكرة (f) من أجل تحديد صفحة الذاكرة الفيزيائية. إذاً يُعرَّف عنوان الذاكرة الفيزيائية من خلال تجميع العنوان الأساسي لصفحة الذاكرة مع انزياح هذا العنوان.

يُعرَّف حجم الصفحة من قبل العتاد، وهو من مضاعفات 2، بين 512 بايت و 16 ميغابايت.



10. جدول الصفحات

يستخدم كل نظام تشغيل طريقة خاصة لتخزين جداول الصفحات، ولكن معظم النظم تحجز جدول صفحات من أجل كل إجراء، حيث يُخزَّن المؤشر إلى هذا الجدول في أحد سجلات كتلة التحكم بالإجراء، ولذلك عندما يُطلَب من مُسند المهمات تنفيذ إجراء جديد، يقوم بتحميل جدول الصفحات الخاص بالإجراء الجديد، وتعديل قيمة السجلات.

هنالك طرق مختلفة لتحقيق جدول الصفحات بشكل عتادي، وأبسط هذه الطرق هي بتحقيق جدول الصفحات من خلال مجموعة من السجلات، حيث يتم تحميل هذه السجلات مع السجلات الأخرى للإجراء. هذه الطريقة جيدة إذا كان حجم الجدول صغير (حوالي 256 مدخل). ولكن حجم الجدول في معظم الحواسب كبير جداً (حوالي 1 مليون مدخل)، لذلك فإن طريقة السجلات غير فعّالة، لذا يتلخص الحل في إبقاء جدول الصفحات في الذاكرة الرئيسية (بدلاً من السجلات)، وتخزين المؤشر إلى هذا الجدول في السجل "سجل جدول الصفحات الأساسي"، وبالتالي يجري تبديل سجل جدول الصفحات الأساسي أثناء عملية التبديل بين الإجراءات، بدلاً من تبديل جميع السجلات.

المشكلة في الطريقة الثانية هو زمن الوصول إلى الذاكرة، حيث نحتاج أولاً إلى زمن للوصول إلى جدول الصفحات في الذاكرة للحصول على العنوان الأساسي للصفحة، ومن ثم استخدام هذا العنوان مع انزياح الصفحة لتشكيل العنوان الفعلي لإطار المعطيات، ومن ثم جلبه من الذاكرة.

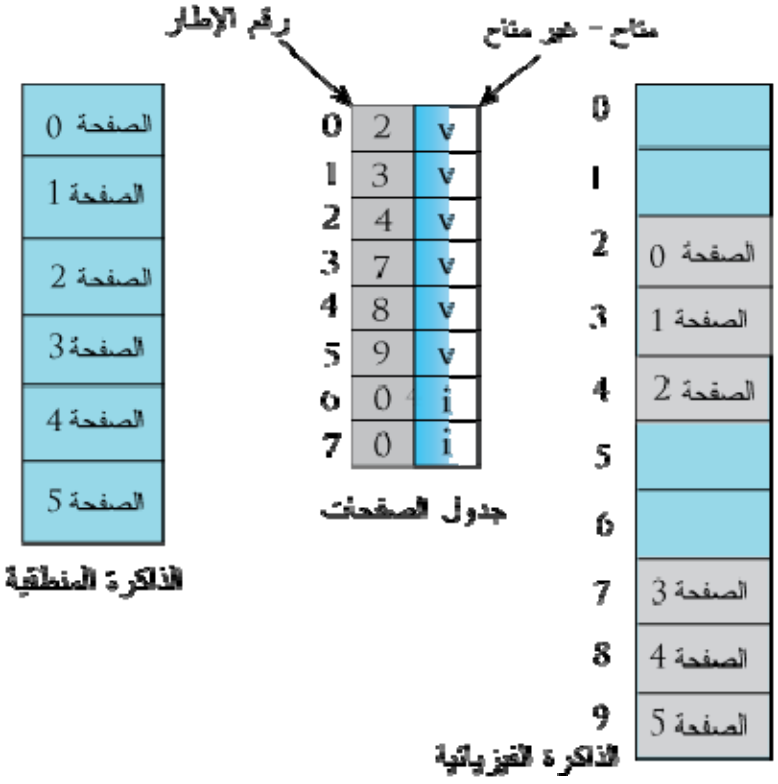
11. حماية الصفحات

يُستخدَم بت إضافي مع كل إطار ذاكرة من أجل الحماية (بت الحماية protection bit)، حيث يُحفظ هذا البت عادةً في جدول الصفحات، ويُعرِّف الصفحة على أنها للقراءة والكتابة، أو للقراءة فقط.

وبما أن عملية الوصول إلى الإطار المناسب من الذاكرة تتم عن طريق جدول الصفحات، من خلال حساب العنوان الفيزيائي للذاكرة، انطلاقاً من العنوان الأساسي ضمن جدول الصفحات. يتم فحص بت الحماية أثناء حساب العنوان الفيزيائي، للتحقق من عدم الكتابة على صفحة للقراءة فقط.

تُسبّب عملية الكتابة على صفحة للقراءة فقط، مقاطعة عتادية تتم معالجتها من قبل نظام التشغيل.

هنالك بت إضافي آخر تتم إضافته عادةً إلى كل صفحة للدلالة إذا كانت الصفحة متاحة للاستخدام أم لا (valid أو in–valid). فإذا كانت الصفحة متاحة، يعني أنها موجودة ضمن مساحة الذاكرة المنطقية التابعة للإجراء، وبالتالي يمكن استخدام هذه الصفحة من قبل الإجراء. وبالعكس إذا كانت الصفحة غير متاحة، فهذا يعني أنها خارج مساحة الذاكرة المنطقية التابعة للإجراء، وبالتالي لا يمكن استخدمها من قبل الإجراء.



12. التقطيع

التقطيع هو آلية تنظيم وإدارة للذاكرة، فمساحة العناوين المنطقية التي تتعامل معها الإجراءات عبارة عن مقاطع، وكل مقطع له اسم وطول.

تحدد العناوين اسم المقطع والانزياح ضمنه، كما يُستخدم عادةً رقم للمقطع بدلاً من الاسم، وذلك من أجل سهولة التعامل.

نحتاج إلى آلية للتقابل بين العناوين المنطقية والعناوين الفيزيائية، ويُستخدم لهذا الغرض جدول للمقاطع كل مدخل منه يحوي على قاعدة المقطع وعلى حد المقطع.

تحوي قاعدة المقطع على العنوان الفيزيائي لبداية المقطع في الذاكرة، بينما يحدد حد المقطع طول هذا المقطع.

يتألف العنوان المنطقي من قسمين:

1. رقم المقطع s: حيث يُستخدم رقم المقطع كدليل للوصول إلى مدخل هذا المقطع ضمن جدول المقاطع.
2. الانزياح ضمن المقطع d: بينما يكون الانزياح بين 0 وحد المقطع.

