OPERATING SYSTEM

Lecture Notes

Dr. Professor, J.M. Khalifeh

قسم المعلوماتية

الوحدة الثانية

النسخة العربية

Unit-2

Operating-System Services and Structures

ملخص

يوفر نظام التشغيل البيئة التي يتم من خلالها تنفيذ البرامج. تختلف أنظمة التشغيل داخليًا اختلافًا كبيرًا في تركيبها، حيث يتم تنظيمها وفقًا للعديد من الخطوط المختلفة. يعد تصميم نظام تشغيل جديد مهمة رئيسية، ومن المهم أن تكون أهداف النظام محددة جيدًا قبل أن يبدأ التصميم، حيث تشكل هذه الأهداف الأساس للاختيارات بين الخوارزميات والاستراتيجيات المختلفة.

يمكننا عرض نظام التشغيل من عدة وجهات نظر. تركز إحدى وجهات النظر على الخدمات التي يوفرها النظام؛ بينما يعتمد آخر، على الواجهة التي يتيحها للمستخدمين والمبرمجين ؛ والثالث على مكوناته وترابطه. في هذه الوحدة، نستكشف الجوانب الثلاثة لأنظمة التشغيل، ونعرض وجهات نظر المستخدمين والمبرمجين ومصممي أنظمة التشغيل. وسنأخذ في الاعتبار الخدمات التي يوفرها نظام التشغيل، وكيف يتم تقديمها، وكيف يتم تصحيحها، وما هي المنهجيات المختلفة لتصميم مثل هذه الأنظمة. أخيرًا، نصف كيفية إنشاء أنظمة التشغيل وكيف يبدأ الكمبيوتر نظام التشغيل الخاص به.

أهداف الوحدة

- تحديد الخدمات التي يقدمها نظام التشغيل.
- توضيح كيفية استخدام استدعاءات النظام لتوفير خدمات نظام التشغيل.
- المقارنة بين الاستراتيجيات المتجانسة، والطبقات، والنواة الدقيقة، والوحدات، والهجين لتصميم أنظمة التشغيل.

خدمات نظام التشغيل

يوفر نظام التشغيل خدمات لكل من المستخدمين والبرامج نذكر منها:

تنفيذ البرنامج Program execution

تتعامل أنظمة التشغيل مع العديد من الأنشطة من برامج المستخدم إلى برامج النظام مثل التخزين المؤقت للطابعة وخوادم الأسماء وخوادم الملفات وما إلى ذلك. يتم تنفيذ كل من هذه الأنشطة كسلسلة من العمليات.

تتضمن العملية سياق التنفيذ الكامل (التعليمات البرمجية للتنفيذ، والبيانات التي يجب معالجتها، والسجلات، وموارد نظام التشغيل قيد الاستخدام). فيما يلي الأنشطة الرئيسية لنظام التشغيل فيما يتعلق بإدارة البرنامج:

- تحميل البرنامج في الذاكرة.
- تنفيذ البرنامج.
- معالجة تنفيذ البرنامج.
- توفير آلية لمزامنة العملية.
- توفير آلية لعملية الاتصال.
- توفير آلية للتعامل مع المأزق.

عملية الإدخال/الإخراج I/O Process

يتكون نظام الإدخال/الإخراج من أجهزة الإدخال/الإخراج وبرامج التشغيل المقابلة لها. يتضمن برنامج تشغيل الجهاز بعض خصوصيات الأجهزة حسب المستخدمين.

يدير نظام التشغيل الاتصال بين المستخدم وبرامج تشغيل الجهاز.

- تعني عملية الإدخال/الإخراج قراءة أو كتابة عملية باستخدام أي ملف أو أي جهاز إدخال/إخراج محدد.
- يوفر نظام التشغيل الوصول إلى جهاز الإدخال/الإخراج المطلوب عند الحاجة.

التعامل مع نظام الملفات Dealing with the file system

يمثل الملف مجموعة من المعلومات ذات الصلة. يمكن لأجهزة الكمبيوتر تخزين الملفات على القرص (التخزين الثانوي) لغرض التخزين طويل المدى. تتضمن أمثلة وسائط التخزين الشريط المغناطيسي والأقراص المغناطيسية ومحركات الأقراص الضوئية مثل الأقراص المضغوطة وأقراص DVD. كل من هذه الوسائط لها خصائصها الخاصة مثل السرعة والسعة ومعدل نقل البيانات وطرق الوصول إلى البيانات.

عادة ما يتم ترتيب نظام الملفات في أدلة لسهولة التصفح والاستخدام. قد تحتوي هذه الأدلة على ملفات وتوجيهات أخرى. فيما يلي الأنشطة الرئيسية لنظام التشغيل فيما يتعلق بإدارة الملفات:

- يحتاج البرنامج إلى قراءة ملف أو كتابة ملف.
- يمنح نظام التشغيل الإذن للبرنامج للعمل في الملف. حيث يختلف الإذن من للقراءة فقط، للقراءة والكتابة، أو مرفوض وما إلى ذلك.
- يوفر نظام التشغيل واجهة للمستخدم لإنشاء/حذف الملفات.
- يوفر نظام التشغيل واجهة للمستخدم لإنشاء/حذف الدلائل.
- يوفر نظام التشغيل واجهة لإنشاء نسخة احتياطية لنظام الملفات.

التواصل Communication

في حالة الأنظمة الموزعة التي هي عبارة عن مجموعة من المعالجات التي لا تشترك في الذاكرة أو الأجهزة الطرفية أو الساعة، فإن نظام التشغيل يدير الاتصالات بين جميع العمليات. تتواصل العمليات المتعددة مع بعضها البعض من خلال خطوط الاتصال في الشبكة.

يتعامل نظام التشغيل مع استراتيجيات التوجيه والاتصال، ومشكلات الخلاف والأمن. فيما يلي الأنشطة الرئيسية لنظام التشغيل فيما يتعلق بالاتصالات:

- تتطلب عمليتان غالبًا نقل البيانات بينهما
- يمكن أن تكون كلتا العمليتين على جهاز كمبيوتر واحد أو على أجهزة كمبيوتر مختلفة، ولكنهما متصلتان عبر شبكة كمبيوتر.
- يمكن تنفيذ الاتصال بطريقتين، إما عن طريق الذاكرة المشتركة أو عن طريق تمرير الرسائل.

معالجة الأخطاء Error Handling

يمكن أن تحدث الأخطاء في أي وقت وفي أي مكان. قد يحدث خطأ في وحدة المعالجة المركزية، في أجهزة الإدخال/الإخراج أو في أجهزة الذاكرة. فيما يلي الأنشطة الرئيسية لنظام التشغيل فيما يتعلق بمعالجة الأخطاء:

- يتحقق نظام التشغيل باستمرار من الأخطاء المحتملة.
- يتخذ نظام التشغيل الإجراء المناسب لضمان الحوسبة الصحيحة والمتسقة.

إدارة الموارد Resource Management

في حالة وجود بيئة متعددة المستخدمين أو متعددة المهام، يتم تخصيص موارد مثل الذاكرة الرئيسية ودورات وحدة المعالجة المركزية وتخزين الملفات لكل مستخدم أو لكل وظيفة. فيما يلي الأنشطة الرئيسية لنظام التشغيل فيما يتعلق بإدارة الموارد:

- يدير نظام التشغيل جميع أنواع الموارد باستخدام أدوات الجدولة.
- يتم استخدام خوارزميات جدولة لتنفيذ ذلك كاستخدام خوارزميات جدولة وحدة المعالجة المركزية لتحسين استخدام وحدة المعالجة المركزية.

الحماية Protection

بالنظر إلى وجود نظام كمبيوتر به عدة مستخدمين وتنفيذه المتزامن لعمليات متعددة، يجب حماية العمليات المختلفة من أنشطة بعضها البعض.

تشير الحماية إلى آلية أو طريقة للتحكم في وصول البرامج أو العمليات أو المستخدمين إلى الموارد المحددة بواسطة نظام الكمبيوتر. فيما يلي الأنشطة الرئيسية لنظام التشغيل فيما يتعلق بالحماية:

- يضمن نظام التشغيل التحكم في الوصول إلى موارد النظام بالكامل.
- يضمن نظام التشغيل حماية أجهزة الإدخال/الإخراج الخارجية من محاولات الوصول غير الصالحة.
- يوفر نظام التشغيل ميزات المصادقة لكل مستخدم عن طريق كلمات المرور.

