

الباب الاول المفاهيم العامة (نبذة)

يهدف هذا الباب الى الاتي:

مكونات الحاسوب بصفة عامة.

الحاسوب بدون نظام تشغيل.

بداية نظم التشغيل.

تعريف نظم التشغيل واهدافها.

الموارد.

واجهات نظام التشغيل.

ما يمثل نظام التشغيل.

مكونات نظام الحاسب

واجهات نظام التشغيل.

وصف العناصر الاساسية في بنية الحاسوب والعلاقة بينها.

خطوات تنفيذ التعليمات.

فهم المقاطعة وكيفيةها و لماذا تستخدم.

الترتيب الهرمي للذواكر.

مفهوم المكدس (Stack) عند استدعاء اجراء معين.

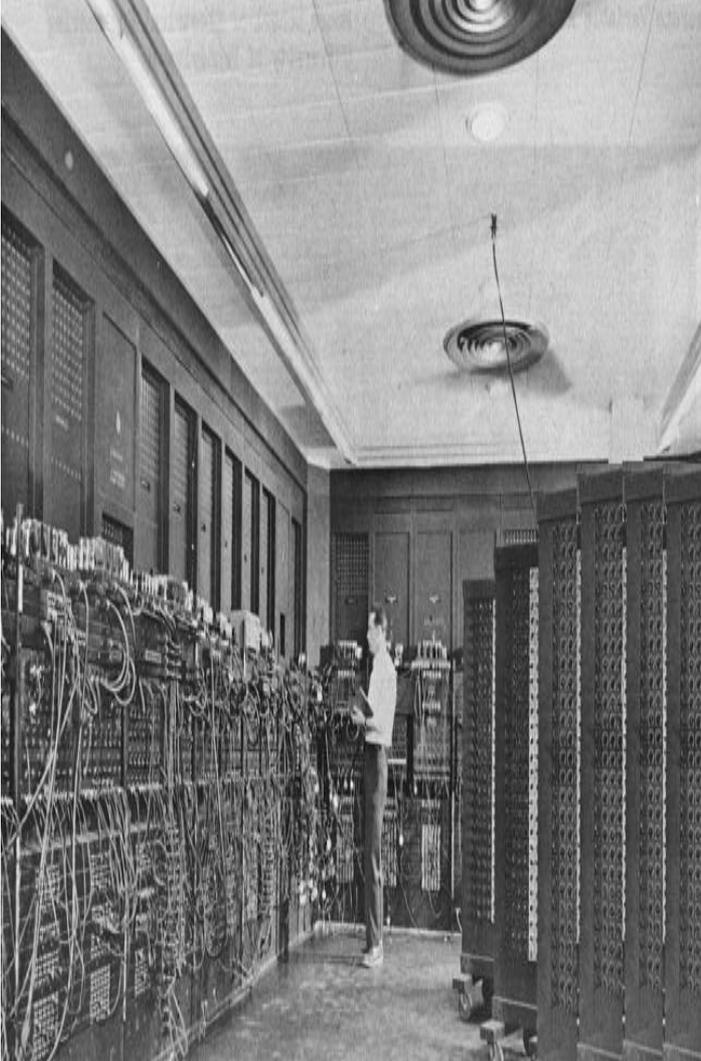
الباب الاول المفاهيم العامة (نبذة)

مدخل

الحاسب جهاز يتكون من مكونات مادية (Hardware) ومكونات برمجية (Software)

- المكونات المادية تمثل جسد الحاسب، أي الأجهزة الملموسة من شاشة، ولوحة المفاتيح ، ومعالج، وذاكرة، وغيرها .
- المكونات البرمجية فتمثل روح الحاسب (هي التي تتحكم في المكونات المادية وتوجه عملها).

الباب الاول المفاهيم العامة (نبذة)



1.1 الحاسبات قبل نظم تشغيل

بدأت الحاسبات قديما بدون برامج وبدون نظم تشغيل، وكان العمل كله يتم بلغة الآلة (شفرة مكونة من أصفار و وحائد)، وبالتالي لا يتعامل مع الحاسب إلا المهندسين المختصين. حتى المبرمج العادي كان يسلم برامجه للمختص ليدخلها له في الحاسب وينفذها له، ثم يسلمه النتائج.

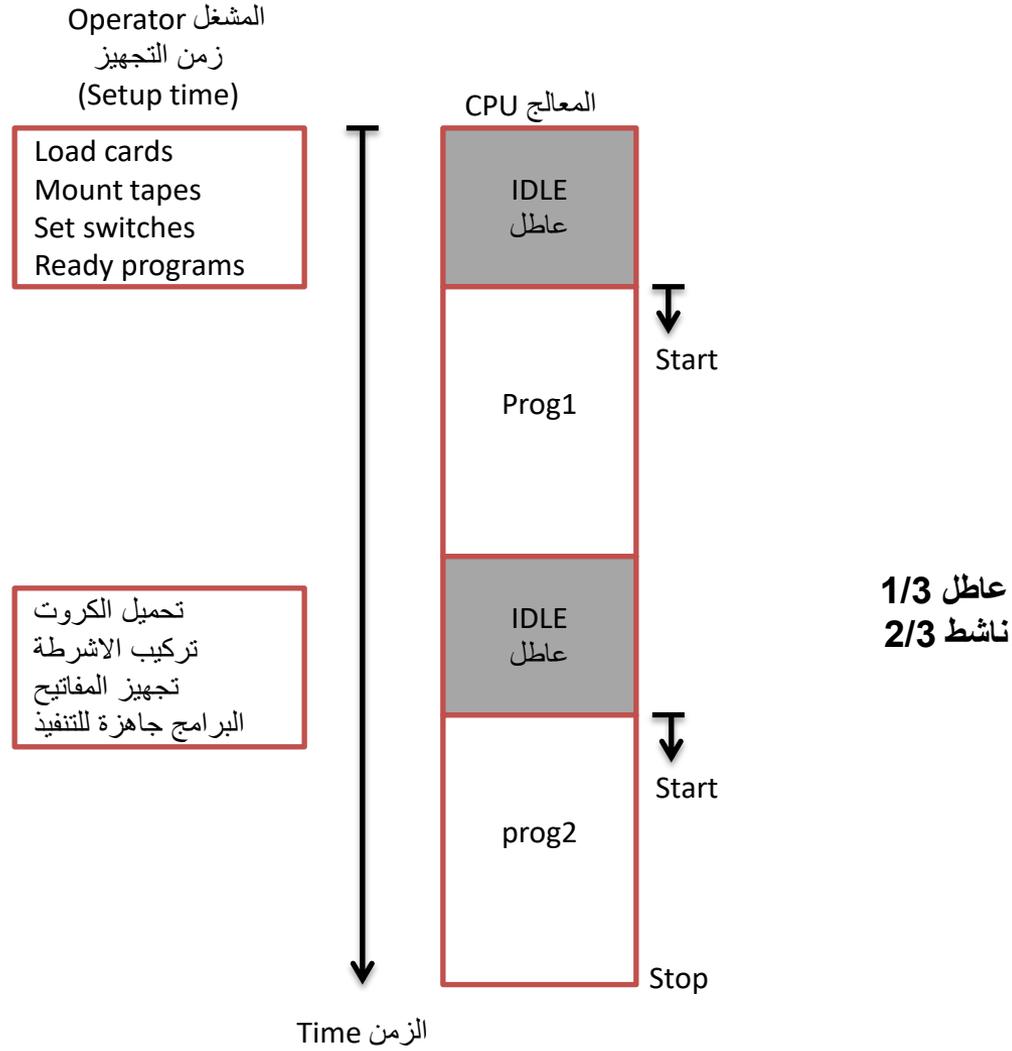
كانت الأجهزة في ذلك الوقت:

- ضخمة،
- بطيئة ومحدودة الأغراض.
- باهظة السعر ولا يستطيع امتلاكها أشخاص عاديين.

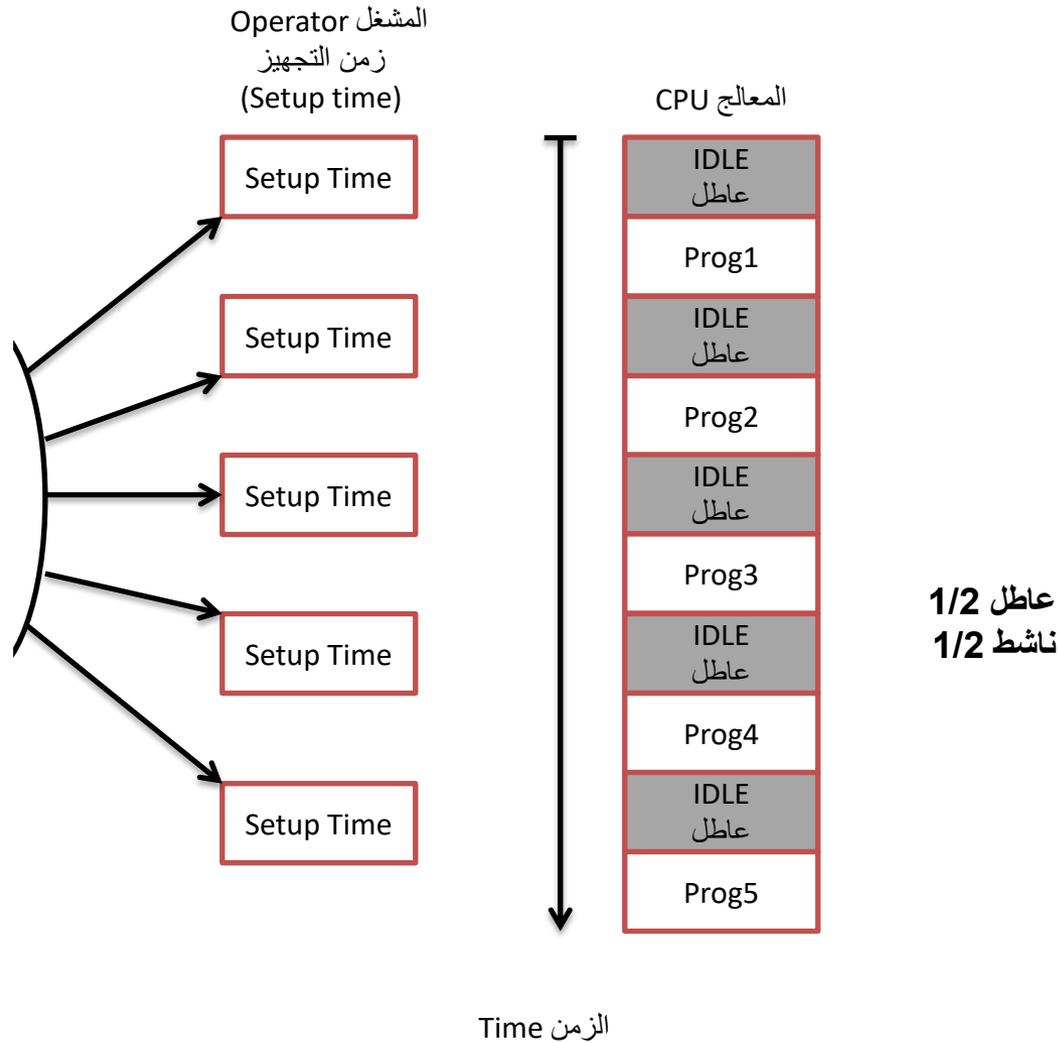
وكانت البرامج:

- تكتب بلغة الآلة.
 - تدخل بالبطاقة المثقبة.
 - تقوم بعمليات حسابية ورياضية محدودة.
 - تظهر مخرجاتها على الورق عبر الطابعة.
- لذلك كان التعامل مع الحاسب أمرا عسيراً و شاقا لا يستطيعه إلا المختصين المهرة.