13. تمرين

ليكن لدينا جدول المقاطع التالي:

المقطع	القاعدة	الطول
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

ما هي العناوين الفيزيائية للعناوين المنطقية التالية:

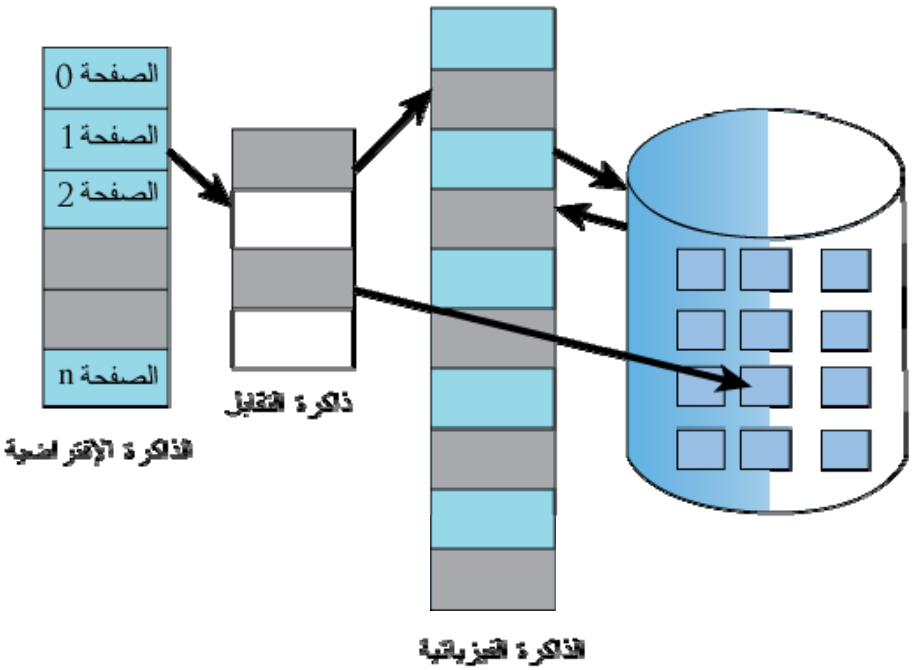
0,430 – 1,10 – 2,500 – 3,400 – 4,112.

14. الذاكرة الافتراضية

إن الهدف الأساسي من تقنيات إدارة الذاكرة (صفحات الذاكرة، والتبديل، والاستراتيجيات الأخرى)، هو إبقاء مجموعة من الإجراءات في الذاكرة بنفس الوقت وتبديل التنفيذ فيما بينها.

أما بالنسبة للذاكرة الافتراضية فهي تقنية تسمح بتنفيذ إجراء لا يمكن تحميله بشكل كامل ضمن الذاكرة الرئيسية، والفكرة هي تحويل الذاكرة الرئيسية بشكل منطقي إلى ذاكرة كبيرة في الحجم، من خلال استخدام مخزن معطيات (القرص الصلب). حيث تظهر الذاكرة الرئيسية والقرص الصلب على أنهما ذاكرة منطقية واحدة بالنسبة للإجراء. وبذلك فعلى الرغم من وجود ذاكرة فيزيائية صغيرة الحجم، إلا أن الإجراء يتعامل مع ذاكرة منطقية كبيرة جداً.

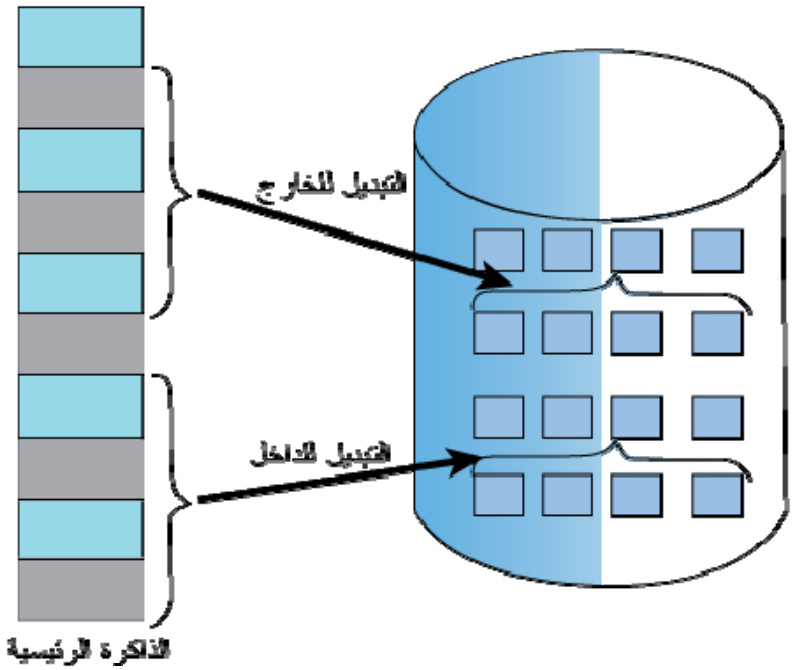
يمكن تحقيق الذاكرة الافتراضية بعدة طرق، واحدى هذه الطرق هي استخدام نظام طلب الصفحات.



15. نظام طلب الصفحات

يشبه نظام طلب الصفحات، نظام التبديل بين الصفحات، حيث توضع الإجراءات في مخزن المعطيات (القرص الصلب)، ويتم تحميل صفحاتها في الذاكرة عند طلب تنفيذ هذا الإجراء.

ولكن في نظام طلب الصفحات، لا تتم عملية تحميل صفحات الإجراء دفعة واحدة إلى الذاكرة، إنما يُستخدم مُبدل يدعى المُبدل الكسول، حيث لا يقوم هذا المُبدل بتحميل الصفحة إلى الذاكرة، إلا عند الحاجة إليها. فعند الحاجة إلى صفحة معينة يقوم المُبدل بتحميلها مباشرة إلى الذاكرة (عملية "تبديل للداخل") في حال وجود مساحة كافية في الذاكرة، أو استبدال أحد الصفحات في الذاكرة بالصفحة المطلوبة وذلك في حال عدم توفر مساحة كافية (عملية "تبديل للخارج" ثم عملية "تبديل للداخل").



نحتاج في هذا النظام إلى تقنية عتادية للتمييز بين الصفحات الموجودة في الذاكرة والصفحات الموجودة على القرص، ولذلك الغرض يمكن استخدام بت الدلالة على متاحية الاستخدام (الذي تم شرحه سابقاً valid–invalid bit)، فإذا كانت الصفحة متاحة هذا يعني أنها في الذاكرة ويمكن استخدامها، أما إذا كانت غير متاحة فهي إما موجودة على القرص (حيث يحتوي جدول الصفحات في مدخل هذه الصفحة على عنوانها على القرص)، أو ضمن الذاكرة ولكن خارج مجال عناوين هذا الإجراء.

16. إدارة مساحة التبديل

تَستخدم الذاكرة الافتراضية مساحة القرص كتوسعة للذاكرة الرئيسية، وبما أن الوصول إلى القرص أبطأ من الوصول إلى الذاكرة، لذلك فإن لاستخدام مساحة التبديل تأثير كبير على أداء النظام.

يختلف حجم مساحة التبديل التي يحتاجها النظام، تبعاً إلى حجم الذاكرة الفيزيائية وحجم الذاكرة الافتراضية، بالإضافة إلى طريقة استخدام الذاكرة الافتراضية.

يمكن أن تكون مساحة التبديل ضمن نظام الملفات نفسه، أو على جزء آخر من القرص.

فعندما تكون مساحة التبديل عبارة عن ملف كبير ضمن نظام الملفات، يهتم نظام الملفات ووظائفه بالتعامل مع هذه المساحة. بينما إذا تم استخدام جزء آخر من القرص من أجل مساحة التبديل، فإن مدير مستقل لعملية التخزين يهتم بآلية حجز وإلغاء حجز كتل المعطيات ضمن مساحة التبديل.

17. حجز أطر الذاكرة

كيف تتم عملية تقسيم مساحة الذاكرة المحدودة بين مجموعة الإجراءات التي يجري استخدامها. وما هو عدد الأطر التي يجب أن يحصل عليها كل إجراء.

يمكن إتباع طريقة بسيطة للحجز، حيث يتم حجز كل الأطر التي يحتاجها نظام التشغيل، ومن ثم استخدام الأطر المتبقية للإجراءات الأخرى في النظام.