واجهة مستخدم أنظمة التشغيل User interface

تسهل واجهة المستخدم (UI) الاتصال بين التطبيق ومستخدمه من خلال العمل كوسيط بينهما. يتم تزويد كل تطبيق بما في ذلك نظام التشغيل بواجهة مستخدم محددة للاتصال الفعال. تتمثل الوظيفتان الأساسيتان لواجهة مستخدم التطبيق في أخذ المدخلات من المستخدم وتوفير المخرجات للمستخدمين. ومع ذلك، قد تختلف أنواع المدخلات التي تتخذها واجهة المستخدم وأنواع المخرجات التي توفرها واجهة المستخدم من تطبيق إلى آخر.

يمكن تصنيف واجهة المستخدم لأي نظام تشغيل إلى أحد الأنواع التالية:

- واجهة المستخدم الرسومية (GUI)
- واجهة مستخدم سطر الأوامر (CLI)

واجهة المستخدم الرسومية Graphical user interface (GUI)

واجهة المستخدم الرسومية هي نوع من واجهة المستخدم الرسومية التي تمكن المستخدمين من التفاعل مع نظام التشغيل عن طريق عمليات التأشير والنقر أو اللمس. تحتوي واجهة المستخدم الرسومية على العديد من الرموز التي تعبر عن التمثيل الرسومي للمتغيرات مثل الملف والدليل والجهاز. يمكن معالجة الرمز الرسومي المتوفر في واجهة المستخدم من قبل المستخدمين باستخدام جهاز تأشير مناسب مثل الماوس وشاشة اللمس والقلم الضوئي. يمكن أيضًا استخدام أجهزة الإدخال الأخرى مثل لوحة المفاتيح لمعالجة هذه الرموز الرسومية. تعتبر واجهات المستخدم الرسومية واجهة سهلة الاستخدام للغاية لأن كل كائن يتم تمثيله بأيقونة مقابلة. على عكس واجهات المستخدم الأخرى، لا يحتاج المستخدمون إلى توفير أمر نصي لتنفيذ المهام.

بعض مزايا نظام التشغيل القائم على واجهة المستخدم الرسومية

- واجهة المستخدم الرسومية سهلة الفهم وحتى المستخدمين الجدد يمكنهم العمل عليها بمفردهم.
- تقر واجهة المستخدم الرسومية بشكل مرئي وتؤكد كل نوع من الأنشطة التي يقوم بها المستخدمون. على سبيل المثال، عندما يحذف المستخدم ملفًا في نظام التشغيل Windows، يطلب نظام التشغيل التأكيد قبل حذفه.
- تمكن واجهة المستخدم الرسومية من أداء عدد من المهام في نفس الوقت. تُعرف ميزات نظام التشغيل هذه أيضًا باسم تعدد المهام.

واجهة سطر الأوامر Command line interface (CLI)

واجهة سطر الأوامر هي نوع من واجهة المستخدم التي تمكن المستخدمين من التفاعل مع نظام التشغيل عن طريق إصدار بعض الأوامر المحددة. لأداء مهمة في هذه الواجهة، يحتاج المستخدم إلى كتابة أمر في سطر الأوامر. عندما يقوم المستخدم بإدخال المفتاح، يتلقى مترجم سطر الأوامر أمرًا. البرنامج المسؤول عن استلام ومعالجة الأوامر الصادرة عن المستخدم. بعد معالجة الأمر يسمى مترجم سطر الأوامر، يعرض مترجم سطر الأوامر موجه الأوامر مرة أخرى مع إخراج الأمر السابق الذي أصدره المستخدم. عيوب CLI هي أن المستخدم يحتاج إلى تذكر الكثير للتفاعل مع نظام التشغيل. لذلك لا تعتبر هذه الأنواع من الواجهة مريحة للغاية من وجهة نظر المستخدمين وتحتاج إلى معرفة مسبقة ببنية وترتيب وسياق ما يحويه سطر الأوامر.

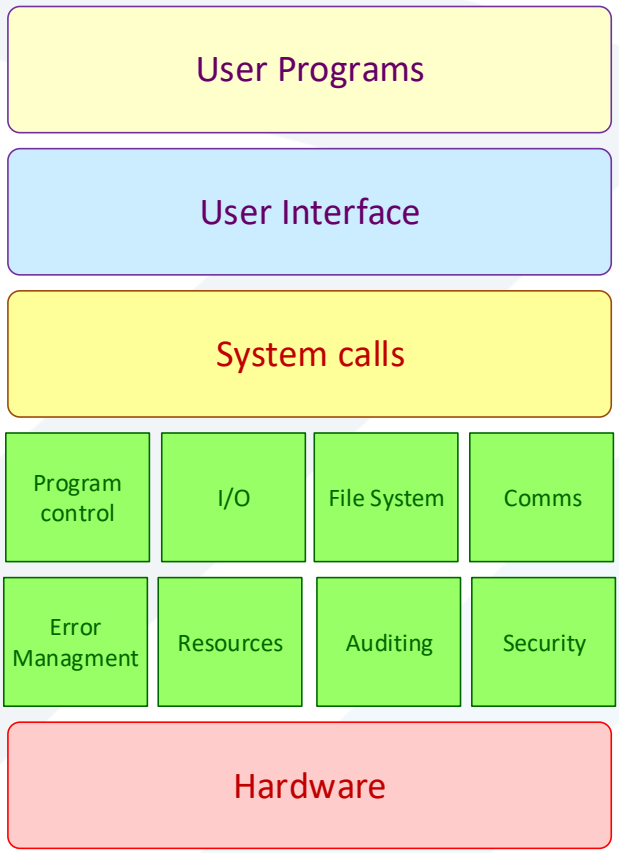
الفرق بين GUI و CUI

Terms	GUI	CUI
Interaction	باستخدام رسومات مثل الصور والرموز.	باستخدام أوامر نصية.
Navigation	التنقل سهل.	التنقل صعب.
Precision	منخفضة الدقة.	CUI لديها دقة عالية.
Usage	سهلة الاستخدام.	يصعب الاستخدام ويتطلب الخبرة.
Speed	منخفضة السرعة.	عالي السرعة.
Peripherals used	يتطلب نظام تشغيل لوحة مفاتيح أو فأرة أو أي جهاز تأشير آخر.	يتطلب نظام التشغيل لوحة المفاتيح فقط.
Memory requirement	يتطلب ذاكرة عالية.	يتطلب ذاكرة منخفضة.
Flexibility	عالية المرونة.	أقل مرونة.
Customize	قابلة للتخصيص بدرجة كبيرة.	لا يمكن تغيير المظهر بسهولة.

استدعاءات النظام System Calls

ما هي استدعاءات النظام؟

يتم توفير التواصل بين العملية ونظام التشغيل من خلال استدعاءات النظام. بشكل عام، تتوفر استدعاءات النظام كتعليمات في لغة التجميع. يتم تضمينها أيضًا في الكتيبات المستخدمة من قبل المبرمجين على مستوى التجميع. يتم إجراء استدعاءات النظام عادةً عندما تتطلب عملية في وضع المستخدم الوصول إلى مورد. ثم يطلب من النواة توفير المورد عبر استدعاء النظام. ويبين الشكل 1 موقع استدعاءات النظام بين مكونات النظام.