الباب الاول المفاهيم العامة (نبذة)



الباب الاول المفاهيم العامة (نبذة)



الباب الاول المفاهيم العامة (نبذة)

1.2 ظهور نظم التشغيل

بدأ العمل بتصميم برامج تؤدي جزء من مهام المشغل وبدأت أعباءه تقل تدريجياً، إلى أن تم إحلال كامل للمشغل ببرامج تقوم بكل مهامه السابقة، فكانت نظم التشغيل التي وفرت الكثير من الوقت المستغرق للتعامل مع الحاسب ليستفاد منه في تطوير نظام التشغيل والتطبيقات الأخرى.

ثم تطورت نظم التشغيل من نظم تكتب أوامرها في شكل نصوص مثل **DOS** إلى نظم تشغيل رسومية في شكل أيقونات ورموز يستطيع كل شخص التعامل معها، وبالتالي أصبح الكل يتعامل مع الحاسب بكل سهولة ويسر.

الباب الاول المفاهيم العامة (نبذة)

1.3 تعريف نظام التشغيل

- هو البرنامج الذي يتحكم في تنفيذ برامج المستخدم ويعمل كوسيط بين المستخدم والمكونات المادية.
- والتعريف الاعم لنظام التشغيل هو ذلك البرنامج الذي نراه عندما نفتح الحاسب ولا يفارقنا إلا عند إغلاقه. وهو أول برنامج يثبت على الحاسب ليدير جميع موارده ويتيح للمستخدم واجهة مستخدم (user interface) تمكنه من التعامل مع المكونات المادية بكل سهولة ويسر.
- هو البرنامج الذي يهتم بإدارة الموارد و الخدمات (Resources and Services) مثل الذاكرة، المعالجات (processors)، و الاجهزة والمعلومات. وبناء على ذلك فهو برنامج لإدارة الموارد المادية او البرمجية عن طريق الجدولة وادارة الذاكرة وبرامج الدخل والخرج وكذلك نظام الملفات.

1.4 أهداف نظام التشغيل الرئيسية هي

- تنفيذ تطبيقات المستخدم.
- توفير بيئة مناسبة وملائمة للاستخدام (convenient)
- الاستفادة القصوى من الموارد وذلك بجعلها تعمل بشكل فعال (efficient).

الباب الاول المفاهيم العامة (نبذة)

1.5 الموارد (resources)

موارد الحاسب تشمل المكونات المادية من لوحة المفاتيح وشاشة وطابعة، وغيرها، وكذلك الملفات والبرامج وصفحات الويب وما شابه. ويقوم نظام التشغيل بإدارة الموارد. و المقصود بإدارة الموارد هو:

- **حجز المورد (allocate)** للبرنامج الذي يطلبه، ثم تحريره (free) بعد الانتهاء منه وإتاحته لتستفيد منه برامج أخرى.
- **يستخدم المورد بكفاءة والاستفادة منه الاستفادة القصوى**: مثلا إذا كان المعالج ينفذ في برنامج معين، وأحتاج هذا البرنامج إلى معلومة من لوحة المفاتيح (قد يستغرق وصول المعلومة وقتا ليس بالقصير مقارنة مع سرعة المعالج)، في هذه الحالة سيقوم نظام التشغيل بالاستفادة من المعالج في تنفيذ برنامج آخر ريثما تصل المعلومة من لوحة المفاتيح، هنا يكون نظام التشغيل قد استفاد من زمن المعالج في هذه الفترة.
- **العدل في استخدام الموارد**: يمنع نظام التشغيل البرامج من حجز الموارد واستخدامها لمدة طويلة (بما يعرف الاستئثار او الاحتكار Monopolizing).

الباب الاول المفاهيم العامة (نبذة)

1.6 تنفيذ برامج المستخدم

يقوم نظام التشغيل بتحميل برامج المستخدم في الذاكرة وتشغيلها، وتوفر معظم نظم التشغيل الحديثة تحميل وتشغيل اكثر من برنامج في وقت واحد (تعدد البرامج).

1.7 نظام التشغيل كبرنامج تحكمي

يتحكم نظام التشغيل في تشغيل البرامج الأخرى ويدير عملية تنفيذها لتفادي الأخطاء وتجنب الاستخدام الغير مرشد لموارد الحاسب وحماية البرامج من بعضها البعض ومن نظام التشغيل.

الباب الاول المفاهيم العامة (نبذة)

1.8 ما يمثل نظام تشغيل ؟

إذا قمت بتثبيت نظام التشغيل ويندوز ستجد معه الكثير من البرامج مثل متصفح الإنترنت، ومشغل الوسائط (media player) ، والرسام والمفكرة والكثير من الألعاب وغيرها من البرامج، هل نعتبر هذه البرامج جزء من نظام التشغيل؟ للمستخدم العادي نقول نعم !.

أما علميا فنقول:

أن نظام التشغيل هو ذلك الجزء الذي يتعامل ويدير المكونات المادية مباشرة، وهو النواة (kernel) التي لا نراها و لا نتعامل معها مباشرة، لكننا لا نستغني عن خدماتها التي هي سبب تشغيل بقية البرامج والواجهات التي نتعامل معها.

البعض يضيف إلى نظام التشغيل الغلاف والواجهة الرسومية التي من خلالها نستخدم النظام.

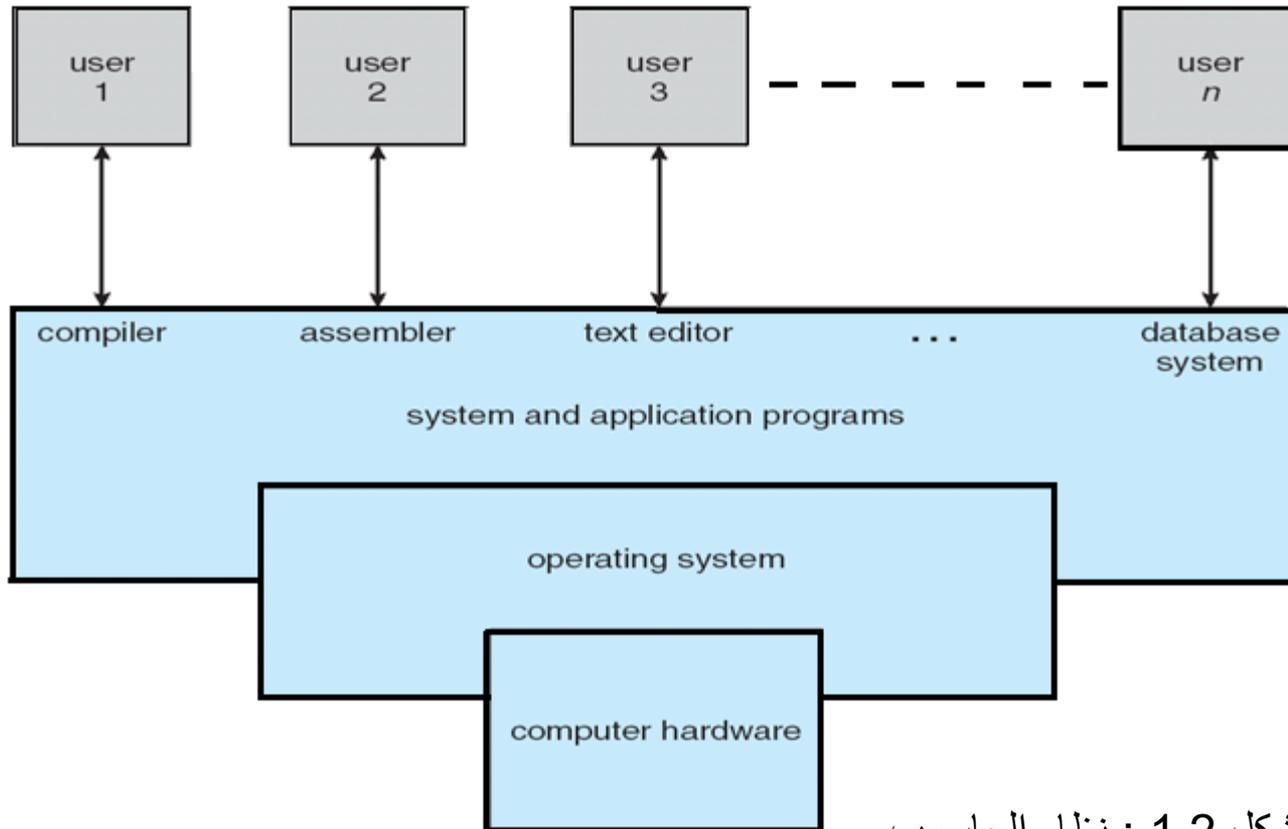
الباب الاول المفاهيم العامة (نبذة)

1.9 مكونات نظام الحاسب

نظام الحاسب و هو عبارة عن مكونات مادية و مكونات برمجية، يمكن تفصيل هذه المكونات بصورة أدق إلى الآتي:

1. مكونات الحاسب المادية (computer hardware).
2. النواة (kernel).
3. واجهات نظام التشغيل (operating system interfaces).
4. تطبيقات المستخدم (applications).
5. المستخدم (user).

مكونات نظام الحاسب

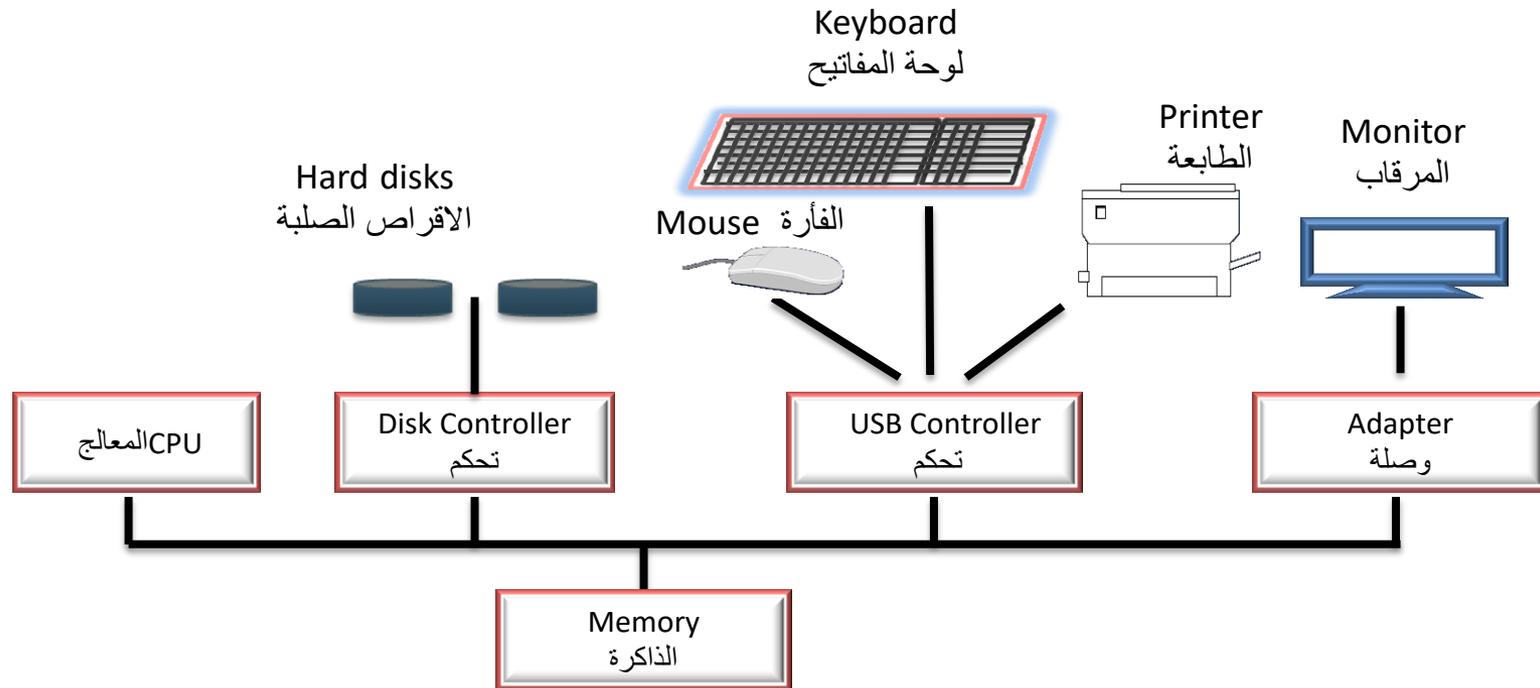


الشكل 1.2 : نظام الحاسوب

مكونات نظام الحاسب

1. المكونات المادية

يتكون الحاسب من معالج أو أكثر، ذاكرة رئيسية، أجهزة تخزين مثل القرص الصلب، أجهزة دخل وخرج، نواقل لتوصيل هذه الأجهزة مع بعضها.