هنالك عدة خوارزميات من أجل تقسيم الأطر غير المحجوزة بين إجراءات النظام:

1. طريقة الحجز المتساوي (equal allocation): وهي أبسط الطرق حيث تعتمد على تقسيم m إطار على n إجراء، بحيث يحصل كل إجراء على m/n إطار.
2. طريقة الحجز المتناسب (proportional allocation): حيث تعتمد على حجز عدد من أطر الذاكرة يتناسب مع حجم كل إجراء.

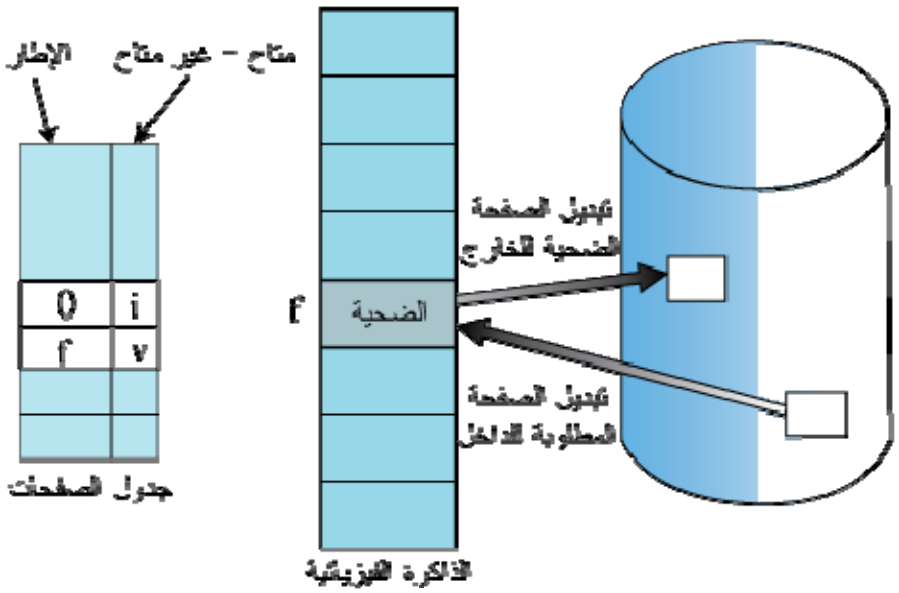
18. استبدال الصفحات

تأخذ عملية استبدال الصفحات المنحى التالي، في حال عدم وجود أي إطار غير محجوز ضمن الذاكرة، البحث عن إطار محجوز وتحريره ومن ثم استخدامه لتحميل الصفحة المطلوبة.

تتم عملية تحرير الإطار من خلال كتابة محتواه ضمن مساحة التبديل، وتغيير جدول الصفحات للإشارة أن هذه الصفحة لم تعد موجودة بعد الآن ضمن الذاكرة.

تتضمن عملية استبدال الصفحات الخطوات التالية:

1. إيجاد مسار الصفحة المطلوب على القرص.
2. إيجاد إطار غير محجوز: في حال وجود إطار غير محجوز يتم استخدامه مباشرةً، يتم استخدام أحد خوارزميات التبديل من أجل اختيار الإطار الضحية لعملية التبديل، ثم كتابة الصفحة الضحية على القرص وتعديل الصفحة وجدول الصفحات.
3. قراءة الصفحة المطلوبة وتحميلها ضمن الإطار غير المحجوز، ثم تعديل الصفحة وجدول الصفحات.
4. متابعة عمل الإجراء.



19. خوارزميات التبديل

هنالك عدة خوارزميات مستخدمة لعملية التبديل، وكل منها يتبع استراتيجية معينة من أجل اختيار الصفحة التي سيجري تبديلها.

1. خوارزمية القادم أولاً يُستبدَل أولاً (FIFO):

هي أبسط خوارزميات التبديل، حيث يُسند إلى كل صفحة تاريخ جلبها إلى الذاكرة، وبالتالي عند الحاجة إلى تبديل أحد الصفحات، تقوم الخوارزمية باختيار الصفحة الأقدم (من خلال زمن جلبها إلى الذاكرة)، ومن ثم استبدالها.

بشكل عام إن أداء هذه الخوارزمية ليس جيد، ولكنها سهلة التحقيق البرمجي.

2. خوارزمية التبديل الأمثلية (Optimal):

تعتبر هذه الخوارزمية الأفضل بين خوارزميات التبديل ولكن تحقيقها البرمجي صعب، وهي تعتمد على فكرة تبديل الصفحة التي لن يتم استخدامها لاحقاً إلا بعد أطول فترة ممكنة، ولذلك فإن هذه الخوارزمية تهتم بالزمن التي ستستخدم فيه الصفحة من جديد، والصفحة ذات الزمن الأطول هي التي تُستبدَل.

3. خوارزمية التبديل للأقل استخداماً مؤخراً (LRU):

تعتمد هذه الخوارزمية على فكرة تبديل الصفحة التي لم يتم استخدامها منذ أطول فترة، ولذلك فإن هذه الخوارزمية تهتم بزمن آخر مرة تم استخدام الصفحة فيها، والصفحة ذات الزمن الأطول هي التي تُستبدَل.

4. خوارزمية عدد مرات الطلب (Counting):

يُخزَن في هذه الخوارزمية مع كل صفحة، عدد المرات التي تم استخدامها، وهنالك استراتيجيتان:

- تبديل الصفحة التي استُخدمت أكبر عدد من المرات، لأنها ربما لم يعد هنالك حاجة إليها.
- تبديل الصفحة التي استُخدمت أقل عدد من المرات، لأنه من المحتمل أكثر استخدام الصفحات الأخرى.

20. تمرين

احسب عدد مرات التبديل التي سيقوم بها النظام من أجل سلسلة عناوين الكلمات التالية:
1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6.

وذلك باستخدام خوارزميات:

- خوارزمية FIFO
- الخوارزمية الأمثلية
- خوارزمية LRU

ومن أجل ذاكرة تحوي 1 أو 2 أو 3 أو 4 أو 5 أو 6 أو 7 أطر.

21. التمارين:

1. إن الذاكرة الرئيسية (RAM) والسجلات داخل المعالج نفسه، هما نوعا الخزن الوحيدين اللذين يمكن لوحدة المعالجة النفاذ إليهما مباشرة:
   A. صح
   B. خطأ

2. يمكن أن تكون حركة المعلومات بين مستويات هرمية الخزن إما صريحة، أو ضمنية، وفقاً لتصميم عتاديات الجهاز وبرمجيات نظام التشغيل:
   A. صح
   B. خطأ

3. نظام التشغيل مسؤول عن الفعاليات التالية فيما يخص إدارة الذاكرة:
   A. تقسيم فضاء الذاكرة إلى حصص، وإعادة تقسيمه بحسب الحاجة
   B. تعقب أجزاء الذاكرة المستخدمة حالياً، وهوية مستخدمها
   C. اختيار الإجراء الذي يجب تحميله في الذاكرة
   D. جميع الإجابات صحيحة

4. هنالك عدة طرق لحجز كتل الملفات:
   A. حجز مفهرس
   B. حجز مباشر
   C. حجز مترابط
   D. الإجابتين 1 و3
   E. الإجابتين 1 و2

5. يُشار عادةً إلى العنوان الذي يتم توليده من قبل المعالج، على أنه عنوان فيزيائي. بينما يُشار إلى العنوان الذي تتعامل معه وحدة الذاكرة، على أنه عنوان منطقي:
   A. صح
   B. خطأ