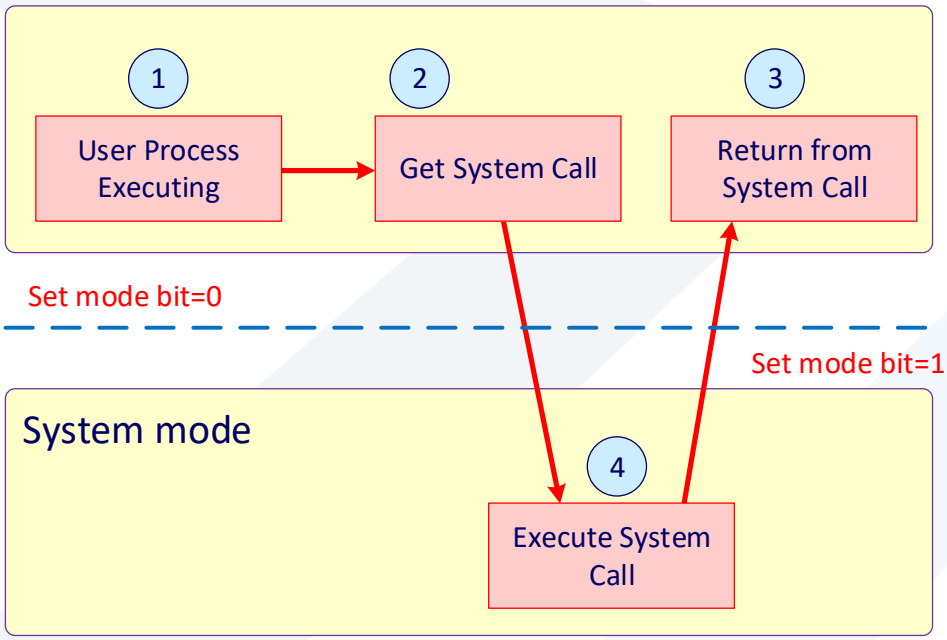
الشكل 1: موقع استدعاءات النظام بين مكونات النظام

العمليات في الوضع المزدوج Dual-mode operations

يمكن أن يؤثر خطأ ما في أحد البرامج سلبًا على العديد من العمليات، وقد يؤدي إلى تعديل بيانات برنامج آخر، أو قد يؤثر أيضًا على نظام التشغيل. على سبيل المثال، إذا كانت العملية عالقة في الحلقة اللانهائية، فإن هذه الحلقة اللانهائية يمكن أن تؤثر على التشغيل الصحيح للعمليات الأخرى. لضمان التنفيذ السليم لنظام التشغيل، هناك طريقتان للتشغيل:

وضع المستخدم

عندما يتم تشغيل نظام الكمبيوتر بواسطة تطبيقات المستخدم مثل إنشاء مستند نصي أو استخدام أي برنامج تطبيق، يكون النظام في وضع المستخدم. عندما يطلب تطبيق المستخدم خدمة من نظام التشغيل أو تحدث مقاطعة أو استدعاء النظام، فسيكون هناك انتقال من المستخدم إلى وضع kernel لتلبية الطلبات.



الشكل 2: تنفيذ العمليات في الوضع المزدوج

ملاحظة: للتبديل من وضع kernel إلى وضع المستخدم، يجب أن تكون بت تمييز الوضع 1.

وضع Kernel

عند بدء تشغيل النظام، يبدأ الجهاز في وضع kernel وعندما يتم تحميل نظام التشغيل، فإنه يبدأ تطبيق المستخدم في وضع المستخدم. لتوفير الحماية للأجهزة، لدينا تعليمات مميزة يتم تنفيذها فقط في وضع kernel. إذا حاول المستخدم تشغيل التعليمات المميزة في وضع المستخدم، فسوف يتعامل مع التعليمات على أنها غير قانونية وتعامل نظام التشغيل.

بعض التعليمات المميزة هي:

1. معالجة المقاطعات
2. التبديل من وضع المستخدم إلى وضع kernel.
3. إدارة المدخلات والمخرجات.

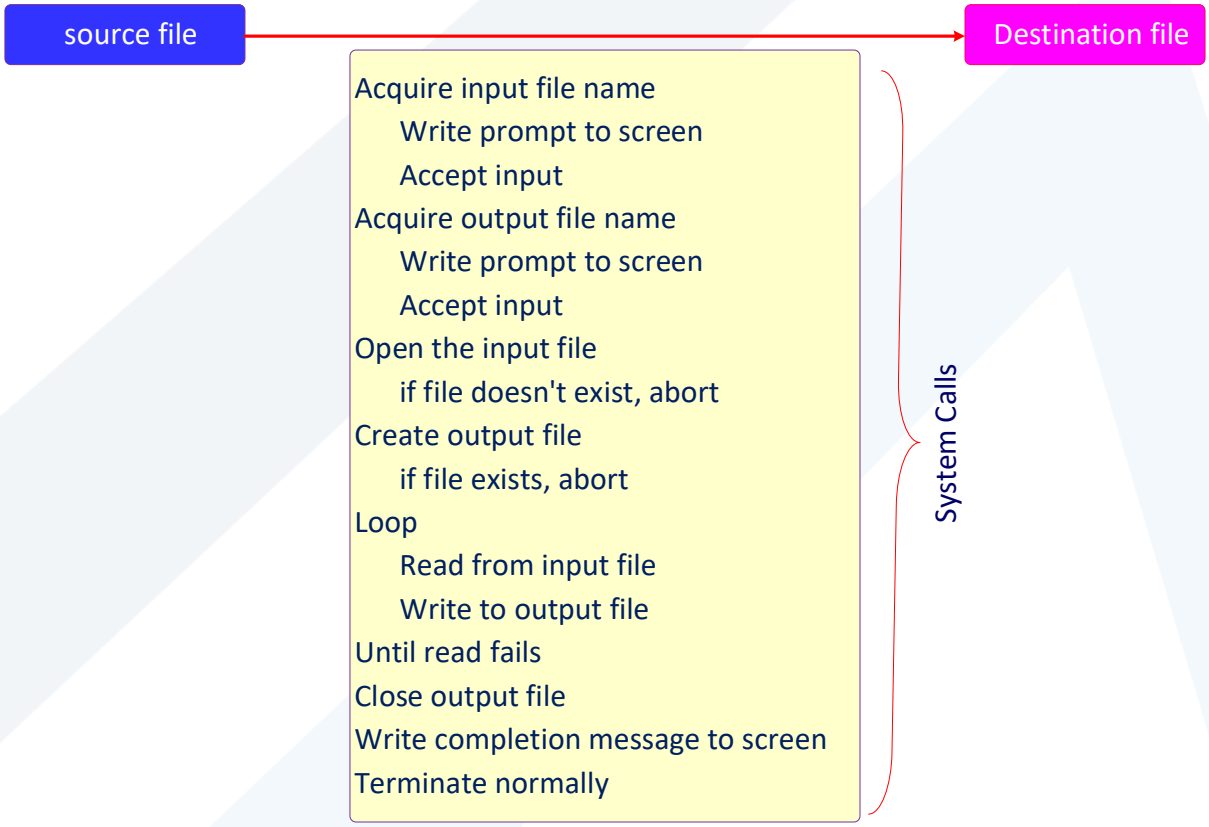
كما يتضح الشكل 2، يتم تنفيذ العمليات بشكل طبيعي في وضع المستخدم حتى يقاطع استدعاء النظام هذا. ثم يتم تنفيذ استدعاء النظام على أساس الأولوية في وضع kernel. بعد تنفيذ استدعاء النظام، يعود عنصر التحكم إلى وضع المستخدم ويمكن استئناف تنفيذ عمليات المستخدم.

بشكل عام، استدعاءات النظام مطلوبة في المواقف التالية:

- إذا كان نظام الملفات يتطلب إنشاء أو حذف الملفات.
- تتطلب القراءة والكتابة من الملفات أيضًا اتصالاً بالنظام.
- إنشاء وإدارة عمليات جديدة.
- تتطلب اتصالات الشبكة أيضًا استدعاءات النظام. وهذا يشمل إرسال واستقبال الحزم.
- يتطلب الوصول إلى أجهزة مثل الطابعة والماسح الضوئي وما إلى ذلك إجراء اتصال بالنظام.

قبل أن نناقش كيفية إتاحة نظام التشغيل لاستدعاءات النظام، لنبدأ بمثال لتوضيح كيفية استخدامها: كتابة برنامج بسيط لقراءة البيانات من ملف ونسخها إلى ملف آخر كما في الشكل 3. أول مُدخل سيحتاجه البرنامج هو اسما الملفين: ملف الإدخال وملف الإخراج. يمكن تحديد هذين الاسمين بطرق عديدة، حسب تصميم نظام التشغيل. إحدى الطرق هي تمرير اسمي الملفين كجزء من الأمر – على سبيل المثال، أمر cp في نظام يونكس:

Unix: cp in.txt out.txt



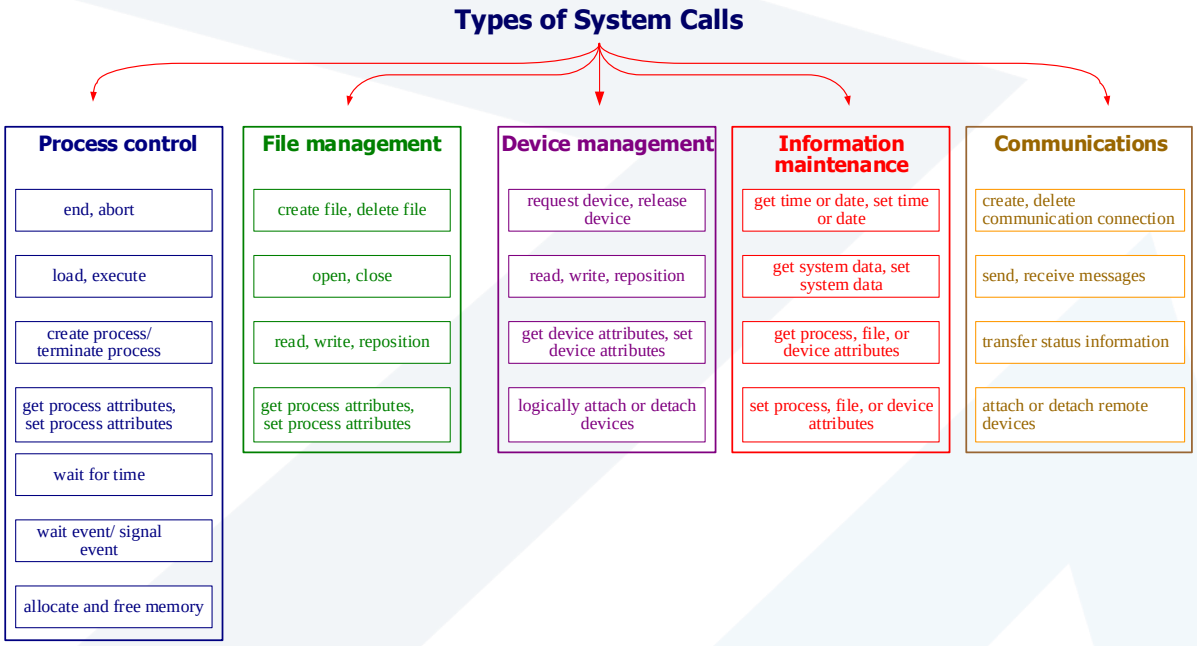
الشكل 3: مثال على استخدام استدعاءات النظام عند القراءة من ملف والنسخ إلى ملف آخر.

أنواع استدعاءات النظام

هناك خمسة أنواع أساسية من استدعاءات النظام كما في الشكل 4. هذه موضحة بالتفصيل على النحو التالي:

التحكم بالعملية

تتعامل استدعاءات النظام هذه مع عمليات مثل إنشاء العملية وإنهاء العملية وما إلى ذلك.



الشكل 4: أنواع استدعاءات النظام

إدارة الملفات

استدعاءات النظام هذه مسؤولة عن معالجة الملفات مثل إنشاء ملف أو قراءة ملف أو الكتابة في ملف وما إلى ذلك.

إدارة الجهاز

تعتبر استدعاءات النظام هذه مسؤولة عن معالجة الجهاز مثل القراءة من المخازن المؤقتة للجهاز والكتابة في المخازن المؤقتة للجهاز وما إلى ذلك.

صيانة المعلومات

تعالج استدعاءات النظام هذه المعلومات ونقلها بين نظام التشغيل وبرنامج المستخدم.

التواصل

استدعاءات النظام هذه مفيدة للاتصال بين العمليات. يتعاملون أيضًا مع إنشاء اتصال اتصال وحذفه.

بعض الأمثلة على جميع الأنواع المذكورة أعلاه من استدعاءات النظام في Windows و Unix مذكورة على النحو التالي:

Types of System Calls	Windows	Linux
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Management	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()

Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()

هناك العديد من استدعاءات النظام المختلفة كما هو موضح أعلاه. فيما يلي تفاصيل بعض هذه الاستدعاءات:

open()

يتم استخدام استدعاء النظام ***open()*** لتوفير الوصول إلى ملف في نظام الملفات. يخصص استدعاء النظام هذا الموارد للملف ويوفر مؤشرًا تستخدمه العملية للإشارة إلى الملف. يمكن فتح ملف من خلال عمليات متعددة في نفس الوقت أو تقييده بعملية واحدة. كل هذا يتوقف على تنظيم الملفات ونظام الملفات.

read()

يتم استخدام استدعاء النظام *read()* للوصول إلى البيانات من ملف مخزن في نظام الملفات. يمكن تحديد الملف المراد قراءته من خلال واصف الملف الخاص به ويجب فتحه باستخدام ***open()*** قبل التمكن من قراءته. بشكل عام، تأخذ استدعاءات نظام *read()* ثلاث وسائط، أي واصف الملف، المخزن المؤقت الذي يخزن بيانات القراءة وعدد البايتات المراد قراءتها من الملف.

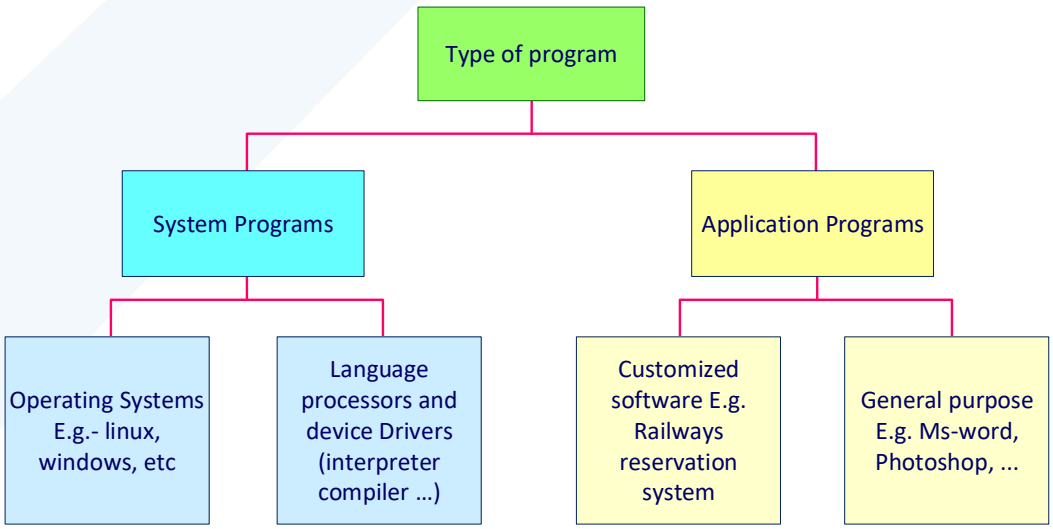
write()

تقوم استدعاءات نظام *write()* بكتابة البيانات من المخزن المؤقت للمستخدم إلى جهاز مثل ملف. يعد استدعاء النظام هذا أحد طرق إخراج البيانات من البرنامج. بشكل عام، تأخذ استدعاءات نظام الكتابة ثلاث وسيطات، أي واصف الملف، ومؤشر إلى المخزن المؤقت حيث يتم تخزين البيانات وعدد البايتات للكتابة من المخزن المؤقت.

close()

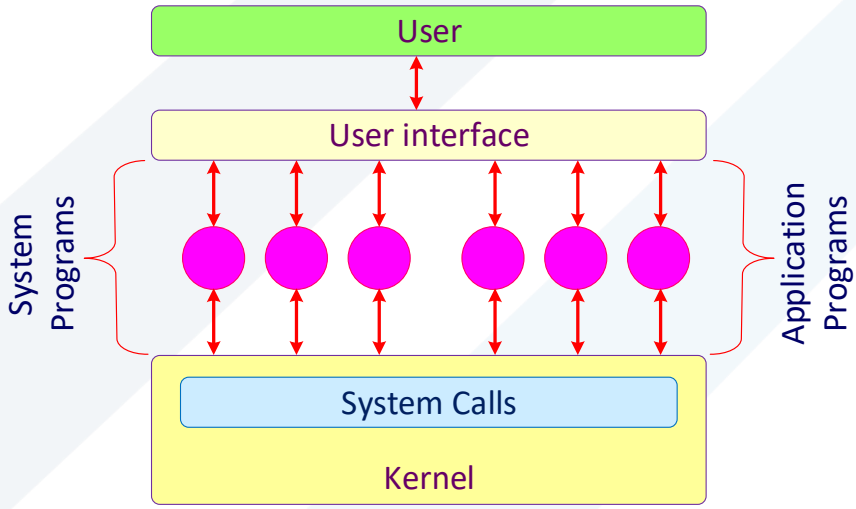
يتم استخدام استدعاء النظام close() لإنهاء الوصول إلى نظام الملفات. يعني استخدام استدعاء النظام هذا أن الملف لم يعد مطلوبًا من قبل البرنامج وبالتالي يتم مسح المخازن المؤقتة، ويتم تحديث البيانات الوصفية للملف وإلغاء تخصيص موارد الملف.