الشكل 1.2 : مكونات الحاسوب الحديثة

مكونات نظام الحاسب

2. النواة – الكرنل (Kernel) : تدير النواة مكونات الحاسب المادية. وتنقسم إلى خمسة أجزاء رئيسية هي:

- جزء مسئول عن إدارة المعالج يسمى مدير العملية (Process Manager).
- جزء مسئول عن الذاكرة الرئيسية يسمى مدير الذاكرة (Memory Manager).
- جزء مسئول عن إدارة أجهزة الدخل والخرج يسمى مدير الأجهزة (I/O Manager).
- جزء مسئول عن أجهزة التخزين ويسمى مدير الملفات (File Manager).
- جزء مسئول عن التعامل مع الشبكة يسمى مدير الشبكة (Network Manager).

مكونات نظام الحاسب

3. واجهات نظام التشغيل

توفر واجهات نظام التشغيل للمستخدم وتطبيقاته الاتصال مع النواة. وهناك ثلاث أنواع من واجهات نظام التشغيل هي:

- واجهة المستخدم الرسومية (GUI).
- الغلاف (shell) أو مترجم الأوامر (command line interpreter).
- واجهة نداء النظام (system call interface).

الباب الاول المفاهيم العامة (نبذة)

1.10 نداء النظام (System calls)

إذا احتاجت برامج المستخدم خدمة معينة من نظام التشغيل تستخدم ما يسمى نداء النظام **system call** ذلك لأن برامج المستخدم غير مسموح لها بالوصول المباشر للمكونات المادية، وإنما نواة نظام التشغيل هي التي تستطيع فعل ذلك. بهذه الطريقة نضمن سلامة المكونات المادية وحمايتها من البرامج التطفلية ومن الاستخدام الخاطئ لها. ولكن احيانا تحتاج بعض تطبيقات المستخدم التعامل مع المكونات المادية، ولأن هذه التطبيقات لا تستطيع الوصول للمكونات المادية مباشرة، ستقوم بإرسال طلب إلى نظام التشغيل ليمنحها بالمعلومات التي تريدها من المكون المادي المعين.

من نداءات النظام المشهورة في نظم التشغيل لينكس:

open, read, write, close, wait, exec, fork, exit, kill معظم نظم التشغيل اليوم تحتوي على كم هائل من نداءات النظام. مثلا يوجد في لينكس حوالي 203 نداء نظام، وفي FreeBSD توجد حوالي 221 نداء نظام.

الجدول التالي يبين الروتينات (System Calls) في نظام POSIX ، وهي شبيهة بنظام UNIX.

Process management

Call	Description
<code>pid = fork()</code>	Create a child process identical to the parent
<code>pid = waitpid(pid, &statloc, options)</code>	Wait for a child to terminate
<code>s = execve(name, argv, environp)</code>	Replace a process' core image
<code>exit(status)</code>	Terminate process execution and return status

File management

Call	Description
<code>fd = open(file, how,...)</code>	Open a file for reading, writing, or both
<code>s = close(fd)</code>	Close an open file
<code>n = read(fd, buffer, nbytes)</code>	Read data from a file into a buffer
<code>n = write(fd, buffer, nbytes)</code>	Write data from a buffer into a file
<code>position = lseek(fd, offset, whence)</code>	Move the file pointer
<code>s = stat(name, &buf)</code>	Get a file's status information

Directory and file system management

Call	Description
<code>s = mkdir(name, mode)</code>	Create a new directory
<code>s = rmdir(name)</code>	Remove an empty directory
<code>s = link(name1, name2)</code>	Create a new entry, name2, pointing to name1
<code>s = unlink(name)</code>	Remove a directory entry
<code>s = mount(speciaf, name, flag)</code>	Mount a file system
<code>s = umount(special)</code>	Unmount a file system

Miscellaneous

Call	Description
<code>s = chdir(dirname)</code>	Change the working directory
<code>s = chmod(name, mode)</code>	Change a file's protection bits
<code>s = kill(pid, signal)</code>	Send a signal to a process
<code>seconds = time(&seconds)</code>	Get the elapsed time since Jan. 1, 1970

Figure 1-18. Some of the major POSIX system calls. The return code *s* is -1 if an error has occurred. The return codes are as follows: *pid* is a process *fd* is a file descriptor, *n* is a byte count, *position* is an offset within the file, and *seconds* is the elapsed time. The parameters are explained in the text.

Unix System Calls

System Call	Description
access()	This checks if a calling process has access to the required file
chdir()	The chdir command changes the current directory of the system
chmod()	The mode of a file can be changed using this command
chown()	This changes the ownership of a particular file
kill()	This system call sends kill signal to one or more processes
link()	A new file name is linked to an existing file using link system call.
open()	This opens a file for the reading or writing process
pause()	The pause call suspends a file until a particular signal occurs.
stime()	This system call sets the correct time.
times()	Gets the parent and child process times
alarm()	The alarm system call sets the alarm clock of a process
fork()	A new process is created using this command
chroot()	This changes the root directory of a file.
exit()	The exit system call is used to exit a process

Windows System Calls

System Call	Description
CreateProcess()	A new process is created using this command
ExitProcess()	This system call is used to exit a process.
CreateFile()	A file is created or opened using this system call.
ReadFile()	Data is read from the file using this system call.
WriteFile()	Data is written into the file using this system call.
CloseHandle()	This system call closes the file currently in use.
SetTimer()	This system call sets the alarm or the timer of a process
Create Pipe()	A pipe is created using this system call
SetFileSecurity()	This system call sets the security for a particular process
SetConsoleMode()	This sets the input mode or output mode of the console's input buffer or output screen buffer respectively.
ReadConsole()	This reads the characters from the console input buffer.
WriteConsole()	This writes the characters into the console output buffer.

API VERSUS SYSTEM CALL

API

A set of protocols, routines, functions that programmers use to develop software to facilitate interaction between distinct systems

Helps to exchange data between various systems, devices and applications

SYSTEM CALL

A programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on

Allows a program to access services from the kernel of the operating system

الباب الاول المفاهيم العامة (نبذة)

1.11 البنية الاساسية للحاسوب

بالرغم من اختلاف الحاسبات الالكترونية في الحجم، التكلفة والكفاءة، إلا أنها تعتمد في تصميمها على اربعة مبادئ اساسية تعرف بنموذج **جون فون نيومن** وتدعى الآلات المصممة وفق هذا التصميم بآلات **IAS** (نسبة الى معهد الدراسات المتقدمة Institute for Advanced Study).

تتلخص المبادئ الأربعة في :

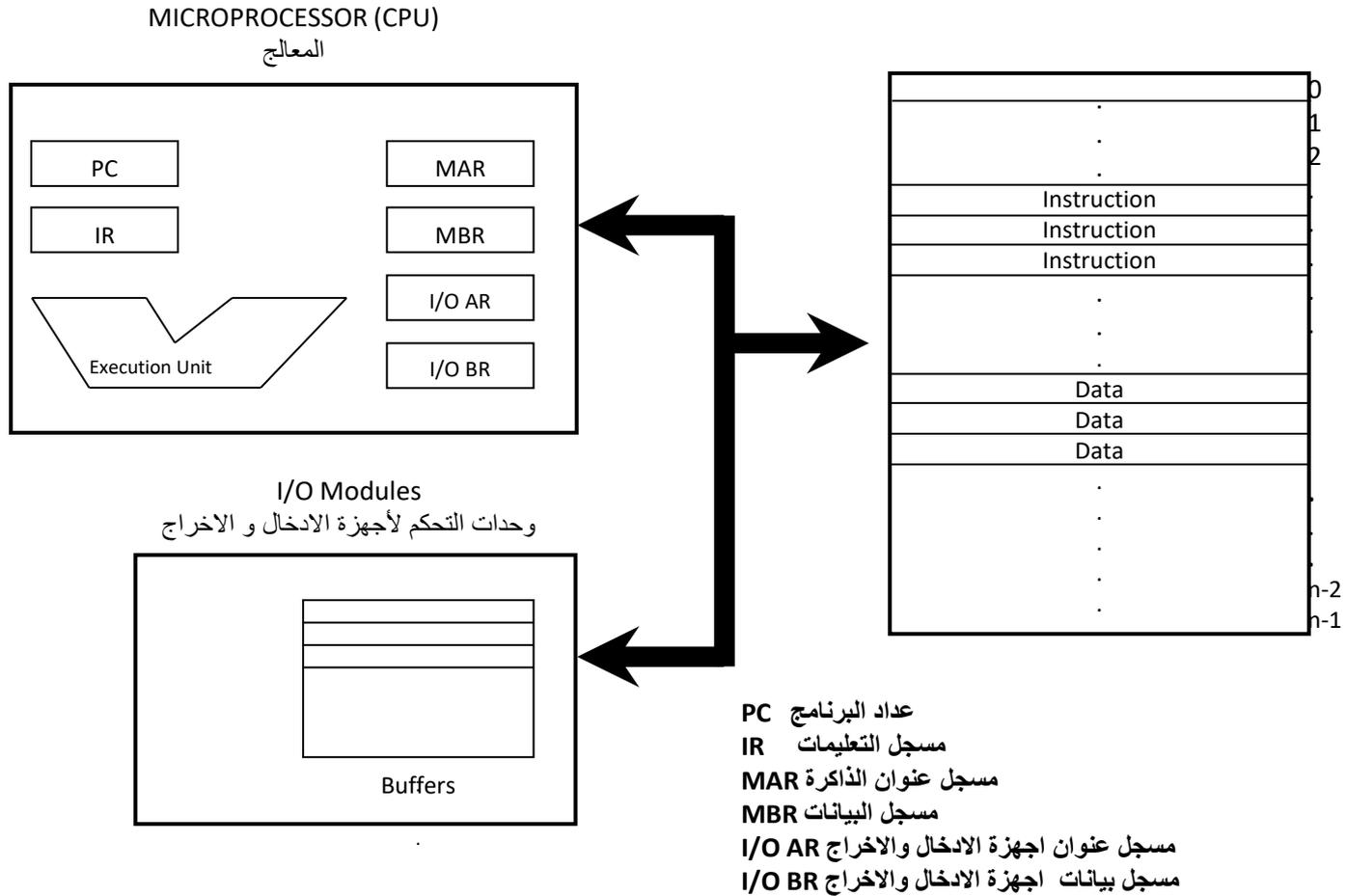
- يجب أن تكون الآلة متعددة الأغراض أي قادرة على حل أنماط مختلفة من المسائل دون الحاجة لتعديل بنيتها الداخلية.
- يجب أن تكون آلة ذاتية العمل: فحل مسألة ما يجب تزويد الآلة بالمعلومات الضرورية منذ البداية ويجب ان تنجز العمل بكامله دون أي تدخل من المستخدم البشري.
- يجب أن تكون آلة إلكترونية تماما، وليس إلكتروميكانيكية.
- يجب أن تكون آلة رقمية، إذ أنه ليس من الممكن إجراء العمليات الحسابية باستخدام تيارات وجهود ذات صيغة تشابهية (Analogue).