6. تقسم الذاكرة الفيزيائية إلى كتل ثابتة الحجم تدعى أطر، كما أن الذاكرة المنطقية مقسمة إلى كتل من نفس الحجم تدعى صفحات:

A. صح

B. خطأ

7. الذاكرة الافتراضية هي تقنية تسمح بتنفيذ إجراء لا يمكن تحميله بشكل كامل ضمن الذاكرة الرئيسية:

A. صح

B. خطأ

8. يمكن تحقيق الذاكرة الافتراضية بعدة طرق، واحدى هذه الطرق هي استخدام نظام طلب الصفحات:

A. صح

B. خطأ

9. في نظام طلب الصفحات، تتم عملية تحميل صفحات الإجراء دفعة واحدة إلى الذاكرة:

A. صح

B. خطأ

10. يختلف حجم مساحة التبديل التي يحتاجها النظام، تبعاً إلى:

A. طريقة استخدام الذاكرة الافتراضية

B. حجم الذاكرة الفيزيائية

C. حجم الذاكرة الافتراضية

D. جميع الإجابات صحيحة

11. هنالك عدة خوارزميات من أجل تقسيم الأطر غير المحجوزة بين إجراءات النظام:

A. طريقة الحجز المتناسب

B. طريقة الحجز المتساوي

C. طريقة الحجز المباشر

D. الإجابتين 1 و2

E. جميع الإجابات صحيحة

12. تتضمن عملية استبدال الصفحات الخطوات التالية:

A. إيجاد إطار غير محجوز

B. متابعة عمل الإجراء

C. إيجاد مسار الصفحة المطلوب على القرص

D. جميع الإجابات صحيحة

13. تعتمد على فكرة تبديل الصفحة التي لن يتم استخدامها لاحقاً إلا بعد أطول فترة ممكنة:

A. خوارزمية التبديل للأقل استخداماً مؤخراً

B. خوارزمية التبديل الأمثلية

C. خوارزمية عدد مرات الطلب

D. خوارزمية القادم أولاً يُستبدَل أولاً

رقم التمرين	الإجابة الصحيحة
.1	(A)
.2	(A)
.3	(D)
.4	(D)
.5	(B)
.6	(A)
.7	(A)
.8	(A)
.9	(B)
.10	(D)
.11	(D)
.12	(D)
.13	(B)

تزامن الإجراءات

الكلمات المفتاحية:

الإجراء المتعاون، المنتج والمستهلك، مقطع حرج، استبعاد متبادل، مقطع الدخول، مقطع الخروج، خوارزمية المخبز، سيمافور، انتظر وأرسل، انتظار مشغول، الإقفال المتبادل، القرّاء والكتّاب، الفلاسفة الطاعمين، المراقب.

ملخص:

يركز هذا الفصل على مفهوم التعاون بين الإجراءات، والمشكلات الناتجة عن هذا التعاون، بالإضافة إلى حلول هذه المشكلات.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- التعرف على الإجراءات المتعاونة.
- مشكلة المنتج والمستهلك.
- مشكلة المقطع الحرج وحلولها (الخوارزمية1، الخوارزمية2، الخوارزمية3، خوارزمية المخبز).
- التعليمات الخاصة للتزامن على مستوى العتاديات.
- السيمافور واستخدامها لحل مشكلة المقطع الحرج.
- مشكلات التزامن (مشكلة المنتج والمستهلك، مشكلة القراء والكتاب، مشكلة الفلاسفة الطاعمين).

1. الإجراءات المتعاونة

إن الإجراء المتعاون هو إجراء يؤثر ويتأثر بالإجراءات الأخرى الموجودة في النظام.
يمكن أن تتشارك الإجراءات المتعاونة مباشرةً في فضاء العنونة المنطقية (الرماز والمعطيات)، ويتم هذا التشارك من خلال استخدام الإجراءات الخفيفة أو مسالك التنفيذ (threads)، كما يمكن أن يُسمح للإجراءات المتعاونة بالتشارك في المعطيات عن طريق الملفات.
يمكن أن يؤدي نفاذ مجموعة من الإجراءات على نحو متوازي إلى المعطيات المشتركة إلى عدم اتساق المعطيات، لذلك نحتاج إلى آليات لضمان التنفيذ بترتيب محدد لإجراءات متعاونة تتشارك في فضاء العناوين المنطقية.

2. مشكلة المنتج والمستهلك (producer consumer)

- لدينا إجراء للمنتج يقوم بتوليد عناصر، ووضعها في صِوان، حتى يتم أخذها من قبل المستهلك، حيث يمثل المتحول counter (عدد صحيح) عدد العناصر الحالي في الصِوان، ويبدأ بالصفر وتتم إضافة واحد إليه مع كل إضافة لعنصر جديد على الصِوان)، وفق الخوارزمية التالية:

```
while (1) {
    /* produce an item in nextProduced */
    while (counter == BUFFER_SIZE)
          ; /* do nothing */
    buffer[in] = nextProduced;
    in = (in + 1) % BUFFER_SIZE;
    counter++;
}
```

- تكون مساحة الصِوان الخاص بالتخزين محدودة (BUFFER_SIZE)، ولذلك يقوم المنتج بإنتاج عناصر، طالما أن الصِوان يحوي مساحة فارغة، وإلا فإنه يتوقف عن العمل بانتظار فراغ أحد عناصر المخزن على الأقل.
- بينما يقوم إجراء المستهلك بأخذ العناصر من الصِوان طالما أنه يحتوي على عناصر، وإلا فإنه يتوقف عن العمل بانتظار قدوم عنصر واحد على الأقل إلى الصِوان، حيث يتم في الخوارزمية انقاص قيمة المتحول counter بمقدار واحد مع كل عملية أخذ عنصر من الصِوان، وفق الخوارزمية التالية:

```
while (1) {
    while (counter == 0)
          ; // do nothing
    nextConsumed = buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    counter--;
    /* consume the item in nextConsumed */
}
```

على الرغم من أن إجرائي المنتج والمستهلك صحيحان كلاً على حدة، إلا أنهما قد لا يقومان بسلوك صحيح إذا نفذا على التوازي. سنوضح ذلك بالمثال التالي:

لنفرض أن قيمة المتحول count تساوي 5، وأن إجرائي المنتج والمستهلك تنفذان العبارتين "counter++" و "counter--" على التوازي، تبعاً لذلك يمكن أن يأخذ المتغير counter إحدى القيم 4، 5، 6، والقيمة الوحيدة الصحيحة هي counter = 5، التي تنتج عندما تُنفذ كلٌ من التعليمتين على حده.

3. مشكلة المقطع الحرج (critical section)

ليكن لدينا نظام يحوي n إجراء $\{P_0, P_1, \dots, P_{n-1}\}$، لكل إجراء مقطع رماز يسمى المقطع الحرج، ويمكن للإجراء أن يغير متحولات مشتركة (جدول، ملف،...).

عندما يجري تنفيذ إجراء ضمن المقطع الحرج الخاص به، لا يُسمح لأي إجراء آخر أن يُنفِذ ضمن المقطع الحرج الذي يخصه، وبالتالي يجري تنفيذ المقاطع الحرجة بوساطة الإجراءات باستخدام "الاستبعاد المتبادل"، وبذلك يجري حلّ مشكلة المقطع الحرج خلال تصميم بروتوكول خاص، يُستخدم لوضع آلية تعاون بين الإجراءات عندما تعمل في مقاطعها الحرجة، حيث يتوجب على كل إجراء أن يطلب السماح له بدخول مقطعه الحرج من خلال (مقطع الدخول)، بينما يجري الخروج من المقطع الحرج من خلال (مقطع الخروج).

```
Do {
```

entry section

critical section

exit section

remainder section

```
} while (1);
```

يجري حلّ مشكلة المقطع الحرج عبر توفير المتطلبات التالية:

- الاستبعاد المتبادل: إذا كان إجراء Pi ينفذ ضمن المقطع الحرج الخاص به، فلا يمكن لأي إجراء آخر أن يكون ضمن المقطع الحرج الخاص به.