System program and Application program برامج النظام وبرامج التطبيقات



الشكل 5: أنواع البرمجيات

يشير مصطلح برمجيات النظام، الشكل 5، إلى فئة من برامج الحاسوب المصممة لإدارة وتسهيل تشغيل عتاد الحاسوب وتطبيقاته، بالإضافة إلى وظائف النظام بشكل عام. تعمل هذه البرمجيات كوسيط بين عتاد الحاسوب وتطبيقات المستخدم، حيث توفر خدمات أساسية مثل إدارة الذاكرة، والتحكم في الأجهزة الطرفية، ومعالجة المدخلات والمخرجات، وتسهيل الاتصال بين مكونات البرمجيات والأجهزة. من الأمثلة الرئيسية على برمجيات النظام أنظمة التشغيل (مثل Windows وmacOS وLinux)، وبرامج تشغيل الأجهزة، والبرامج الثابتة، والأدوات المساعدة، حيث تعمل جميعها معًا لضمان كفاءة وفعالية تشغيل نظام الحاسوب.



الشكل 6: برمجيات النظام وبرمجيات التطبيقات

أما برامج التطبيقات، الشكل 6، فتخصص لأداء مهام أو تطبيقات محددة للمستخدم. تُعرف أيضًا باسم "التطبيق" أو "البرنامج". وبخلاف برامج النظام، التي تُوفر الأساس والبنية التحتية لتشغيل الحاسوب وإدارة موارد الأجهزة، تُصمم برامج التطبيقات لتلبية احتياجات المستخدم ومتطلباته.

تستطيع هذه البرامج أداء وظائف متنوعة، بما في ذلك الإنتاجية والترفيه والتواصل والتعليم وغيرها. إذ توجد أنواع عديدة من برامج التطبيقات، بما في ذلك معالجات النصوص وجداول البيانات وبرامج البريد الإلكتروني ومتصفحات الويب وبرامج تحرير الفيديو وبرامج الألعاب وغيرها الكثير. وغني عن القول أن هذه البرامج تصمم عادةً لتكون سهلة الاستخدام، ويمكن تثبيتها وتشغيلها على أجهزة مختلفة، بما في ذلك أجهزة الكمبيوتر والهواتف الذكية والأجهزة اللوحية وأجهزة الألعاب.

مقارنة بين استدعاء النظام وبرنامج النظام في نظام التشغيل

فيما يلي بعض المقارنات بين System Call and System Program:

Features	System Call	System Program
Definition	إنها تقنية يطلب فيها برنامج نظام الكمبيوتر خدمة من نواة نظام التشغيل.	يوفر بيئة لتنصيب وتشغيل التطبيقات.
Request	يلبي الطلبات منخفضة المستوى لبرنامج المستخدم.	يفي بالطلب أو المتطلبات عالية المستوى لبرنامج المستخدم.

Programming Languages	عادة ما تكون مكتوبة بلغات البرمجة C و C ++. تُستخدم لغة على مستوى التجميع في استدعاءات النظام حيث يلزم الوصول المباشر إلى الأجهزة.	عادة ما تكون مكتوبة بلغات البرمجة عالية المستوى فقط.
User View	يحدد الواجهة بين الخدمات وعملية المستخدم التي يوفرها نظام التشغيل.	يحدد واجهة المستخدم (UI) لنظام التشغيل.
Action	تطلب عملية المستخدم خدمة نظام تشغيل باستخدام استدعاء نظام.	إنه يحول طلب المستخدم إلى مجموعة من استدعاءات النظام اللازمة للوفاء بالمتطلبات.
Classification	يمكن تصنيفها إلى معالجة الملفات، والتلاعب بالجهاز، والاتصال، والتحكم في العمليات، وصيانة المعلومات، والحماية.	يمكن تصنيفها إلى إدارة الملفات، وتحميل البرامج وتنفيذها، ودعم لغة البرمجة، ومعلومات الحالة، وتعديل الملف، والاتصال.

التعامل مع أجهزة متعددة Managing multiple devices

عندما يقوم أكثر من جهاز بإصدار إشارة طلب مقاطعة، فحينئذٍ تكون هناك حاجة إلى معلومات إضافية لتحديد الجهاز الذي يجب أخذه في الاعتبار أولاً. تُستخدم الطرق التالية لتحديد الجهاز المراد التعامل معه:

الاستقصاء، والمقاطعات الموجهة vectored interrupts، والمقاطعة المتداخلة.

الاستقصاء: هو أسلوب يُستخدم في تصميم الأنظمة للتحقق من حالة النظام أو جمع البيانات من مصادر متعددة بشكل دوري. يتضمن هذا الاستعلام أو التحقق المستمر من الأجهزة أو المكونات الأخرى على فترات زمنية محددة مسبقًا للتحقق من وجود أي معلومات جديدة أو استيفاء شروط معينة.

في سياق تصميم الأنظمة، يتضمن الاستطلاع عادةً قيام نظام مركزي (مثل الخادم) بإرسال طلبات بانتظام إلى أجهزة أو عُقد عميلة متعددة لجمع المعلومات أو تحديد حالتها.

يمكن أن يشمل ذلك التحقق من التحديثات، أو استرجاع البيانات، أو التحقق من اكتمال مهام معينة.

في الاستقصاء، أول جهاز تواجهه بت IRQ هو الجهاز الذي سيتم صيانته أولاً. يتم استدعاء ISR المناسب لخدمة نفس الشيء. إنه سهل التنفيذ ولكن يضيع الكثير من الوقت عن طريق استقصاء بت IRQ لجميع الأجهزة.

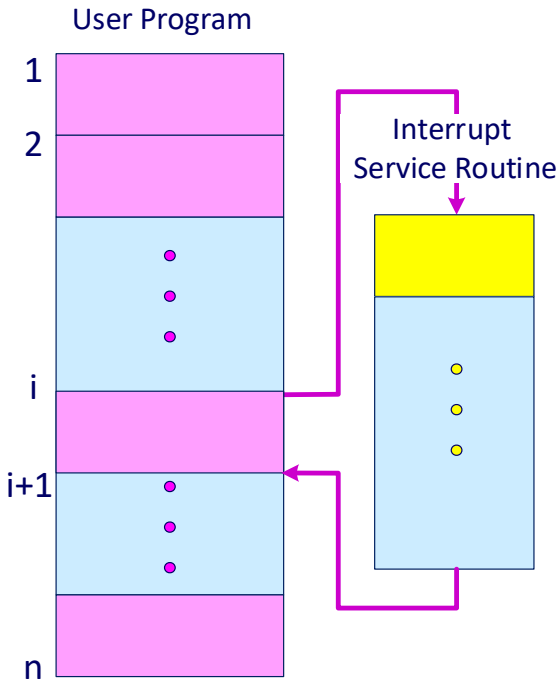
المقاطعات: في المقاطعات ، يقوم الجهاز الذي يطلب المقاطعة بتعريف نفسه مباشرة عن طريق إرسال رمز خاص إلى المعالج عبر الناقل. يمكّن هذا المعالج من تحديد الجهاز الذي تسبب في حدوث المقاطعة. يمكن أن يكون الرمز الخاص هو عنوان البداية لـ ISR أو حيث يوجد ISR في الذاكرة ويسمى ناقل المقاطعة.

المقاطعات المتداخلة: في هذه الطريقة، يتم تنظيم جهاز الإدخال/الإخراج في بنية الأولوية. لذلك، يتم التعرف على طلب مقاطعة من جهاز ذي أولوية أعلى بينما لا يتم التعرف على طلب من جهاز ذي أولوية أقل. يقبل المعالج المقاطعات فقط من الأجهزة/العمليات ذات الأولوية الأعلى.

المقاطعات في بنية الحاسوب

ما هي المقاطعة؟

المقاطعة هي إشارة تُصدرها الأجهزة أو البرامج عندما تحتاج عملية أو حدث إلى معالجة فورية. تُنبه هذه الإشارة المعالج إلى وجود عملية ذات أولوية عالية تتطلب مقاطعة العملية الحالية. في أجهزة الإدخال/الإخراج، يُخصص أحد خطوط التحكم في الناقل لهذا الغرض، ويُسمى روتين خدمة المقاطعة Interrupt Service Routine (ISR)، الشكل 7.