الباب الاول المفاهيم العامة (نبذة)

ويمكن معرفة كيفية تجسد هذه المبادئ في بنية الحاسوب كالآتي:

- من المفترض أن تكون آلة حساب، أي يجب أن تتضمن وحدة حساب بمعنى "تستطيع إنجاز عمليات حسابية أولية محددة." وهو ما يعبر عليه بوحدة الحساب والمنطق (ALU).
- أن تكون ذاتية العمل بشكل تام، أي أن تتلقى المعلومات من الوسط الخارجي بدون تدخل بشري وتكون قادرة على تخزين المعلومات واسترجاعها عند الضرورة بل أيضا لها القدرة على الاحتفاظ بالقيم المرحلية. وهو ما يعبر عنه بالمعالج (CPU).
- أن تكون الآلة متعددة الاهداف، بمعنى أنه عند كل استخدام يجب أن نحدد لها نوعية العمل وذلك بتخزين التعليمات الضرورية لإنجاز مهمة معينة بالإضافة للمعلومات الرقمية في وحدة تخزين تسمى الذاكرة (Memory) وتسمى هذه التعليمات بالبرامج المخزنة "Stored Program".
- يجب أن يكون لديها القدرة على تلقي المعلومات اللازمة من المستخدم، ومن ثم تزويد المستخدم بنتائج الحساب أي أن تتضمن وحدات الادخال والايخراج.

يبين الشكل 1.3 العناصر الاساسية بشكل مختزل او موجز وهو ما يعبر عنه بالمشهد الاعلى حيث ترتبط هذه العناصر مع بعضها البعض لا داء مهام الحاسوب والتي هي بالأساس تنفيذ برامج معدة لأغراض معينة. وهذه العناصر هي:



الشكل 1.3 العناصر الأساسية

الباب الاول المفاهيم العامة (نبذة)

- **المعالج** : يتحكم في عمليات الحاسوب المختلفة، ويقوم بإعداد بياناتها، وإذا كان معالج واحد فقط فإنه يطلق عليه وحدة المعالجة المركزية (CPU) Central Processing Unit.
- **الذاكرة الرئيسية (Main Memory)**: وظيفتها تخزين البيانات و التعليمات، وهي تعتبر غير مستقرة "متطايرة (Volatile) " لأنها تفقد البيانات المخزنة بها بمجرد انقطاع التغذية (التيار الكهربى).
- **وحدات التحكم لأجهزة الإدخال و الإخراج (I/O Modules)**: وهي تعمل على نقل البيانات من المعالج الى وحدات الاخراج، والعكس في حالة الادخال.

1.12 تنفيذ التعليمات

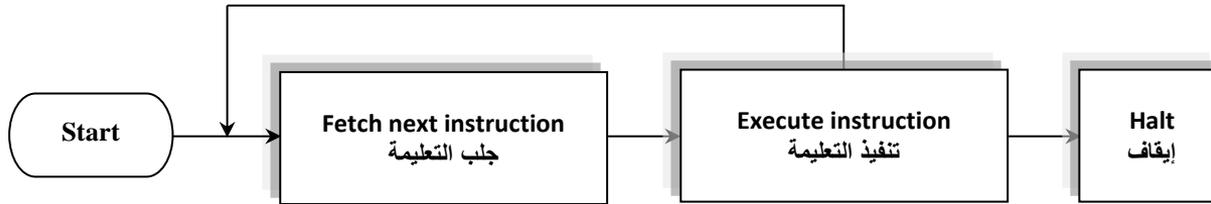
خلال تنفيذ اي تعليمة يجب ان تؤدي وحدة التحكم الوظائف كالتالي:

- **سلسلة التعليمات**: تخزن تعليمات البرامج في مواقع متجاورة في الذاكرة؛ ويشير **عداد عنوان التعليمة الحالية (PC)** الى موقع التعليمة المراد جلبها حالياً مع زيادة مسجل العنوان (**PC**) بقيمة واحد ولهذا يسمى **عداداً**، وهذه الطريقة لا تستخدم في حالة الانتقال الغير متسلسل.
- **جلب التعليمة (Instruction fetch)**: لجلب التعليمة من الموقع المراد نحتاج الى خطوتين:
 - ✓ تزويد **مسجل عنوان الذاكرة (MAR)** بالقيمة المخزنة في مسجل عداد البرنامج (**PC**).
 - ✓ بدء عملية القراءة من الذاكرة وذلك بتفعيل المدخل المناسب من مداخل التحكم بالذاكرة.

الباب الاول المفاهيم العامة (نبذة)

وتتلخص العمليات او الاجراءات في أربعة فئات:

- معالج-ذاكرة (Processor-Memory): يمكن نقل البيانات من المعالج الى الذاكرة والعكس.
- معالج- أجهزة الادخال والإخراج (Processor-I/O): يمكن نقل البيانات من المعالج الى أجهزة الادخال والايخراج بواسطة ما يعرف بوحدات التحكم لأجهزة الإدخال و الإخراج (I/O Modules).
- معالجة البيانات (Data processing): يمكن للمعالج أن يُجْري عمليات حساب ومنطق.
- التحكم (Control): يمكن تغيير نسق التنفيذ بموجب عمليات محددة.



الشكل 1.4 حلقة تنفيذ التعليمة

طرق الاتصال بأجهزة الادخال و الاخراج

طرق الاتصال بأجهزة الادخال و الاخراج

هناك ثلاث طرق لإدارة عملية نقل البيانات بين الذاكرة وأجهزة الادخال و الاخراج:

1. عملية الادخال و الاخراج المبرمجة (Programmed I/O),

2. عبر المقاطعة في عملية الادخال و الاخراج (Interrupt-driven I/O)

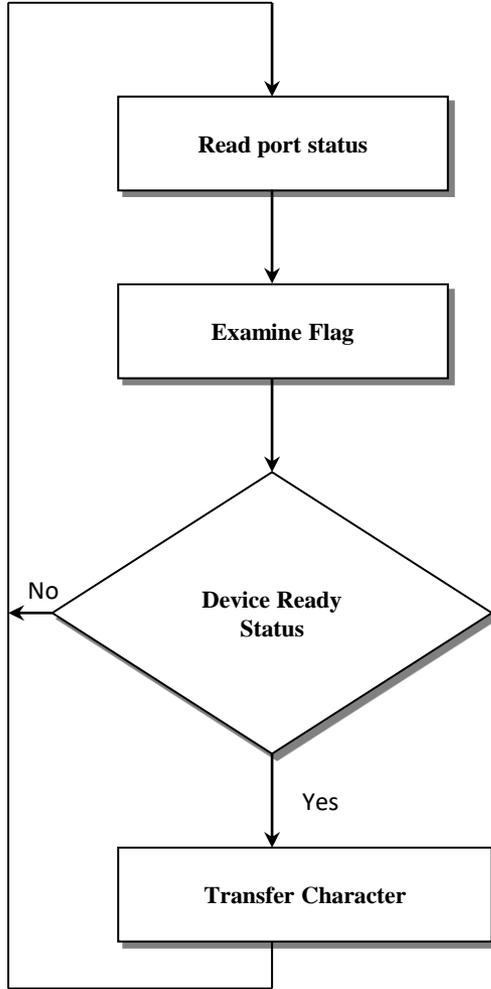
3. الولوج المباشر للذاكرة (Direct Memory Access-DMA).

يعتمد مدى انشغال المعالج في عملية نقل البيانات بين الذاكرة وأجهزة الادخال و الاخراج على نوعية ومقدرة الواجهة (I/O Interface) التي تؤمن عملية الاتصال بأجهزة الادخال و الاخراج.

طرق الاتصال بأجهزة الادخال و الاخراج

1. عملية الادخال و الاخراج المبرمجة (Programmed I/O)

في هذه التقنية يرتبط المعالج ارتباطا تاما مع وحدات الدخل/الخرج. حيث أن كل عملية نقل يتم تنفيذها داخل حلقة البولنج او الاستفتاء (Polling Loop) كما هو موضح بالشكل 1.17.



الشكل 1.17 : عملية الاستفتاء

في هذا المثال يقوم المعالج بعملية نقل البيانات للطابعة وذلك بقراءة مسجل الحالة التابع للواجهة (I/O interface) ليتأكد من ان الطابعة جاهزة لاستقبال الحرف التالي.

عند اختبار المؤشر (Flag) الموجود بهذا المسجل فانه يبين جاهزية الطابعة من عدمها. فاذا كانت غير جاهزة فان المعالج يظل يعيد الاختبار لغاية ان تتغير قيمة المؤشر, عند ذلك يقوم المعالج بنقل الحرف الى مسجل البيانات (Data register).

وفي كل مرة يتم فيها الكتابة في مسجل البيانات تقوم الواجهة (I/O interface) بتغيير قيمة المؤشر لتصبح غير جاهزة مما يؤدي بالمعالج الى اعادة اختبار المؤشر لنقل الحرف التالي.

من عيوب هذه التقنية هو بقاء المعالج في حالة الاختبار هذه لحين اتمام عملية الكتابة او القراءة, الامر الذي يتطلب من المعالج زمنا طويلا من الانتظار (busy-waiting) دون ان يؤدي اي وظيفة اخرى.

طرق الاتصال بأجهزة الادخال و الاخراج

2. المقاطعة (Interruption)

تعتبر هذه التقنية اكثر نجاعة بحيث تسمح للأجهزة بإرسال اشارة تدل على انها بحاجة للخدمة اي انها جاهزة لاستقبال او ارسال البيانات. مما يؤدي الى اعطاء فرصة للمعالج بتنفيذ بعض النشاطات الأخرى.

صممت المعالجات (Processors) الحديثة لتنفيذ تعليمات متتالية، حيث يخزن عنوان اول تعليمة في مسجل خاص يسمى عداد البرنامج (PC) ومن هذا العنوان يتم تنفيذ التعليمات بشكل متتالي وفقا للخطوات التالية :

1. تقوم وحدة التحكم بجلب التعليمة المشار اليها بمحتويات مسجل عداد البرنامج.
2. يتم فك ترميزية التعليمة.
3. زيادة محتويات مسجل عداد البرنامج بقيمة تمثل طول التعليمة الحالية.
4. تنفيذ التعليمة.