- التقدم: إذا لم يكن هنالك أي إجراء ينفذ ضمن المقطع الحرج الخاص به، وصدف وجود مجموعة إجراءات أخرى تريد دخول مقاطعها الحرجة، عندئذ يمكن للإجراءات التي توجد خارج مقاطعها الحرجة فقط، أن تشارك في تقرير من هو الإجراء التالي للدخول إلى المقطع الحرج.
- الانتظار المحدود: عند طلب إجراء ما الدخول إلى مقطعه الحرج، فهنالك حد لعدد المرات التي يسمح فيه لإجراءات أخرى بدخول مقاطعها الحرجة قبل تلبية هذا الطلب.
-

4. حلول لمشكلة المقطع الحرج في حالة إجراءين (الخوارزمية 1)

لنفرض وجود إجراءين P_0 و P_1 (أو نرمز إليهم p_i و p_j).

في الخوارزمية 1، لنترك الإجراءين يتشاركان في متحول صحيح مشترك turn يبدأ بالصفر (أو الواحد)، فإذا كان turn == i، عندها يُسمح للإجراء p_i بالتنفيذ في المقطع الحرج الخاص به.

```
Do {
    while (turn != i) ;

    critical section

    turn = j ;

    remainder section

} while (1);
```

يضمن هذا الحل وجود إجراء واحد فقط داخل المقطع الحرج الخاص به، لكنه لا يحقق شرط التقدم، لأنه يتطلب تبديلاً يدوياً ثابتاً للإجراءات لتنفيذ المقطع الحرج. مثلاً إذا كان turn == 0 و p_1 مستعد لدخول المقطع الحرج الخاص به، فإن p_1 لا يستطيع ذلك، حتى ولو لم يكن p_0 في المقطع الحرج.

5. حلول لمشكلة المقطع الحرج في حالة إجراءين (الخوارزمية 2)

تكمن مشكلة الخوارزمية 1 بأنها لا تحتفظ بمعلومات كافية عن حالة كل إجراء، فهي تتذكر فقط الإجراء الذي يُسمح له بدخول المقطع الحرج الخاص به، ولحل هذه المشكلة، يمكن أن نستخدم بدلاً من المتغير turn المصفوفة flag التالية: Boolean flag[2].

تأخذ عناصر المصفوفة القيمة false في البداية، فإذا كان flag[i] يساوي true فإن الإجراء p_i مستعد لدخول المقطع الحرج الخاص به.

في هذه الخوارزمية يقوم الإجراء p_i بإسناد القيمة true إلى flag[i] إشارةً إلى أنه مستعد لدخول المقطع الحرج الخاص به، بعد ذلك، يتحقق p_i أن P_j ليس مستعداً لدخول المقطع الحرج الخاص به، لأنه لو كان مستعد فينبغي على p_i أن ينتظر إشارة من p_j تدل على أنه لم يعد في مقطعه الحرج.

```
Do {
    flag[i] = true;
    while (flag[j]) ;
        critical section
    flag[i] = false;
        remainder section
} while (1);
```

تحقق هذه الخوارزمية شرط الاستبعاد المتبادل، في حين لا تحقق شرط التقدُم: فعلى سبيل المثال، لن يتحقق التقدم إذا نفذ p_1 عملية إسناد flag[0] إلى true ، ونفذ p_0 ينفذ عملية إسناد flag[1] إلى true .

flag[0] = true

flag[1] = true

6. حلول لمشكلة المقطع الحرج في حالة إجراءين (الخوارزمية 3)

تستفيد هذه الخوارزمية من الأفكار الأساسية في الخوارزميتين السابقتين، للحصول على حل صحيح لمشكلة المقطع الحرج، تتحقق فيه المتطلبات الثلاثة.

نُعرِّف: الجدول البولياني flag والمتحول الصحيح turn حيث تكون قيمة كلٍ من flag[0] و flag[1] تساوي false وتكون قيمة turn 0 أو 1.

كي يدخل الإجراء Pi إلى المقطع الحرج، يُسنِد الإجراء القيمة true إلى flag[i] والقيمة j إلى turn، ليتأكد بذلك أنه دور الإجراء الآخر للدخول في حال كان الشرط محققاً لذلك.

إذا حاول الإجراءان الدخول في نفس الوقت، فإن turn سيأخذ القيمتين i و j بنفس الوقت، وإحدى عمليتي الإسناد سوف تبقى، والقيمة الأخيرة التي يأخذها turn (بعمليتي إسناد متتاليتين) هي التي تقرر أي منهما سيسمح له بالدخول أولاً إلى مقطعه الحرج.

```
Do {
    flag[i] = true;
    turn = j;
    while (flag[j] && turn == j);

        critical section

    flag[i] = false;

        remainder section

} while (1);
```

تضمن هذه الخوارزمية تحقق الشروط الثلاثة لمشكلة المقطع الحرج أي، الاستبعاد المتبادل، والتقدم، والانتظار المحدود.

7. حلول في حالة عدة إجراءات (خوارزمية المخبز)

يمكن تلخيص مبدأ خوارزمية المخبز بما يلي:

- عند الدخول إلى المخزن، يحصل كل زبون على رقم، والزبون الحاصل على أصغر رقم هو من يُخدَّم في المرة القادمة.
- في حال حصول إجراءين (زبونين) مختلفين على نفس الرقم، فإن الإجراء ذو الاسم الأصغر هو الذي يُخدَّم أولاً (إذا حصل الإجراءان P_i و P_j على نفس الرقم وكان i<j فإن P_iهو الذي يُخدَّم أولاً).
- تستخدم هذه الخوارزمية بنى المعطيات التالية:
  - مصفوفة من n متحول منطقي choosing
  - مصفوفة من n عدد صحيح number.

  حيث n هو عدد الإجراءات
- يجري في البداية إسناد القيمتين false و 0 إلى المصفوفتين السابقتين على الترتيب.

  int number[n]
- ونُعرِّف فيها العلاقات التالية:

$$(a,b) < (c,d) \quad \text{if} \quad [(a = = c \text{ and } b < d) \text{ or } (a < c)]$$

$$\max(a_0,\ldots,a_{n-1})==k \quad \text{where } (k >= a_i) \quad \text{for } i = 0,\ldots,n-1$$

```
Do {

    choosing[i] = true;
    number[i] = max(number[0], number[1], …, number[n-1]) + 1;
    choosing[i] = false;
    for (j = 0; j < n; j++) {
        while (choosing[j]);
        while (number[j] != 0) &&
          ((number[j],j]) <(number[i],i)));
    }

critical section

    number[i] = 0;

remainder section

} while (1);
```

تحقق هذه الخوارزمية شروط المقطع الحرج.