الشكل 7: كيفية تنفيذ ISR

عندما يُطلق جهاز ما مقاطعة، فإن المعالج يُكمل أولًا تنفيذ التعليمة التي يقوم بتنفيذها، ثم يُحمّل عداد البرامج program counter (PC) بعنوان أول تعليمة من روتين خدمة المقاطعة. قبل تحميل عداد البرامج بالعنوان، يُنقل عنوان التعليمة التي تمت مقاطعتها إلى موقع مؤقت. بحيث يُمكن للمعالج متابعة العملية التي تمت مقاطعتها بعد معالجة المقاطعة.

أثناء معالجة المعالج للمقاطعات، يجب عليه إبلاغ الجهاز بأنه قد تم التعرف على طلبه ليتوقف عن إرسال إشارة طلب المقاطعة. كما يتم حفظ السجلات لاستعادة العملية التي تمت مقاطعتها في المستقبل، وهذا يزيد من التأخير بين وقت استلام المقاطعة وبدء تنفيذ ISR، وهذا ما يُسمى بـ "زمن استجابة المقاطعة" Interrupt Latency.

أنواع المقاطعة

يمكن للبرامج أو الأجهزة المرتبطة بالحدث أن تُطلق إشارات المقاطعة، أي تنقسم المقاطعة إلى فئتين: مقاطعات البرامج أو مقاطعات الأجهزة.

مقاطعات البرامج Software Interrupt

وهو المقاطعة التي ينتجها البرنامج أو النظام، وليس الجهاز. وتسمى "الفخ" Traps و"الاستثناء" exception. وهي بمثابة إشارة لنظام التشغيل أو خدمة النظام لتنفيذ وظيفة معينة أو الاستجابة لحالة خطأ. عادةً، تحدث مقاطعات البرامج نتيجة استخدام تعليمات محددة أو استثناءات في العملية.

تُستخدم تعليمة مُحددة تُعرف باسم "تعليمة المقاطعة" interrupt instruction لإنشاء مقاطعات برمجية. عند استخدام تعليمة المقاطعة، يُوقف المعالج ما يفعله وينتقل إلى كود مُحدد لمعالج المقاطعة. يُكمل روتين مُعالج المقاطعة ISR العمل المطلوب أو يُعالج أي أخطاء قبل إعادة التحكم إلى التطبيق المُقاطع.

مقاطعات الأجهزة Hardware Interrupt

في مقاطعة الأجهزة، تتصل جميع الأجهزة بخط طلب المقاطعة. يُستخدم خط طلب واحد لجميع الأجهزة لطلب مقاطعة، يُغلق الجهاز مفتاحه المرتبط. تنقسم مقاطعات الأجهزة إلى نوعين:

مقاطعة قابلة للإخفاء Maskable Interrupt MI: يمكن تفعيل وتعطيل مقاطعات الأجهزة بشكل انتقائي بفضل سجل قناع المقاطعة المدمج، والذي يُستخدم عادةً في المعالجات. يُقابل بت في سجل القناع إشارة كل مقاطعة؛ في بعض الأنظمة، تُفعّل المقاطعة عند ضبط البت وتُعطّل عند مسح البت، ولكن في أنظمة أخرى، تُلغى المقاطعة عند ضبط البت.

المقاطعة غير القابلة للإخفاء Non-Maskable Interrupt NMI: هي مقاطعة أجهزة لا يمكن لتقنيات إخفاء المقاطعات القياسية في النظام تجاهلها. تحدث عادةً للإشارة إلى أخطاء الأجهزة غير القابلة للإصلاح.

مقاطعة خفية Spurious Interrupt: تُعرف مقاطعة الأجهزة التي لا يتم تحديد مصدرها باسم مقاطعة خفية. قد تُسمى هذه الظاهرة أيضًا بالمقاطعات الوهمية أو الخفية. عند توصيل دارة مقاطعة OR سلكية بمدخل معالج حساس للمستوى، فإن مثل هذه المقاطعات تكون قابلة للحدوث. وعندما يفشل النظام في الأداء، قد يكون من الصعب تحديد مصدر هذه المقاطعات.

ويكون تسلسل الأحداث المتضمنة في معالجة طلب مقاطعة Interrupt Request IRQ كما يلي:

1. تُطلق الأجهزة طلب مقاطعة IRQ
2. يقاطع المعالج البرنامج الجاري تنفيذه.
3. يُعلم الجهاز بأنه قد تم التعرف على طلبه، فيقوم بإلغاء تنشيط إشارة الطلب.
4. يتم تنفيذ الإجراء المطلوب.
5. يتم تفعيل المقاطعة، ويُستأنف البرنامج الذي تمت مقاطعته.

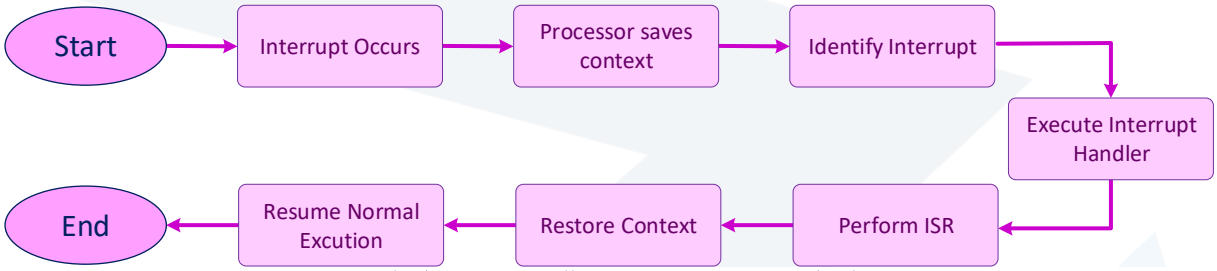
ويمكن إيضاح خطوات الاستجابة للمقاطعة من قبل النظام كما في الشكل 8:

الخطوة ١: تحديد نوع المقاطعة، فعند حدوث أي مقاطعة، قد تكون إما مقاطعة إدخال/إخراج أو مقاطعة نظام.

الخطوة ٢: بعد ذلك، تُخزَّن الحالة الحالية، بما في ذلك السجلات وعداد البرنامج، للحفاظ على حالة العملية.

الخطوة ٣: يتم تحديد المقاطعة الحالية ومعالجها من خلال جدول متجه المقاطعة في المعالج.

الخطوة ٤: ينتقل هذا التحكم الآن إلى معالج المقاطعة، وهو دالة موجودة في مساحة النواة.



الشكل 8: خطوات الاستجابة للمقاطعة من قبل النظام

الخطوة ٥: يتم تنفيذ مهام محددة بواسطة روتين خدمة المقاطعة ISR، وهي ضرورية لإدارة المقاطعة.

الخطوة ٦: يتم استرجاع الحالة من الجلسة السابقة للبناء على العملية من تلك النقطة.

الخطوة ٧: بعد ذلك، يُعاد التحكم إلى العملية الأخرى التي كانت معلقة، وتستمر العملية العادية.

بنى نظام التشغيل

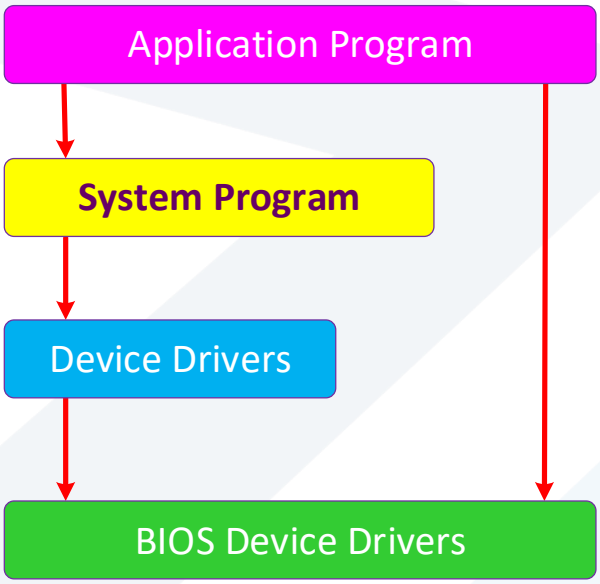
نظام التشغيل هو بنية تسمح لبرامج تطبيقات المستخدم بالتفاعل مع مكونات النظام المادية. ونظرًا لتعقيد بنية نظام التشغيل، يجب تصميمه بعناية فائقة ليسهل استخدامه وتعديله. ومن الطرق السهلة لذلك إنشاء نظام التشغيل على أجزاء. يجب أن يكون كل جزء محددًا بدقة مع مدخلات ومخرجات ووظائف واضحة. ووفقا لكيفية تواصل هذه الأجزاء مع بعضها البعض تم الاعتماد عل تصميم نظام التشغيل وفقا لبنى محددة منها:

- البنية البسيط Simple Structure
- البنية المتجانسة Monolith Structure
- بنية النواة الدقيقة Micro-Kernel Structure
- بنية النواة الخارجية Exo-Kernel Structure
- البنية الطبقية Layered Structure
- البنية المعيارية Modular Structure
- الآلات الافتراضية Virtual Machines
- البنى الضبابية Cloud Structure

البنية البسيطة

وهي بنية نظام التشغيل الأكثر وضوحًا، وهي مناسبة فقط للاستخدام مع الأنظمة الصغيرة والمقيدة. نظرًا لأن الواجهات ودرجات الوظائف في هذه البنية محددة بوضوح، فإن البرامج قادرة على الوصول إلى إجراءات الإدخال/الإخراج، والتي قد تؤدي إلى الوصول غير المصرح به إلى إجراءات الإدخال/الإخراج. مثال على ذلك MS-DOS.