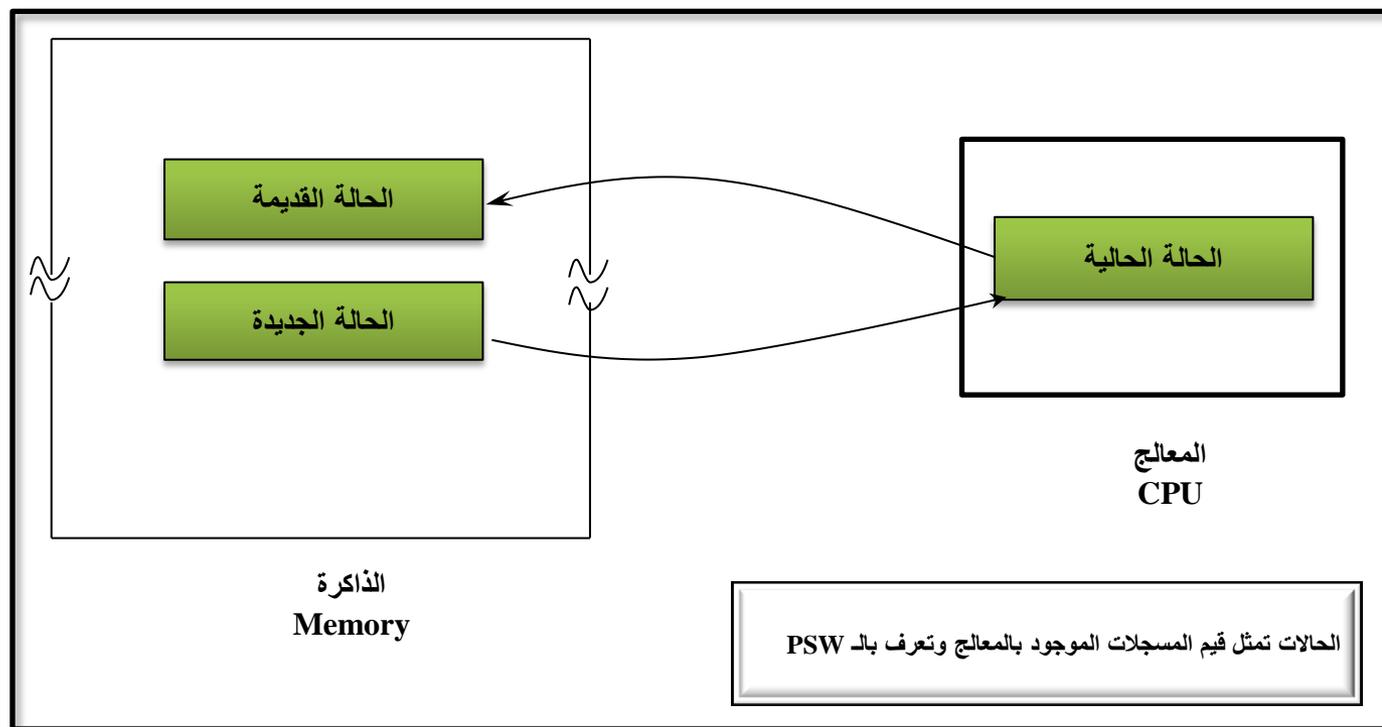
يكرر المعالج الخطوات السابقة 1-4 لغاية حدوث احدى امرين.

- التعليمية الحالية تقوم بتغيير التسلسل لاحتوائها على جملة **goto** او اي جملة شرطية اخرى.
- حدوث حدث (مقاطعة) يتطلب انتقال المعالج لتنفيذ اجراء ما.

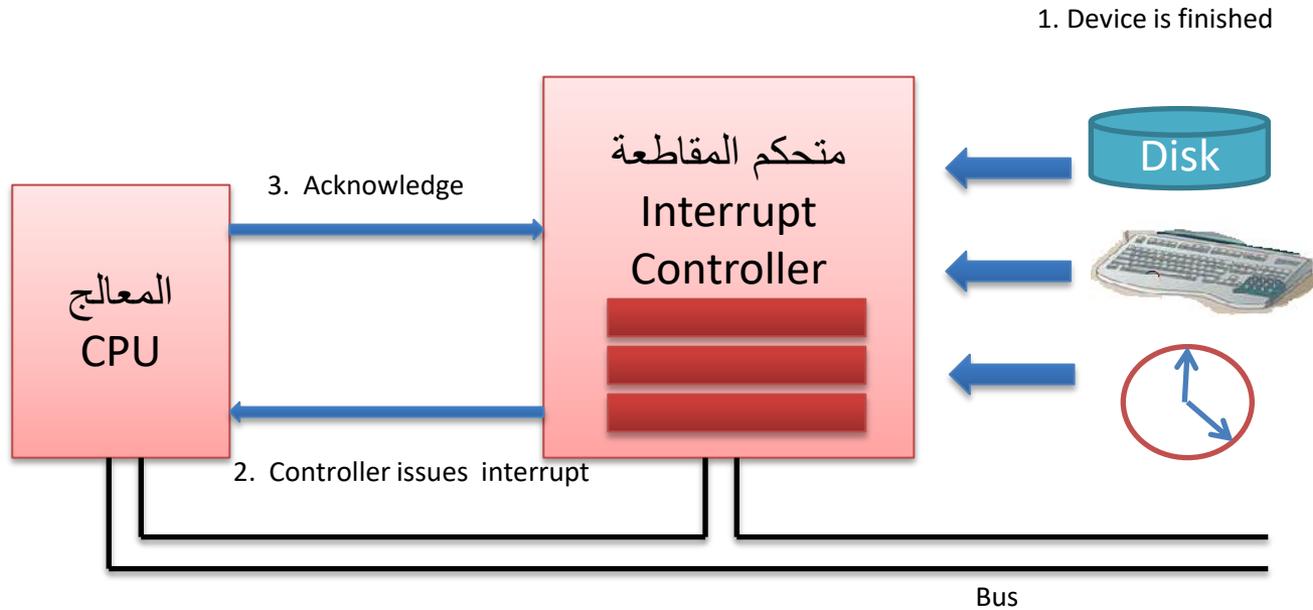
مثل هذا الحدث يسمى مقاطعة (**Interruption**)، ويمكن تلخيص اسباب حدوث او انواع المقاطعات في الجدول التالي :

تتكون نتيجة لتحقق شرط معين أثناء تنفيذ تعليمة ما مثل الفيض (Overflow)، القسمة على الصفر، محاولة تنفيذ تعليمة غير مسموح بها و الوصول الي عنوان خارج المساحة المحددة للبرنامج والبيانات.	البرنامج (Program)
إشارة الساعة الداخلية التي تنظم تنفيذ المهام على فترات متساوية.	التوقيت (Timer)
تنتجها وحدة التحكم الخاصة بأجهزة الادخال و الاخراج وهي اشارة اتمام عملية كتابة أو قراءة أو نتيجة حدوث أخطاء معينة.	دخل/خرج (I/O)
تحدث نتيجة فقدان التغذية (Power) او خطأ في بيانات الذاكرة (Parity error).	عطب او قصور في مكون مادي (Hardware Failure)

والمقاطعة تستخدم لتساعد في الاستفادة من وقت المعالج، حيث يمكن للمعالج الانتقال لتنفيذ برنامج اخر. وهي ضرورية وذلك لبطيء اجهزة الادخال و الاخراج. عند حدوث المقاطعة فانه يوجد مكانان لاستقبال المعلومات الضرورية لانتقال المعالج من اجراء معين الى اجراء اخر (انظر الشكل التالي) وهي طريقة مبسطة لتقريب مفهوم المقاطعة وما يحدث عند حدوثها.



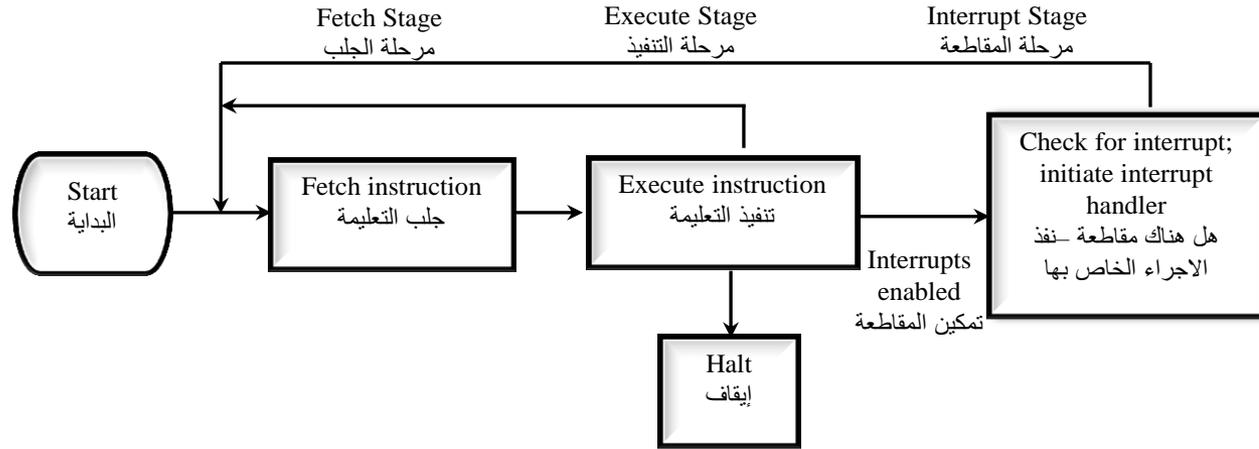
عند حدوث المقاطعة فان الاجهزة ترسل هذه المقاطعة عبر متحكم المقاطعة الذي بدوره يحدد الاولوية ويعلم المعالج بوجود مقاطعة، حيث يقوم المعالج بتجاهل هذه المقاطعة لحين اتمام التعليمات قيد التنفيذ. ثم بعد ذلك يقوم باختبار هذه المقاطعة ليقرر تنفيذها او تجاهلها. فاذا ما قرر تنفيذها فانه ينتقل الى تنفيذ البرنامج الخاص بالمقاطعة (**Interrupt handler or service routine**).



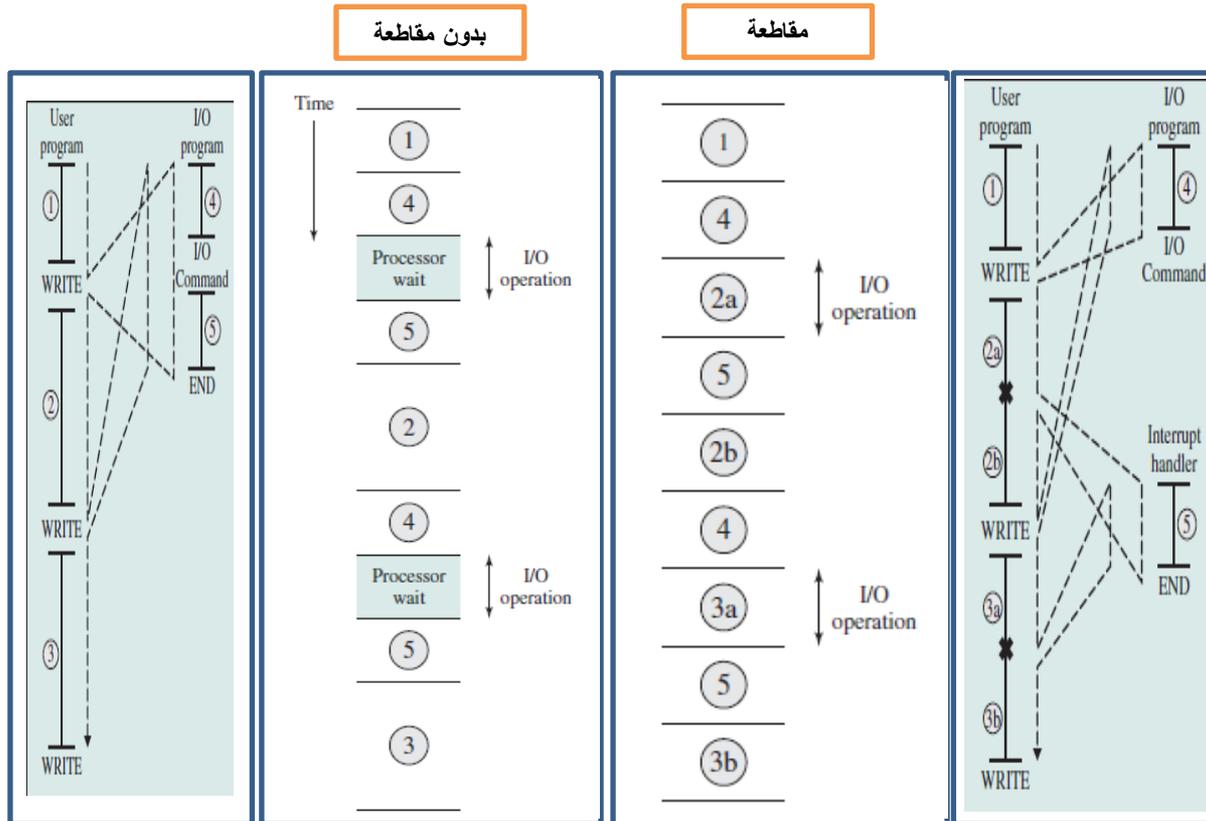
يوجد لكل مقاطعة برنامج خاص يخزن في الذاكرة ويتم الوصول اليه وتنفيذه عن طريق جدول مرقم من $n-0$ طبقا لنوع المقاطعة، حيث تؤشر محتويات كل خانة من الجدول بداية البرنامج الخاص بالمقاطعة يسمى مثل هذا الجدول بالـ **Table vector**.