8. تعليمات التزامن على مستوى العتاديات

- هناك تعليمات عتادية بسيطة متاحة في العديد من النظم، من أجل حل مشكلة المقطع الحرج.
- ففي بيئة أحادية المعالج، يمكن أن تُحلّ مشكلة المقطع الحرج من خلال منع حدوث مقاطعة أثناء تعديل متحول مشترك، مما يعني أن تنفيذ سلسلة من التعليمات الحالية، يجري بالترتيب، وبدون مقاطعة، وبدون إجراء أية تعديلات غير متوقعة على المتحول المشترك.
- أما في بيئة متعددة المعالجات، فتتوفر في العديد من الآلات، تعليمات عتادية خاصة، تسمح إما بتعديل محتوى كلمة، أو التبديل بين محتوى كلمتين، وذلك دون تجزئة.
- نُعرِّف تعليمة TestAndSet (اختبر وأسند)، وهي تعليمة لا تقبل التجزئة، فإذا تم تنفيذ تعليمتي TestAndSet في نفس الوقت (كل منها على معالج مختلف)، فإن تنفيذهما يجري على التسلسل وفق ترتيب معين.
- تعريف تعليمة TestAndSet:

```
boolean TestAndSet (boolean &target) {
    boolean rv = target;
    target = true;
    return rv;
}
```

- من أجل تحقيق الاستبعاد المتبادل باستخدام TestAndSet، يجري التصريح عن متغير بولياني lock، يأخذ القيمة الابتدائية false، وتتغير القيمة وفقاً لحجز المقطع الحرج أو لا.
- خوارزمية تحقيق الاستبعاد المتبادل باستخدام TestAndSet:

```
Do {
    while (TestAndSet(lock));
        critical section
    lock = false;
        remainder section
} while (1);
```

- كذلك، نُعرِّف تعليمة التبديل (swap)، التي تعمل على تبديل محتوى كلمتين من دون تجزئة، ويجري تنفيذ الاستبعاد المتبادل باستخدام swap، عبر التصريح عن متحول بولياني عام lock، يأخذ القيمة false، بالإضافة إلى متحول بولياني محلي key لكل إجراء.
- تعريف تعليمة التبديل swap:

```
boolean Swap (boolean &a, boolean &b) {
    boolean temp = a;
    a = b;
    b= temp;
}
```

- خوارزمية تحقيق الاستبعاد المتبادل باستخدام swap:

```
Do {

    key = true;
    while (key == true);
       swap(lock,key)

        critical section

    lock = false;

        remainder section

} while (1);
```

- خوارزمية تحقق جميع متطلبات المقطع الحرج باستخدام تعليمة TestAndSet:

```
Do {
```

```
waiting[i] = true;
key = true;
while (waiting[i] && key)
    key = TestAndSet(lock);
waiting[i] = false;
```

critical section

```
j = (i+1) % n;
while ((j != i) && !waiting[j])
    j = (i+1) % n;
if (j == i)
    lock = false;
else
    waiting[j] = false;
```

remainder section

```
} while (1);
```

ولكن الخوارزميتين السابقتين لا تحققان شرط الانتظار المحدود، ومن أجل تحقيق ذلك يجري تعريف خوارزمية تحقيق جميع متطلبات المقطع الحرج، باستخدام تعليمة TestAndSet وبنية معطيات مشتركة مؤلفة من جدول بولياني waiting (بطول n حيث n عدد الإجراءات)، ومتحول بولياني lock. تأخذ بنى المعطيات الآنفة الذكر، القيمة الابتدائية false.

9. سيمافور

- ليس سهلاً في المسائل المعقدة، تعميم حلول مشكلة المقطع الحرج، ولذلك يمكننا اللجوء إلى أداة لتحقيق التزامن ندعوها سيمافور.
- نُعرِّف السيمافور بأنه عبارة عن متحول صحيح S، يجري النفاذ إليه من خلال عمليتين قياسيتين غير قابلتين للتجزئة wait (انتظر) و signal (أرسل).
- تعريف التعليمة wait.

```
wait (S) {
    while (S ≤ 0)
    ;//no-op
    S--;
}
```

- تعريف التعليمة signal.

```
signal (S) {
    S++;
}
```

- يجب أن تكون التعديلات التي تُجريها العمليتان wait و signal على قيمة المتغير الصحيح الخاص بالسيمافور، غير قابلة للتجزئة، أي إذا كان أحد الإجراءات يعدّل قيمة السيمافور، فيجب ألا يتمكن أي إجراء آخر من تعديل قيمة السيمافور نفسها في نفس الوقت.
- يمكن استخدام السيمافور من أجل حل مشكلة المقطع الحرج في حالة n إجراء، من خلال تشارك هذه الإجراءات في نفس السيمافور وبحيث يأخذ المتحول mutex القيمة الابتدائية 1.

10. التحقيق البرمجي للسيمافور

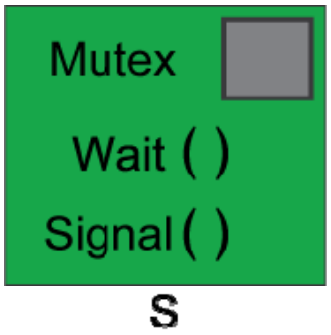
- تكمن السيئة الرئيسية لحلول الاستبعاد المتبادل المذكورة باستخدام السيمافور، في أنها تستخدم انتظاراً مشغولاً، فأثناء وجود إجراء في المقطع الحرج الخاص به، يجب على كل إجراء آخر يحاول دخول المقطع الحرج الخاص به، أن يدور في حلقة ضمن مقطع الدخول، وهذا الدوران يستهلك دورات وحدة المعالجة، التي يمكن استخدامها بعمل مفيد لإجراء آخر.
- لحل المشكلة السابقة يمكننا تعديل تعريف عمليتي wait و signal. فعندما يُنفِذ إجراء عملية wait وتكون قيمة السيمافور غير موجبة، فإن على هذا الإجراء أن ينتظر، وبدلاً من الانتظار المشغول، يتوقف الإجراء وينتظر في رتل انتظار خاص بالسيمافور وتتبدل حالته إلى حالة الانتظار.
- يتابع الإجراء المتوقف عمله عندما ينفذ إجراء آخر عملية signal، وذلك من خلال عملية wakeup (إيقاظ).

نُعرِّف السيمافور من خلال تسجيلة تحوي عنصرين: الأول value يحتوي على قيمة السيمافور، والثاني L يمثل رتلاً من الإجراءات المنتظِرة. يتناسب هذا التعريف الجديد مع المتطلبات الجديدة التي نسعى لتحقيقها.

كما نعيد تعريف العمليتين wait و signal لتأخذان بالحسبان التعريف الجديد للسيمافور. حيث تؤدي عملية block إلى توقف الإجراء الذي ينفذها، وتقوم العملية wakeup(p) بمتابعة تنفيذ الإجراء المتوقف p.

- تعريف بنية السيمافور.

```
typedef struct {
    int value;
    struct process *L;
} semaphore;
```



- التعريف الجديد لعملية wait (يتوقف الإجراء وينتظر في رتل انتظار خاص بالسيمافور وتتبدل حالته إلى حالة الانتظار).

```
void wait(semaphore S) {
    S.value--;
    if (S.value < 0) {
        add this process to S.L;
        block();
    }
}
```