يستخدم نظام التشغيل MS-DOS البنية البسيطة المبينة في الشكل 9 :



الشكل 9: البنية البسيطة

- هناك أربع طبقات تشكل نظام التشغيل MS-DOS، الشكل 9، ولكل منها مجموعة ميزاتها الخاصة.
- تتضمن هذه الطبقات برامج تشغيل أجهزة ROM BIOS وبرامج تشغيل أجهزة MS-DOS وبرامج التطبيقات وبرامج النظام.
- يستفيد نظام التشغيل MS-DOS من الطبقات لأنه يمكن تعريف كل مستوى بشكل مستقل، وعند الضرورة، يمكن أن تتفاعل مع بعضها البعض.
- إذا كان النظام مبنيًا على طبقات، فسيكون من الأسهل تصميمه وإدارته وتحديثه. لهذا السبب، يمكن استخدام الهياكل البسيطة لبناء أنظمة مقيدة أقل تعقيدًا.
- عندما يفشل برنامج المستخدم، يتعطل نظام التشغيل ككل.
- نظرًا لأن أنظمة MS-DOS تتمتع بمستوى منخفض من التجريد، فإن البرامج وإجراءات الإدخال/الإخراج مرئية للمستخدمين، مما يمنحهم إمكانية الوصول غير المرغوب فيه.

تتمتع البنية البسيطة ببعض المزايا ومنها:

- أنه ونظرا لوجود عدد قليل من الواجهات والمستويات، فمن السهل تطويرها.
- كما أنه ونظرًا لوجود طبقات أقل بين الأجهزة والتطبيقات، فإنه يوفر أداءً فائقًا.

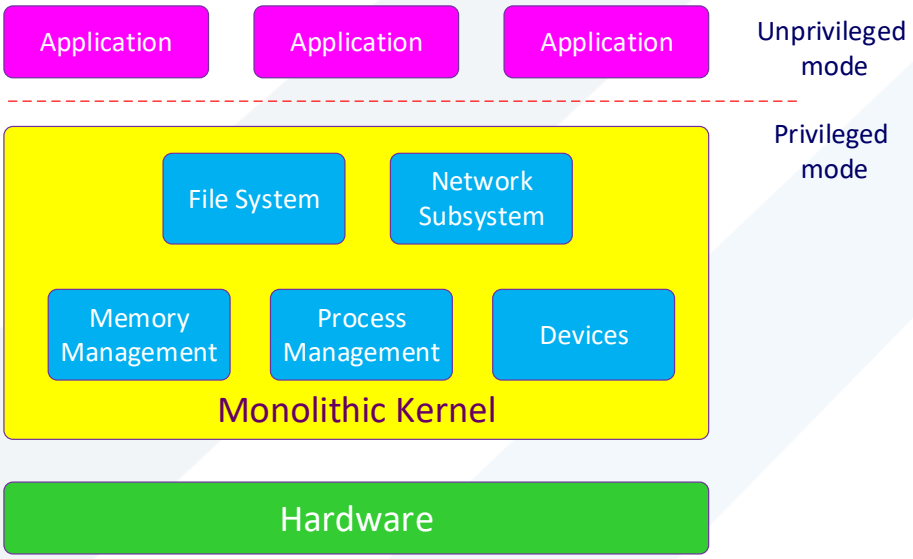
ولكنها تعاني أيضا من سلبيات عديدة ومنها:

- يتعطل نظام التشغيل بالكامل في حالة تعطل برنامج مستخدم واحد فقط.
- بما أن الطبقات مترابطة وتتواصل مع بعضها البعض، فلا يوجد تجريد أو إخفاء للبيانات.
- يمكن الوصول إلى عمليات نظام التشغيل من قبل الطبقات الأعلى، مما قد يؤدي إلى العبث بالبيانات وفشل النظام.

البنية المتجانسة monolithic

يتحكم نظام التشغيل المتجانس *monolithic* في جميع جوانب تشغيل نظام التشغيل، بما في ذلك إدارة الملفات وإدارة الذاكرة وإدارة الجهاز والعمليات التشغيلية، كما في الشكل 10.

يُطلق على جوهر نظام تشغيل أجهزة الكمبيوتر اسم النواة. إذ يتم تزويد جميع مكونات النظام الأخرى بالخدمات الأساسية بواسطة kernel، حيث يكون نظام التشغيل مدمجًا في قطعة واحدة من الأجهزة.



الشكل 10: البنية المتجانسة

غالبًا ما يشار إلى نظام التشغيل هذا باسم النواة المتجانسة. تعمل تقنيات البرمجة المتعددة مثل معالجة الدُفعات ومشاركة الوقت على زيادة قابلية استخدام المعالج. تعمل النواة المتجانسة فوق نظام التشغيل وتحت القيادة الكاملة لجميع الأجهزة، وهي تؤدي دور الكمبيوتر الافتراضي. وهذه بنية نظام تشغيل قديم تم استخدامه في البنوك لتنفيذ مهام بسيطة مثل معالجة الدُفعات ومشاركة الوقت، مما يسمح للعديد من المستخدمين في محطات طرفية مختلفة بالوصول إلى نظام التشغيل.

تتمتع هذه النية بمزايا منها أنه:

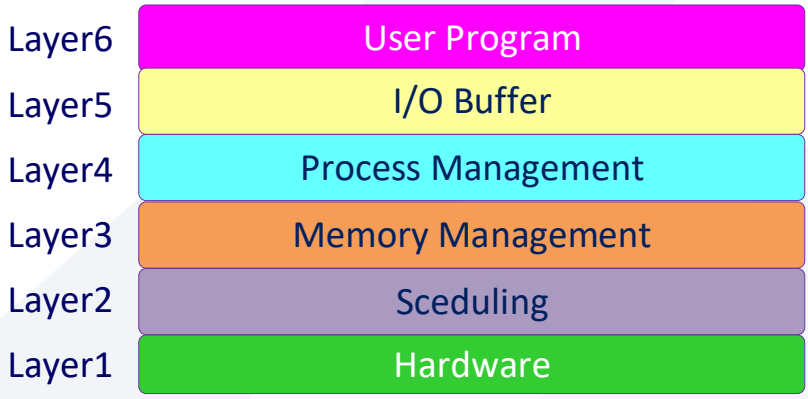
- نظرًا لأن الطبقات غير ضرورية والنواة وحدها هي المسؤولة عن إدارة جميع العمليات، فمن السهل تصميمها وتنفيذها.
- ونظرًا لحقيقة أن وظائف مثل إدارة الذاكرة، وإدارة الملفات، وجدولة العمليات، وما إلى ذلك، يتم تنفيذها في نفس منطقة العنوان، فإن النواة تعمل بسرعة إلى حد ما عند مقارنتها بالأنظمة الأخرى. يؤدي استخدام نفس العنوان إلى تسريع وتقليل الوقت المطلوب لتخصيص العنوان للعمليات الجديدة.

كما أنها تعاني من مساوئ منها:

- أن خدمات النواة المتجانسة مترابطة في مساحة العنوان ولها تأثير على بعضها البعض، لذلك إذا تعطل أي منها، فإن النظام بأكمله يتعطل كذلك.
- كما أن هذه البنية غير قابلة للتكيف. لذلك، فإن إطلاق خدمة جديدة أمر صعب.