ملاحظة : تعتمد طريقة المقاطعة على بنية الحاسوب ، فمثلا قد لا يتوفر متحكم المقاطعة الذي ذكر في الشكل السابق وقد يتعامل متحكم الاجهزة مباشرة مع المعالج.

قبل الانتقال لتنفيذ أي مقاطعة (Service routine) يقوم المعالج بحفظ البيانات التي تشمل محتويات المسجل (PC) و مسجل الحالة (SR) في المكس (Stack). كما يجدر الإشارة الى انه لكي يتم تشغيل المقاطعة لابد ان يتم تضمينها في بنية المعالج كما هو موضح بالشكل التالي:



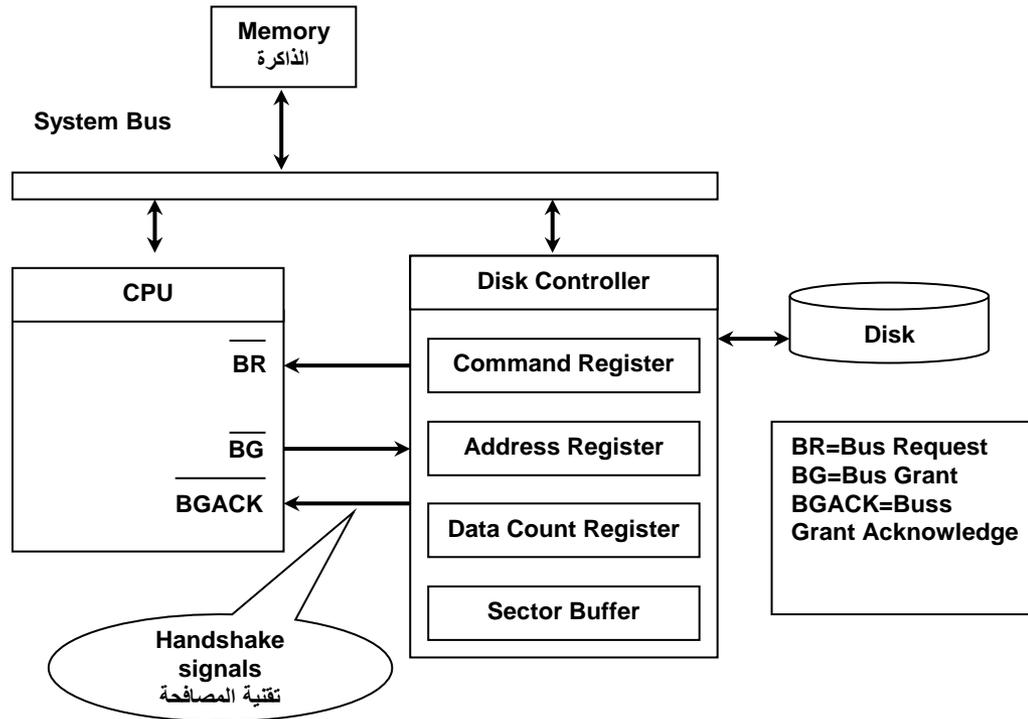
مثال يوضح الفرق بين استخدام المقاطعة و بدون استخدام المقاطعة



طرق الاتصال بأجهزة الادخال و الاخراج

3. تقنية الولوج المباشر للذاكرة (Direct Memory Access-DMA)

من عيوب تقنية المقاطعة هو استمرار نقل البيانات من مسجلات المعالج بالإضافة للأعمال الإضافية (**Overheads**) التي تتطلبها حفظ مسجل العداد (**PC**) وباقي المسجلات الخاصة بالمعالج. عند التعامل الكثيف مع أجهزة الادخال و الاخراج يصبح المعالج بطيء، وكحل لهذه المسألة تم استخدام تقنية الولوج المباشر للذاكرة (**DMA**)، وهي تعمل على نقل البيانات مباشرة من الذاكرة وبكميات كبيرة تعرف بالقوالب او الكتل (**Blocks**).



فاذا كانت العملية هي عملية "قراءة" من القرص الصلب فان المعالج يقوم بتحميل (الشكل السابق) :

1. **عنوان المحتجز (Buffer)** وهو مكان بالذاكرة لاستقبال البيانات الى مسجل العنوان (Address register) الخاص بوحدة التحكم في القرص (Disk controller or DMA).
2. **عدد الكلمات المراد نقلها (Word count)** الى مسجل عدد الكلمات (Data count register).
3. ارسال بعض **البايتات (Bytes)** التي تمثل العنوان المراد نقل البيانات منه ونوع العملية (في هذه الحالة قراءة) الى مسجل الامر (Command register).

تستجيب وحدة التحكم بتحريك رأس القراءة والكتابة (ق/ك) الى العنوان المراد نقل البيانات منه, ثم تبدأ عملية نقل سيل من البتات (Bits) الى المحتجز الخاص بوحدة تحكم القرص (Sector buffer). بعد ذلك تقوم وحدة التحكم DMA بنقل البيانات الى المكان المخصص والموجود في مسجل العنوان (Address register) وبعدها الكلمات المعطاة في مسجل عدد البيانات (Data count register) بمساعدة تقنية المصافحة (Handshake signals) كما هو موضح بالشكل (السابق).

بعد اتمام عملية النقل ترسل وحدة التحكم (DMA) اشارة مقاطعة للمعالج لإعلامه بجاهزيتها لاستقبال اي عملية اخرى.

تسمى العملية السابقة بالـ Burst transfer mode, اي النقل المتتابع للكلمات لغاية اتمام عملية النقل بالكامل. بالرغم من سرعة هذه الطريقة الا أنها تتسبب في تعطيل المعالج لان خطوط نقل البيانات (System Bus) مشغولة من قبل الـ DMA اثناء عملية النقل.

في اغلب الانظمة طورت هذه التقنية للإبطاء المعالج بدلا من ايقافه وذلك باعتماد طريقة سرقة زمن بسيط يسمح بنقل البيانات على دفعات (في كل دفعة يتم نقل كلمة واحدة), وتسمى عملية السرقة بالـ Cycle Stealing Mode. في اثناء عملية السرقة هذه قد يحتاج المعالج لخطوط النقل (System Bus) فانه يحرم اثناء نقل كلمة واحدة ثم يستأنف المعالج نشاطه.

الترتيب الهرمي للذاكر

تنقسم الذاكر الى عدة مستويات (Levels) كل مستوى له سرعة و حجم مختلف. التقنية المستخدمة في تصميم الذاكر تشمل اشباه الموصلات (Semiconductors) للذاكر الداخلية (Internal) وهي التي تكون قريبة من المعالج (CPU) كما هو موضح بالشكل 1.15, وللذاكر الخارجية الابطى (External) تستخدم تقنية السطوح الممغنطة و البصرية (Magnetic and Optical) والتي تبتعد عن المعالج.

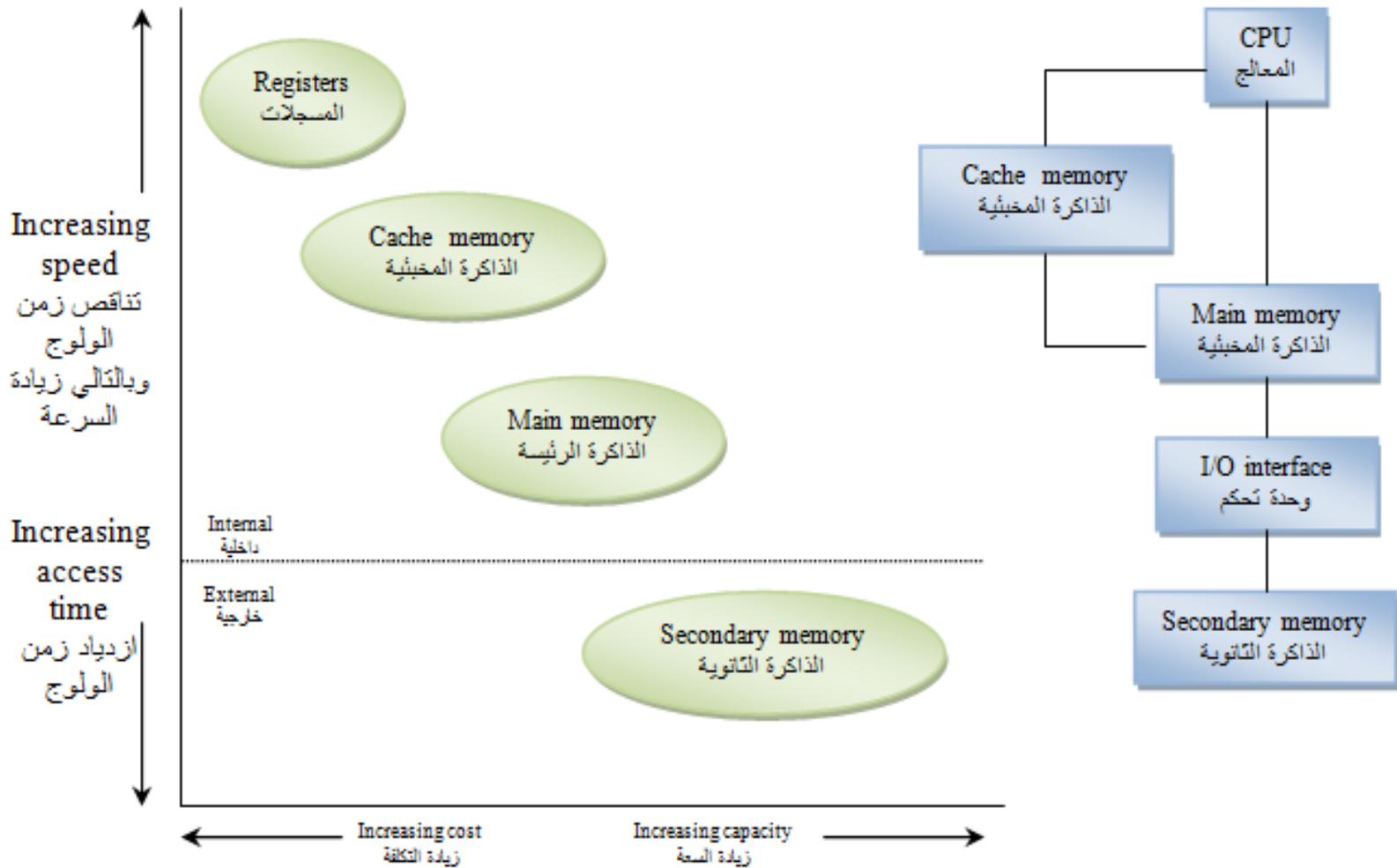
الشكل (1.15) نموذج الترتيب الهرمي :

ذاكرة المعالج (Processor memory)

تتمثل ذاكرة المعالج في عدد من المسجلات (Registers) التي تحتل الترتيب الاعلى. توفر هذه المسجلات سعة محدودة وهي تستخدم للتخزين المؤقت للتعليمات (Instructions) و البيانات (Data).