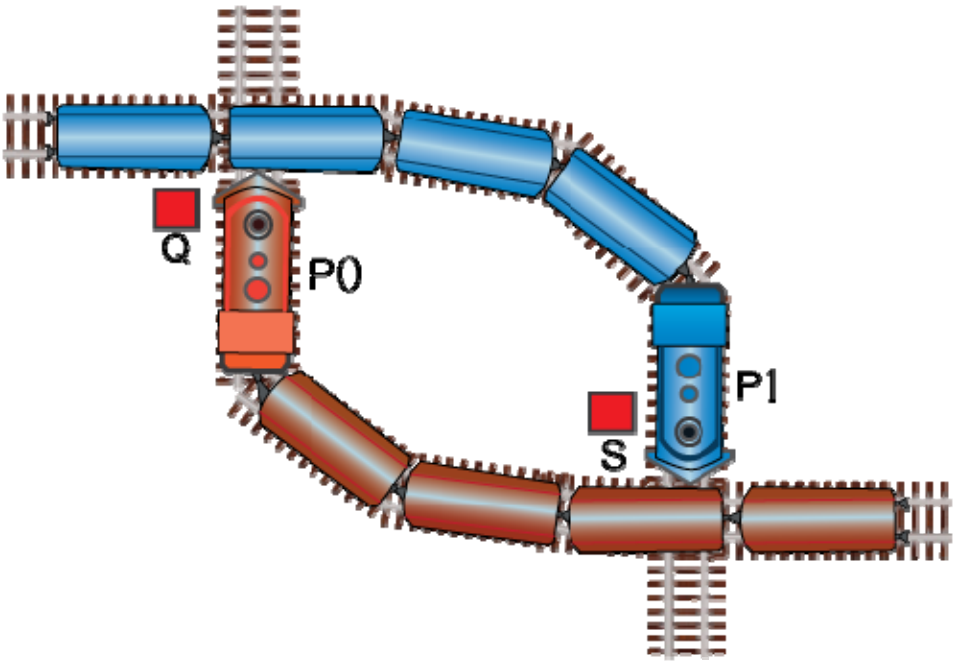
- التعريف الجديد لعملية signal.

```
void signal(semaphore S) {
    S.value++;
    if (S.value <= 0) {
        remove a process P from S.L;
        wakeup(P);
    }
}
```

11. الإقفال المتبادل

- يمكن أن يؤدي استخدام رتل الانتظار في تحقيق السيمافور إلى وضعٍ نجد فيه إجرائين أو أكثر في حالة انتظار لانهائي لحدث معين، وبحيث لا يمكن لهذا الحدث أن يتم إلا بتشغيل واحد من الإجراءات المنتظرة (الحدث المشار إليه هنا هو عملية signal لأحد السيمافور الموجودين في حالة انتظار).
- عندما تقع مثل هذه الحالة، نقول إن هذه الإجراءات قد وصلت إلى حالة الإقفال المتبادل.
- لتوضيح هذه الحالة، نأخذ نظام مؤلف من إجرائين P_0 و P_1 يريدان الدخول إلى السيمافور، S وإلى السيمافور، Q.
- لنفرض أن P_0 ينفذ wait(S) وأن P_1 ينفذ wait(Q)، وعندما ينفذ P_0 wait(Q)، يجب أن ينتظر أن ينفذ P_1 العملية signal(Q)، وبالمثل عندما ينفذ P_1 العملية wait(S)، يجب أن ينتظر أن ينفذ P_0 العملية signal(S)، ولأنه لا يمكن تنفيذ هاتين العمليتين، يتواجد P_0 و P_1 في حالة إقفال متبادل.
- مثال:

P_0	P_1
`wait(S);`	`wait(Q);`
`wait(Q);`	`wait(S);`
.	.
.	.
.	.
`signal(S);`	`signal(Q);`
`signal(Q);`	`signal(S);`



12. مشكلات التزامن – مشكلة المنتج والمستهلك (producer consumer)

- نفترض وجود مجموعة تحوي n صِوان، يتسع كل واحد إلى عنصر واحد. كما نستخدم السيمافور mutex لتوفير وظيفة الاستبعاد المتبادل أثناء النفاذ إلى مجموعة لصِوانات، حيث تأخذ mutex القيمة الابتدائية 1.
- نستخدم أيضاً السيمافور empty والسيمافور full، من أجل عدد الأصوِنة الفارغة والممتلئة على الترتيب، كما تُسند القيمتين n و 0 لهما على الترتيب أيضاً.

- بنية إجراء المنتج.

```
Do {
        ...
    produce an item in nextp
        ...
    wait(empty);
    wait(mutex);
        ...
    add nextp buffer to
        ...
    signal(mutex);
    signal(full);
} while(1);
```

- بنية إجراء المستهلك.

```
Do {
    wait(full);
    wait(mutex);
        ...
    remove an item from buffer to nextc
        ...
    signal(mutex);
    signal(empty);
        ...
    consume the item in nextc
        ...
} while(1);
```

13. مشكلة القراء والكتاب (readers writers)

- تتشارك الإجراءات المتوازية في أغراض المعطيات، وتجري عليها عمليات قراءة أو كتابة، ومن هنا نميز نوعين من الإجراءات القرّاء والكتّاب، إذا قام قارئان بالنفاذ إلى غرض مشترك في نفس الوقت، فليس هنالك من مشكلة، في حين إذا قام كاتب وإجراءات أخرى (قرّاء أو كتّاب) بالنفاذ إلى الغرض التشاركي فقد تحدث مشكلة، لذلك سنجعل الوصول إلى الغرض التشاركي بالنسبة للكتّاب، وصول حصري (أي يمنع أي إجراء آخر من الوصول إلى هذا الغرض).
- تتشارك إجراءات القرّاء ببنى المعطيات التالية: السيمافور، mutex و wrt بالإضافة إلى متحول صحيح readcount.
- تبدأ السيمافور mutex والسيمافور wrt بالقيمة 1، كما يبدأ المتغير readcount بالقيمة 0.
- يُستخدم السيمافور mutex لضمان الاستبعاد المتبادل عندما يجري تحديث المتغير readcount، ويُستخدم السيمافور wrt (وهو عبارة عن متحول مشترك بين إجراءات الكتّاب والقرّاء)، للاستبعاد المتبادل بين الكتّاب، كما يستخدمه القارئ الأول والأخير الذي يدخل المقطع الحرج أو يخرج منه.

- بنية إجراء الكاتب.

```
wait(wrt);
...
writing is performed
...
signal(wrt);
```

- نلاحظ من خوارزميات القرّاء والكتّاب، أنه إذا وُجِد كاتب في المقطع الحرج، وكان عدد القراء الذين ينتظرون يساوي n، فإن قارئاً واحداً سوف يدخل في رتل wrt، و $n-1$ قارئاً سوف يدخل في رتل mutex.
- بنية إجراء القارئ.

```
wait(mutex);
readcount++
if (readcount == 1)
    wait(wrt);
signal(mutex);
    ...
    reading is performed
    ...
wait(mutex);
readcount--;
if (readcount == 0)
    signal(wrt);
signal(mutex);
```

- كما نلاحظ أنه عندما ينفذ كاتب العملية signal(wrt) فإنه يمكن متابعة تنفيذ إما جميع القراء الذين ينتظرون وإما كاتب واحد ينتظر.

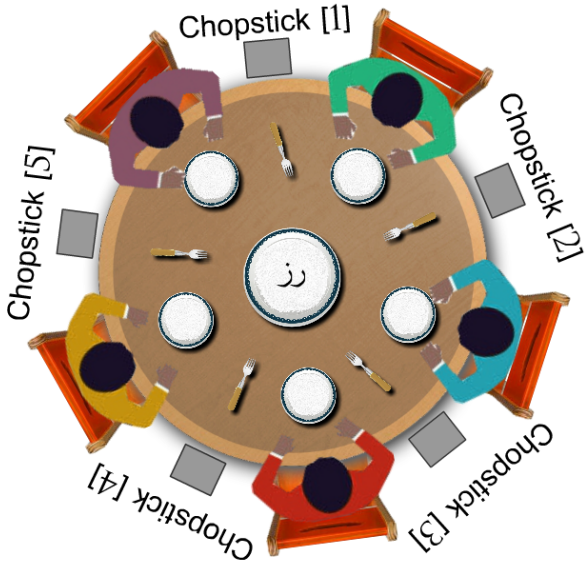
14. مشكلة الفلاسفة المفكرين والطاعمين (dining philosophers)

لنأخذ خمسة فلاسفة يقضون حياتهم في التفكير وتناول الطعام، حيث يتشارك الفلاسفة في طاولة دائرية يحيط بها خمسة كراسي (كرسي لكل فيلسوف)، وفي وسط الطاولة صحن أرز، ويوجد على الطاولة خمس أعواد. عندما يفكر الفيلسوف لا يتفاعل مع زملائه، وعندما يشعر بالجوع يحاول التقاط أقرب عودين إليه (العودين اليميني واليساري)، لا يستطيع الفيلسوف سوى التقاط عود واحد في وقت واحد، كما لا يستطيع أن يلتقط العود الموجود في يد جاره.