البنية الطبقية

يتم فصل نظام التشغيل في هذه النية إلى طبقات أو مستويات. تحتوي الطبقة 0 (الطبقة الدنيا) على الأجهزة، وتحتوي أعلى طبقة على واجهة المستخدم الطبقة N (في مثالنا في الشكل11 n=6). ويتم تنظيم هذه الطبقات بشكل هرمي، حيث تستفيد طبقات المستوى الأعلى من إمكانات الطبقات ذات المستوى الأدنى.



الشكل 11: البنية الطبقية

يتم فصل وظائف كل طبقة في هذه الطريقة، ونظرًا لأن البنى ذات الطبقات هرمية، فإن تصحيح الأخطاء يكون أبسط، وبالتالي يتم تصحيح أخطاء جميع طبقات المستوى الأدنى قبل فحص الطبقة العليا. نتيجة لذلك، يجب مراجعة الطبقة الحالية وحدها وضمان أنه تم بالفعل فحص جميع الطبقات السفلية.

ولهذه البنية مجموعة من الميزات منها:

- واجبات العمل منفصلة لأن كل طبقة لها وظائفها الخاصة، وهناك قدر من التجريد.
- التصحيح أبسط لأنه يتم فحص الطبقات السفلية أولاً، متبوعة بالطبقات العليا.

كما أنها تعاني من عدة عيوب مثل:

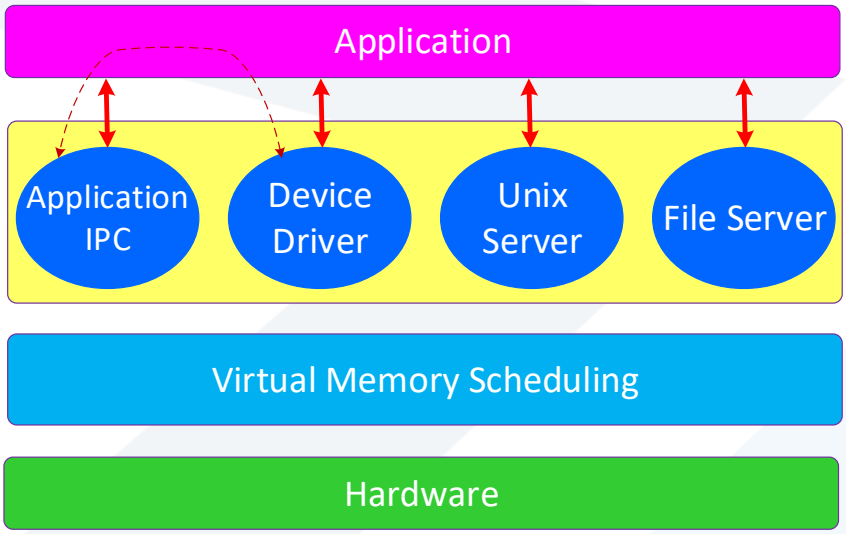
- تأثر الأداء بسبب الطبقات.
- أن بناء الطبقات يتطلب تصميمًا دقيقًا لأن الطبقات العليا تستفيد فقط من قدرات الطبقات السفلية.

البنية ذات النواة الصغيرة

يتم إنشاء نظام التشغيل باستخدام micro-kernel حيث يتم تخليص النواة من أي أجزاء غير ضرورية. تُستخدم الأنظمة وتطبيقات المستخدم لتنفيذ مكونات النواة الاختيارية هذه..

يتم إنشاء كل Micro-Kernel بشكل منفصل ويتم الاحتفاظ به بعيدًا عن الآخرين. نتيجة لذلك، يصبح النظام أكثر موثوقية وأمانًا. في حالة حدوث عطل فيMicro-Kernel ، فلن يتأثر نظام التشغيل المتبقي ويستمر في العمل بشكل طبيعي.

ويوضح الشكل 12 بنية نظام تشغيل هذا.



الشكل 12: بنية النواة الصغيرة Micro-kernel

كما تتمتع هذه البنية بمجموعة من المزايا مثل:

- يتيح إمكانية نقل نظام التشغيل عبر الأنظمة الأساسية.
- نظرًا لعزل كل Micro-Kernel، فهي موثوقة وآمنة.
- يسمح الحجم الصغير لـ Micro-Kernels بإجراء اختبار ناجح.
- يظل نظام التشغيل المتبقي غير متأثر ويستمر في العمل بشكل صحيح حتى في حالة فشل أحد المكونات أو Micro-Kernel.

ولكنها تعاني من المساوئ التالية:

- ينخفض أداء النظام من خلال زيادة الاتصال بين الوحدات.
- بناء نظام معقد.

البنى الافتراضية (VMs)

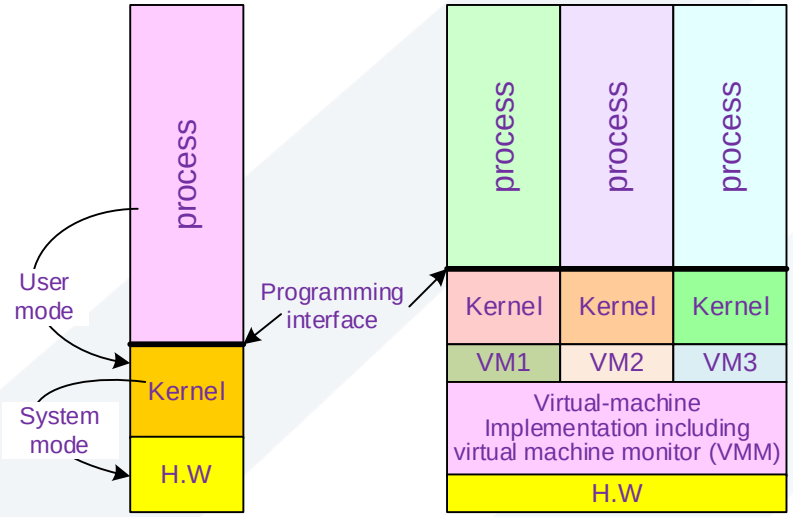
يتم تجريد أجهزة الكمبيوتر الشخصي لدينا، بما في ذلك وحدة المعالجة المركزية ومحركات الأقراص وذاكرة الوصول العشوائي و NIC (بطاقة واجهة الشبكة)، بواسطة بنية افتراضية (برمجية) في مجموعة متنوعة من مسارات التنفيذ المختلفة بناءً على احتياجاتنا، مما يمنحنا انطباعًا بأن كل بيئة تنفيذ هو جهاز كمبيوتر منفصل كما في الشكل 13.

باستخدام جدولة وحدة المعالجة المركزية وتقنيات الذاكرة الافتراضية، يتيح لنا نظام التشغيل تنفيذ عمليات متعددة في وقت واحد مع إعطاء الانطباع بأن كل واحدة تستخدم معالجًا وذاكرة افتراضية منفصلة. تعد استدعاءات النظام ونظام الملفات أمثلة على الوظائف الإضافية التي يمكن أن تمتلكها العملية ولا يستطيع الجهاز تقديمها. بدلاً من تقديم هذه الميزات الإضافية، تقدم طريقة البنية الافتراضية فقط واجهة مشابهة لتلك الموجودة في معظم الأجهزة الأساسية. يتم توفير نسخة افتراضية من نظام الكمبيوتر تحتها لكل عملية.

يمكننا تطوير جهاز افتراضي لمجموعة متنوعة من الأسباب، وكلها مرتبطة بشكل أساسي بالقدرة على مشاركة نفس الأجهزة الأساسية مع دعم بيئات التنفيذ المختلفة في نفس الوقت، أي أنظمة التشغيل المختلفة.

ومن مزايا هذه البنية:

- نظرًا للعزل التام بين كل جهاز افتراضي وكل جهاز افتراضي آخر، لا توجد مشكلات تتعلق بالأمان.
- قد يوفر الجهاز الظاهري بنية لمجموعة التعليمات تختلف عن تلك الموجودة في أجهزة الكمبيوتر الفعلية.
- سهولة التوافر وإمكانية الوصول وسهولة الاسترداد.



System models, (a) Nonvirtual machine, (b) Virtual machine.

الشكل 13: البنى الافتراضية مقارنة مع البنى التقليدية

أما عيوبها فتضم:

- اعتمادًا على حجم العمل، فإن تشغيل العديد من الأجهزة الافتراضية في وقت واحد على جهاز كمبيوتر مضيف قد يكون له تأثير سلبي على أحدها.
- عندما يتعلق الأمر بالوصول إلى الأجهزة، تكون أجهزة الكمبيوتر الافتراضية أقل فعالية من الأجهزة المادية. مزايا