الذاكرة الرئيسية (Main memory)

الذاكرة الرئيسية لها سعة اكبر بكثير من ذاكرة المعالج (وهي في العادة ما بين 1-8 ميغا بايت) وهي ايضا يمكن الولوج لمحتوياتها مباشرة من قبل المعالج (CPU), ويتم تصميمها بتقنية DRAM. وحيث أن سرعة الولوج للبيانات في الذاكرة الرئيسية تمثل تقريبا عشر سرعة المعالج، لذلك فهي تمثل مختنق يؤثر على سرعة المعالج.



الشكل 1.15 الترتيب الهرمي للذواكر

الذاكرة المخبئية (Cache memory)

الذاكرة المخبئية وهي تستخدم للتخفيف من المخبئق السابق ذكره بين الذاكرة الرئيسة والمعالج. حيث تتوسط ما بين المعالج والذاكرة الرئيسة (الشكل 1.15) ولها سرعة افضل من سرعة الذاكرة الرئيسة وتصنع من تقنية **SRAM** وهي اعلی ثمننا. تستخدم الذاكرة المخبئية لتخزين الاجزاء الاكثر استخداما سواء كانت هذه الاجزاء من البرنامج او البيانات. و تختبر الذاكرة المخبئية في كل حالة جلب للتعليقات او البيانات, فاذا وجدت هذه المعلومات في الذاكرة المخبئية فانه يتم احضارها الى المعالج وتسمى عملية "Hit", وان لم تتوفر فالبحت سوف يجري في الذاكرة الرئيسة وتسمى عملية "Miss" (الشكل 1.18)

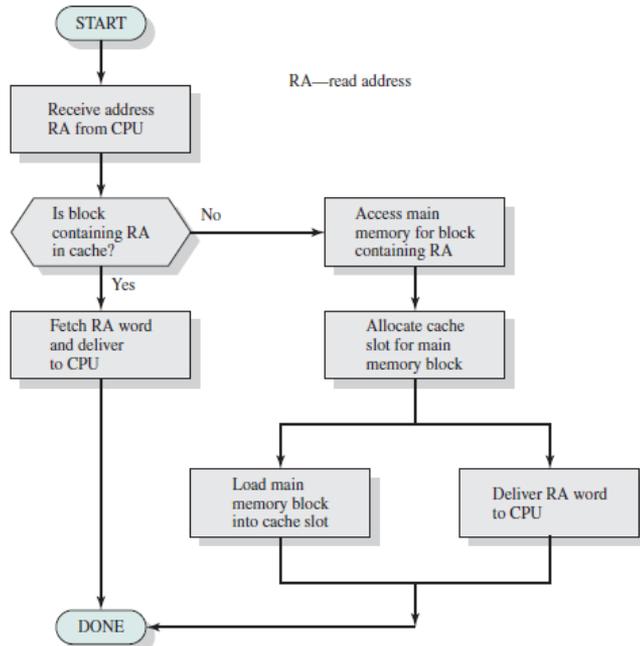


Figure 1.18 Cache Read Operation

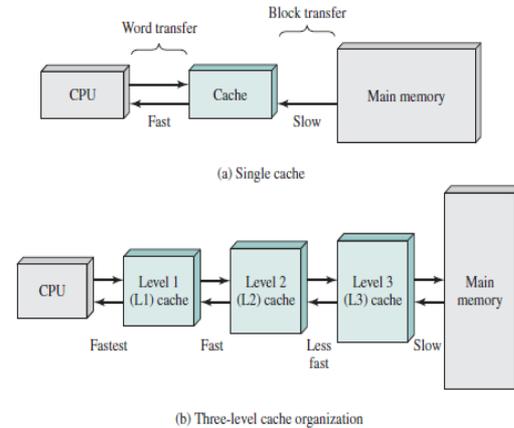


Figure 1.16 Cache and Main Memory

كما تحتوي بعض الانظمة على عدة مستويات من الذاكرة المخبئية كما هو موضح بالشكل 1.16.

الذواكر الثانوية (Secondary memory)

الذواكر الثانوية هي التي ليس لها اتصال مباشر بالمعالج وبالتالي تسمى خارجية (External), وتشمل الأقراص و الاشرطة الممغنطة بصفة عامة (الصلب, المرن, البصرية). يتم التواصل مع هذه الذواكر عن طريق وحدات التحكم المعروفة بالواجهات (I/O interfaces). توفر هذه الذواكر سعة كبيرة (مقارنة بالذواكر السابقة) لتخزين البرامج والبيانات, وهي تتميز بانها ذواكر مستقرة بمعنى انها لا تفقد محتوياتها بعد اطفاء الجهاز على عكس الذواكر السابقة فهي غير مستقرة (Volatile).

يمكننا تلخيص أن تصميم الذواكر يعتمد على (الشكل 1.15) :

1. **السعة :** وهي كمية البيانات التي يمكن تخزينها في الذاكرة.
2. **السرعة :** وهي زمن الولوج او الوصول للبيانات.
3. **التكلفة :** ثمن تكلفة البت (Bit).

من البديهي ان يكون اختيار الدواكر باختيار الافضل مواصفات وذلك بان يكون الاسرع في زمن الولوج, والاكثر سعة, والاقل تكلفة, الا ان هذا الاختيار لا ينطبق على الواقع وذلك للأسباب الآتية:

- الاسرع في زمن الولوج هو الاغلى ثمننا.
- والاكثر سعة اقل تكلفة.
- والاكثر سعة يحتاج الى زمن ولوج اكثر.

يمكن ترتيب الدواكر حسب السعة من الاقل الى الاكثر سعة، كما هو مبين بالشكل (1.15) حيث نلاحظ ان العلاقة بين العناصر الثلاث السابقة هي كالاتي كلما اتجهنا الى الاسفل:

- تتناقص التكلفة.
- تزداد السعة.
- يزداد زمن الولوج للبيانات.
- تتناقص عدد مرات الولوج للبيانات.

أنظمة المعالجات المتعددة (Multiprocessors and Multicore Organization)

لعله من المعروف ان تنفيذ التعليمات تأخذ النمط التتابعي اي التسلسلي وان كان هناك بعض الانماط توفر نوع من التنفيذ المتوازي مثل تداخل جلب التعليمات مع تنفيذها, كما ان الاشارات التي تحدث دون اي ترتيب معين. ولغرض زيادة قدرات الحواسيب وتقليل وقت تنفيذ البرامج وذلك بتنفيذ عدة مهام في وقت واحد تم استنباط تقنية المعالجات المتعددة بطريقتين.

1. المعالجات المتماثلة (Symmetric Multiprocessors-SMP)

يمكن تعريف هذه التقنية بمعرفة خواصها الاتية (انظر الشكل 1.19):

- هناك اثنان او اكثر من المعالجات لها نفس المواصفات.
- تتشارك في الذاكرة واجهزة الادخال والاخراج وترتبط هذه الوحدات بخطوط اتصال عامة (System Bus).
- تتشارك هذه المعالجات في اجهزة الادخال والاخراج ام عن طريق نفس القنوات.
- كل المعالجات تنفذ نفس المهام او الوظائف (Functions).
- نظام تشغيل موحد يضمن ادارة الموارد من الملف الى اقل وحدة تعامل وهي التعليمات او البيانات.

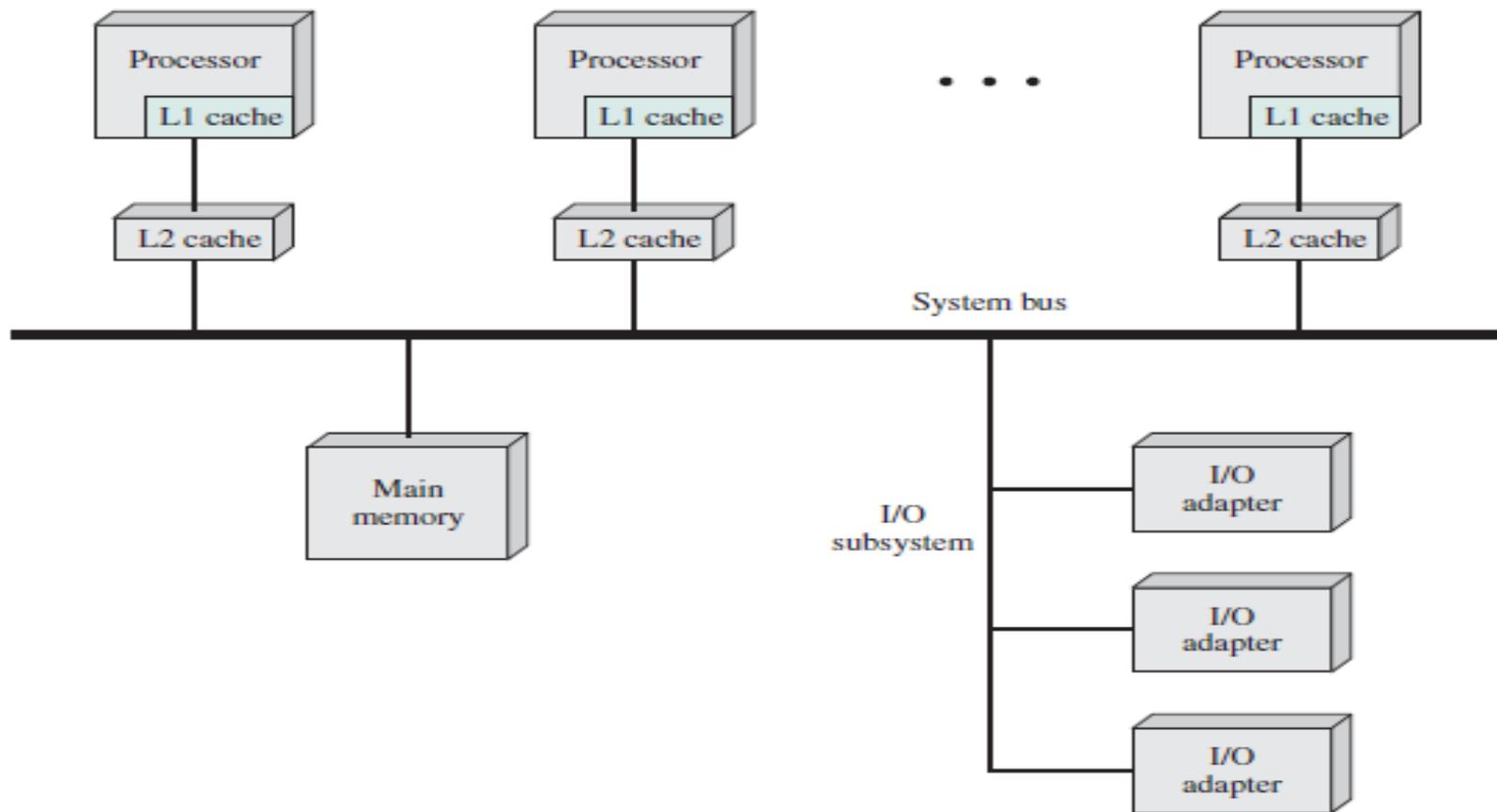


Figure 1.19 Symmetric Multiprocessor Organization

مميزات تقنية المعالجات المتعددة

- **الكفاءة (Performance):** اذا قسم العمل الى عدة اجزاء يمكن تنفيذه على التوازي فان كل جزء يمكن تنفيذه في معالج منفصل وبالتالي فان كفاءة النظام تتحسن.
- **التوفر (Availability):** عند توقف معالج ما فان النظام ككل لا يتوقف ولكن الكفاءة تتأثر.
- **التطور المتزايد او الاضافي (Incremental growth):** اذا كانت البنية الاساسية للحاسوب تسمح بتركيب معالجات اضافية فان الكفاءة حتما تتحسن.
- **التوسع (Scaling):** بمعنى زيادة قدرات المعالجات في حد ذاتها تؤدي الى تحسين اداء النظام ككل.