عندما يحصل الفيلسوف على عوديه يستطيع أن يأكل، على أن يتخلى عن عوديه عندما ينتهي.

تُعتبر هذه المسألة أحد المشاكل التقليدية في التزامن، ويعتمد حل المشكلة على تمثيل كل عود بسيمافور حيث يحاول الفيلسوف التقاط عود بتنفيذ عملية wait على ذلك السيمافور، ويحرر عوداً بتنفيذ عملية signal. يجري تمثيل المعطيات المشتركة في مصفوفة chopstick من خمسة سيمافورات وتأخذ جميع عناصر المصفوفة القيمة الابتدائية 1.

يضمن هذا الحل ألا يأكل جاران في نفس الوقت، لكنه لا يحل مشكلة الإقفال المتبادل الذي يسبب مجاعة (في حال جاع الفلاسفة جميعاً في نفس الوقت والتقط كل منهم أحد الأعواد على جانبيه).



- بنية إجراء الفيلسوف i.

```
Do {
    wait(chopstick[i]);
    wait(chopstick[i+1] % 5);
      ...
      eat
      ...
    signal(chopstick[i]);
    signal(chopstick[i+1] % 5);
      ...
      think
      ...
} while (1);
```

هنالك عدة أفكار لحل مشكلة الإقفال المتبادل:

- يُسمح لأربعة فلاسفة على الأكثر بالجلوس إلى الطاولة في نفس الوقت
- يُسمح لفيلسوف بالتقاط عوديه إذا كانا متوفران معاً
- يُسمح بحل لا متناظر يلتقط فيه الفيلسوف ذو الرقم الفردي عوده اليساري ومن ثم اليميني، في حين يلتقط الفيلسوف ذو الرقم الزوجي عوده اليميني ومن ثم اليساري

15.المراقب

يُعتبر المراقب بنية أخرى لتحقيق التزامن بين الإجراءات، وحل مشكلات المقطع الحرج.

يتألف المراقب من مجموعة من المتحولات (متحولات المراقب)، ومجموعة من الإجرائيات التي تنفذ العمليات على هذه المتحولات، حيث لا يمكن الوصول إلى المتحولات إلا من خلال هذه الإجرائيات، كما لا يمكن لهذه الإجرائيات إلا النفاذ لهذه المتحولات.

تضمن هذه البنية ألا يوجد سوى إجراء واحد فعال داخل المراقب في لحظة معينة.

16. نشاط: مسألة الحلاق النائم (sleeping barber)

- لدينا صالون للحلاقة يحوي على n كرسي للانتظار، بالإضافة إلى كرسي للحلاق.
- يمكن للحلاق أن ينام في حال عدم وجود زبائن، ولكن إذا دخل زبون إلى صالون الحلاقة، وكانت جميع الكراسي مشغولة، فإن الزبون يغادر الصالون.
- إذا كان الحلاق مشغولاً وكان هنالك كراسي فارغة، جلس الزبون في أحد الكراسي، وفي حال كان الحلاق نائماً، فإن الزبون يُوقظ الحلاق.
- اكتب برنامج ينظم عمل الحلاق والزبائن.

الحل:

```
#define CHAIRS 5  /* # chairs for waiting customers * /
typedef int semaphore; / *use your imagination * /
semaphore customers = 0; / *# of customers waiting for service*/
semaphore barbers = 0; / * # of barbers waiting for customers * /
semaphore mutex = 1; / * for mutual exclusion * /
int waiting = 0; / * customers are waiting (not being cut) * /

void barber(void)
{
while (TRUE) {
down(customers); / * go to sleep if # of customers is 0 * /
down(mutex); / * acquire access to 'waiting' * /
waiting = waiting - 1; / * decrement count of waiting customers * /
up(barbers); / * one barber is now ready to cut hair * /
up(mutex); / * release 'waiting' * /
cut

hair(); / * cut hair (outside critical region) * /
}
}


void customer(void)
{
down(mutex); / * enter critical region * /
if (waiting < CHAIRS) { / * if there are no free chairs, leave * /
waiting = waiting + 1; / * increment count of waiting customers * /
up(customers); / * wake up barber if necessary * /
up(mutex); / * release access to 'waiting' * /
down(barbers); / * go to sleep if # of free barbers is 0 * /
get

haircut(); / * be seated and be serviced * /
} else {
up(mutex); / * shop is full; do not wait * /
}
```

}

17.نشاط: مسألة مدخِّنات السجائر (cigarette smokers)

- لدينا ثلاثة إجراءات مدخِّنة وإجراء وكيل، حيث يقوم كل إجراء مدخِّن بلف سيجارة ثم تدخينها.
- يحتاج الإجراء المدخِّن إلى ثلاث مكونات: (تبغ، وورق، وأعواد ثقاب).
- يمتلك أحد الإجراءات المدخِّنة الورق، و يمتلك آخر التبغ، و يمتلك الثالث أعواد الثقاب، بينما يمتلك الإجراء الوكيل مخزون من المكونات الثلاثة.
- يضع الوكيل مكونين من ثلاثة على الطاولة، حيث يقوم الإجراء المدخن الذي يمتلك المكون الناقص بلف سيجارة وتدخينها، ومن ثم يقوم الوكيل بوضع مكونين من جديد على الطاولة وهكذا تستمر الحلقة.
- اكتب برنامج لمزامنة الإجراءات المدخِّنة والإجراء الوكيل.

18. التمارين:

1. **يمكن أن تتشارك الإجراءات المتعاونة مباشرةً في فضاء العنونة المنطقية (الرماز والمعطيات):**
   A. **صح**
   B. **خطأ**

2. **على الرغم من أن إجرائي المنتج والمستهلك صحيحان كلاً على حدة، إلا أنهما يقومان بسلوك صحيح إذا نفذا على التوازي:**
   A. صح
   B. خطأ

3. يجري حلّ مشكلة المقطع الحرج عبر توفير المتطلبات التالية:
   A. التقدم
   B. الاستبعاد المتبادل
   C. الانتظار المحدود
   D. جميع الإجابات صحيحة

4. تعليمة TestAndSet (اختبر وأسند)، وهي تعليمة تقبل التجزئة:
   A. صح
   B. خطأ

5. نُعرِّف السيمافور بأنه عبارة عن متحول صحيح S، يجري النفاذ إليه من خلال عمليتين قياسيتين غير قابلتين للتجزئة wait (انتظر) و signal (أرسل):
   A. صح
   B. خطأ

6. يتألف المراقب من مجموعة من المتحولات (متحولات المراقب)، ومجموعة من الإجرائيات التي تنفذ العمليات على هذه المتحولات:
   A. صح
   B. خطأ

رقم التمرين	الإجابة الصحيحة
.1	(A)
.2	(B)
.3	(D)
.4	(B)
.5	(A)
.6	(A)