المعالجات المتعددة من نوع Multicore

وهي وجود اثنان او اكثر من المعالجات ("Cores") في شريحة واحدة بحيث يشتمل كل معالج على نفس مواصفات المعالج العادية مثل المسجلات ووحدة الحساب والمنطق ووحدة التحكم بالإضافة للذاكرة المخبئية (L1), كما يمكن اضافة ذواكر مخبئية (L2, L3).
الشكل 1.20 يوضح شريحة واحدة تضم 4 معالجات بذاكرة مخبئية (L1, L2) لكل معالج مع ذاكرة مخبئية (L3) مشتركة.

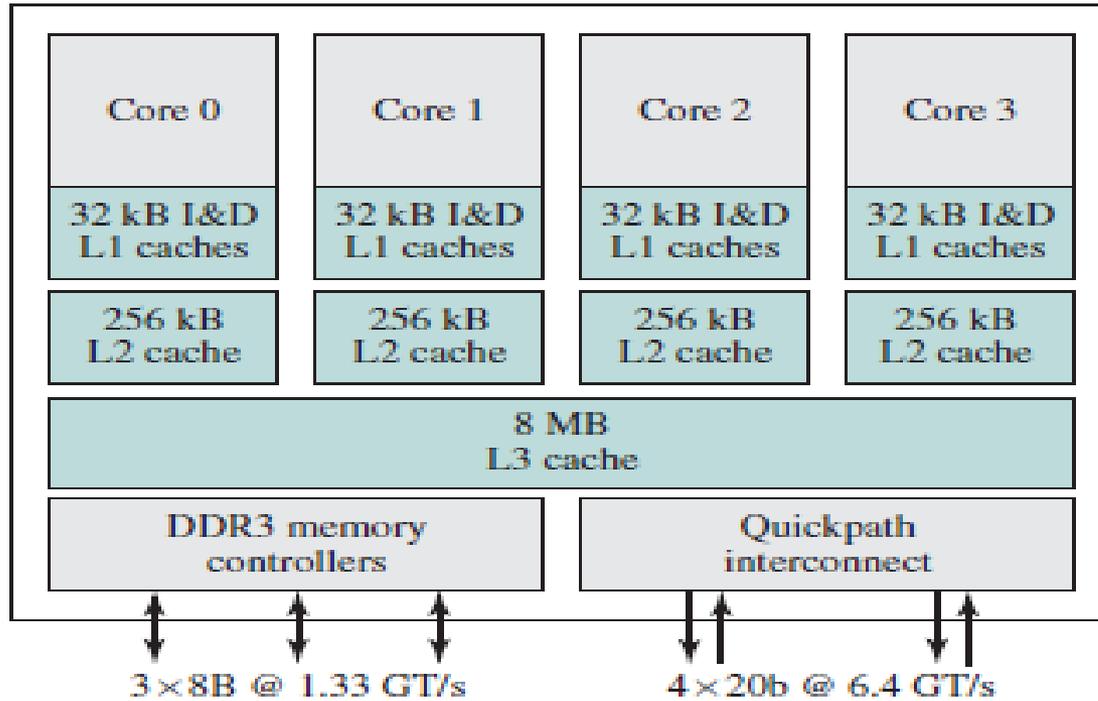


Figure 1.20 Intel Core i7 Block Diagram

2. المعالجات غير المتماثلة Asymmetric multiprocessor

يوجد معالج رئيسي واحد يتحكم في النظام وينفذ مهمة رئيسية كبيرة، بينما بقية المعالجات تنفذ ما يأمر به هذا المعالج الرئيسي، فهو قد يقسم المهمة الكبيرة إلى أجزاء صغيرة يوزعها على بقية المعالجات ثم يجمع النتائج بعد التنفيذ.

الأنظمة الموزعة (Distributed systems)

توزيع العمل عبر الشبكة إلى عدة حاسبات، هذه الحاسبات قد تكون قريبة أو بعيدة، وكل حاسب لديه معالجه، ذاكرته، وقرصه وأجهزته الطرفية الخاصة به. وقد تكون هذه الأجهزة متباينة في المكونات المادية ونظم التشغيل التي تديرها.

الميزات:

- التشارك في الموارد.
- زيادة سرعة التنفيذ.
- توزيع الحمل بين هذه الحاسبات (load balancing)

العيوب:

- زمن الاتصال بين أجزاء النظام قد يؤثر على أداء النظام ككل.

الأجهزة المتجمعة (Clustered Systems)

هي مجموعة من الأجهزة المتواجدة في مكان واحد والمتصلة مع بعضها البعض بشبكة محلية سريعة وغالبا ما تكون متشابهة في المكونات المادية ونظم التشغيل التي تديرها. تتشارك في التخزين (قد يكون هناك جهاز واحد بقرص صلب والبقية بدونه مثلا).

الأجهزة ذات الزمن الحقيقي (Real-time)

هي حواسيب موجودة في أجهزة تحكم مثل:

- أجهزة تجميع السيارات.
- الماكينات.
- عملية الطيران .
- إطلاق الصواريخ.
- النظم الطبية.
- الإنسان الآلي (Robotics).

تتصف نظم التشغيل التي تدير مثل هذه الحواسيب بقيد زمني، حيث لا بد من أن يتم التنفيذ في فترة زمنية محددة، لان التنفيذ مرتبط بعمل يجب أن ينجز في وقت معين وقد يتسبب في تلف ما إن نفذ في وقت متأخر أو متقدم عن الزمن المحدد له.

الأجهزة الكفية (hand held)

هذه الأجهزة صغيرة غالبا ما تحمل في الجيب أو في الكف ومن هنا جاءت تسميتها بالأجهزة الكفية (hand held أو الجيبية) ، (pocket PC) ، وتتصف بالآتي:

- حجم صغير.
- ذاكرة محدودة.
- معالج بطيء.
- شاشة صغيرة.
- بطارية محدودة التشغيل.

أمثلة لها: I-mate, Nokia E95, Toshiba e750

نظام التشغيل لهذا النوع من الأجهزة يدعم المحادثات التلفونية والتصوير الرقمي وتبادل البيانات مع الأجهزة الأخرى عبر الشبكات اللاسلكية والبلوتوث. كما يهدف نظام التشغيل هنا على المحافظة على البطارية، والتعامل مع الشاشات الصغيرة وتوفير طرق سهلة للتعامل مع الأوامر (في شكل نقرات دون الحاجة لاستخدام لوحة المفاتيح التي يكون التعامل معها صعبا لصغر حجمها).

أمثلة لنظم التشغيل التي تعمل على هذا النوع من الأنظمة:

• الأنظمة المضمنة (Emedded Systems)

هي حاسبات ذات أغراض معينة، صممت لتقوم بوظيفة واحدة، وغالبا ما تكون مضمنة داخل جهاز تتحكم في عمله، مثل التلفزيون، السيارة، المايكرويف، مشغل MP3، الموجهات (routers)، إشارات المرور، مسجلات DVD، وغيرها.

بعض هذه الحاسبات تحتوي على نظام تشغيل ينفذ أعمال تحكمية.

الفرق بين الحاسبات الكفية والمضمنة أن الأخيرة لا يمكن إضافة برامج جديدة إليها، فهي تأتي ببرامجها مخزنة في ذاكرة ROM من الشركة.

• أنظمة البطاقات الذكية (Smart card Systems)

البطاقات الذكية هي كروت بلاستيكية بحجم البطاقة الائتمانية مزودة برقاقة إلكترونية صغيرة لها القدرة على معالجة المعلومات، وهذا يعني أنها تمتلك القدرة على استقبال البيانات أو المدخلات ومعالجتها من خلال البرمجيات المثبتة على هذه الشريحة. تمتلك هذه البطاقة ذاكرة وطاقة تحويل إلكتروني تتخذ أساليب دفع مختلفة، مثل الدفع الإلكتروني كبطاقة ائتمانية، التأمين الصحي، التعرف على الشخصية (بدل البطاقة الشخصية) وغيرها من المهام.

شريحة البطاقة الذكية تستطيع معالجة البيانات بالإضافة إلى تخزينها. نظام التشغيل المصمم لهذا النوع من الأجهزة يعتبر صغير جدا. تنتشر البطاقة الذكية في أوروبا أكثر من إنتشارها في أمريكا.



شكل رقم (9-1): بطاقة ذكية تستخدم للتأمين الصحي بفرنسا

أنظمة الخادم Server Systems

هي أجهزة سريعة وقوية تستخدم لتوفير خدمات لبقية الأجهزة في الشبكة. هنالك نظم تشغيل خاصة بهذه الأجهزة مثل Windows Advanced Server ،Unbuntu Server، حيث تدعم هذه النظم أهم صفة يتصف بها الخادم وهي إتاحة المشاركة بين المستخدمين وخدمة أكبر عدد منهم في وقت واحد. مثلا مخدم الويب (web server) يسمح لعدد كبير من المتصفحين من الاتصال وتصفح المواقع.

الأنظمة أعلاه متنوعة في الشكل والغرض وبالتالي لابد من إدارة كل نوع بطريقة مختلفة. وبالتالي يختلف الهدف من بناء نظام تشغيل باختلاف هذه الأنظمة.

تمارين محلولة

أختار الإجابة الصحيحة (قد تكون هنالك أكثر من إجابة صحيحة)
تنقسم نواة نظام التشغيل إلى 5 أجزاء رئيسية منها:

- مدير الشاشة
- مدير الأجهزة. (1)
- مدير القرص الصلب.
- مدير القرص الضوئي

ما الذي يعتبر واجهة من واجهات نظام التشغيل:

- سطح المكتب (1)
- المتصفح
- واجهة نداء النظام (1)
- لا شيء مما ذكر.

مثال لنظام تشغيل أحادي البرامج :

- DOS (1)
- ويندوز XP
- لينكس
- ماك

من نظم التشغيل التالي:

- رينكس
- مفاهيم نظم التشغيل
- زينكس
- ماك
- لا شيء مما ورد (1)

يتميز تعدد المعالجات بأنه:

- كثير التكلفة
- سريع (1)
- معقد
- إذا تعطل معالج فسيتوقف النظام

من الميزات التي لا توجد في الأنظمة الموزعة :

- التشارك في الموارد.
- زيادة سرعة التنفيذ.
- توزيع الحمل بين هذه الحاسبات
- لا شيء مما ذكر (1)

الأجهزة الكفية (hand held) تتصف بالآتي:

- حجم صغير. (1)
- ذاكرة كبيرة.
- معالج سريع.
- شاشة صغيرة. (1)

من أمثلة نظم تشغيل الحاسبات الكفية :

- بالم (Palm) (1)
- ينكس (Unix)
- سيمبيان (Symbian) (1)
- زينكس (Zenex)

الفرق بين الحاسبات الكفية والمضمنة أن الأجهزة المضمنة:

- صغيرة المعالج
- ذاكرتها محدودة
- لا يمكن إضافة برامج جديدة إليها (1)
- برامجها مخزنة في ذاكرة ROM من الشركة. (1)

أجب بنعم أو لا (مع تصحيح الإجابة الخاطئة)

لا يؤثر الاتصال بين أجزاء النظام الموزع على أداء النظام. لا ، يؤثر

تعتبر الأنظمة المجمعة (cluster) مجموعة من الأجهزة المتواجدة في أماكن مختلفة والمتصلة مع بعضها البعض بشبكة عالمية. لا ، متواجدة في مكان واحد ومرتبطة بشبكة محلية.

في تعدد المعالجة (multiprocessor) يمكن تعريف المعالجات المتماثلة بأنها مجموعة من المعالجات بحيث يكون فيها معالج رئيسي واحد يتحكم في النظام وينفذ مهمة رئيسية كبيرة، بينما بقية المعالجات تنفذ ما يأمرها به هذا المعالج الرئيسي. لا ، المعالجات غير المتماثلة.