CH.2 Operating-System Structures

* Operating System Services

- OS provides an environment for the execution of programs.
- It provides certain services to programs and to users of those programs.

⇐ نظام التشغيل بتقدم الخدمات للبرامج والمستخدمين هاي البرامج.

① User Interface

⇐ من أهم الخدمات اللي بتقدمها الـ OS هي اليوزر إنترفيس (واجهة المستخدم)

User Interface: something that allows the user to interact with the operating system or to interact with the computer itself.

⇒ Command Line Interface (CLI)

⇒ Graphical user Interface (GUI)

② Program Execution

⇐ نظام التشغيل بنفذ البرامج بمعنى إنه بنقلها على الهارد عشان تتنفذ هناك (.exe)

③ I/O Operations

⇐ نظام التشغيل بتحكم باستخدام أجهزة الـ I/O، يعني هو اللي بنظم المدخلات والمخرجات، وهو بكونه وسيط بين المستخدم والبرامج وبين الأجهزة

④ File-System Manipulation

The operating system must control how the files are manipulated or how this file are managed. The operating system controls:

1- Create files
2- Delete files
3- Modify files
4- Search for a file
5- Permissions
6- Restrections of access

⇐ يعني نظام التشغيل بتحكم بكل اشي بخص الفايلات من فتح وإنشاء وحذف والخصوصية وغيرها.

⑤ Communications

: Communications between processes

⇐ بكون في عدة عمليات تبعر وهاي العمليات محاجة إنها تتواصل مع بعضها عشان تتم هاي العمليات بطريقة صحيحة ومتعالية، فهاي العمليات بتتم التحكم بتواصلها مع بعض عن طريق نظام التشغيل

⑥ Error Detection

The operating system needs to be constantly aware of possible errors that can occur like CPU errors, memory errors, I/O Devices errors and so on.

⑦ Resource Allocation

: Allocating resources to different processes or different users.

Resources: CPU, Files, I/O Devices, Main memory, ... etc

⇐ في عدة عمليات تبعر على الجهاز، وجميع هاي العمليات بتحتاج ريسورس معينة، فبتقوم نظام التشغيل إنه عم يقوم بتوزيع هاي الموارد بطريقة متعالية بحيث جميع العمليات تاخذ الريسورس اللي بدها إياها ووقت عالية تنتهي ريسورس وما تاخذه

⑧ Accounting

what we means by Accounting is that we want to keep track of which users use, how much, and what kind of computer resources. By having a statistics of this usage it can be a valuable tool for researchers who wish to recognfigure the system or to improve the computing services.

⇐ الفكرة من هاي الخدمة إنها بتوفر زي إحصائيات عن استخدام اليوزرز للجهاز ومواردة، الهدف منها إنه الباحثين يشوفوا هاي الإحصائيات فبالتالي يقدروا يطوروا خدمات الجهاز

⑨ Protection and Security

① When processes or several different processes are executing at the same time, It should not be possible for one process to interfere with the others or with the operating system itself

② ensuring that all access to system resources is controlled : protection

③ The system is not accessed from outsiders who are not allowed to access the system

④ The security extends to defending your external I/O Devices like modems and your network adaptors and so on (you should make sure that these are prevented from invalid access attempts)

* User Operating system Interface

① **Command-Line Interface (CLI) or command Interpreter:**
allows users to directly enter commands that are to be performed by the operating system

② **Graphical User Interface (GUI):**
Allows the user to interface with the operating system with GUI:
- user-friendly kind
- Includes buttons and Icons

⇒ CLI :

- The command Interpreter is included in the Kernal in some OSs.
- Windows XP and UNIX treat the command Interpreter as a special program
- On system with multiple command interpreters to choose from, the interpreter are known as **shell**
  - Bourne shell
  - C shell
  - Bourne-again shell (Bash)
  - korn shell, etc

⇐ لاحظ هنا انه الـ CLI بتحتوي على Command Interpreter ، هو بكون موجود في الكيرنال في بعض نظم التشغيل (الكيرنال هو باختصار قلب الـ OS) ، وبعض أنظمة التشغيل تجعل الـ Interpreter كبرنامج.

③ Touchscreen Interfaces

⇐ بسبب ظهور الأجهزة الخلوية اي عندها شاشة لمس، صار في شي اسمه Touchscreen Interface، ويعتمد على عدة تقنيات:

① الإشارات واللمس
② الكيبورد الافتراضي
③ الأوامر الصوتية

* System Calls

System calls provide an interface to the services made available by an operating system

User mode & Kernal mode are two modes in which a program can be executed

⇐ إذا البرنامج تم تنفيذه بالـ User mode يكون البرنامج ما إله وصول مباشر على الذاكرة والهارد ديسك وباقي الريسورس، بس إذا كان منفذ عن طريق الـ Kernal mode يصير إله وصول على الذاكرة وباقي الموارد

Kernal mode ⇒ Privileged mode (Having direct access to many of the resources)

⇐ المشكلة بالكيرنال مود إنه إذا صار للبرنامج crash أثناء التنفيذ ما بيتم كلّه بوقف عن الشغل، بينما بالـ User mode إذا صار شي للبرنامج بيتم ما بيصير شي وقف

User mode ⇒ Safer mode

- Most of programs execute mostly in user mode.

⇐ بما إنه أغلب البرامج بتنفذ بالـ User mode فهي بحاجة وصول للموارد فهي بتستدعي الـ OS، بتطلب منه ريسورس، فبتم تحويل البرنامج من الـ User mode إلى الـ Kernal mode

- systems call: call that made by the programs when it needs to access certain resources.
- When a program executing in user mode need to be switched to kernal mode for a particular time this system call is made.

- System call: The programmatic way in which a computer program requests a service from the kernal of the operating system.

- Generally written in C and C++.

⇐ امثال اي بالعمليات: قراءة ملف ونسخه على ملف ثاني جديد

- Acquire Input Filename — يطلب اسم الملف الأول.
- Write prompt to screen — يبدأ اليوزر عند اسم الملف
- Accept Input — يقبل اسم الملف المُدخل

- Acquire Output FileName — نفس الخطوات اي فوق بس للملف الثاني.

⋮

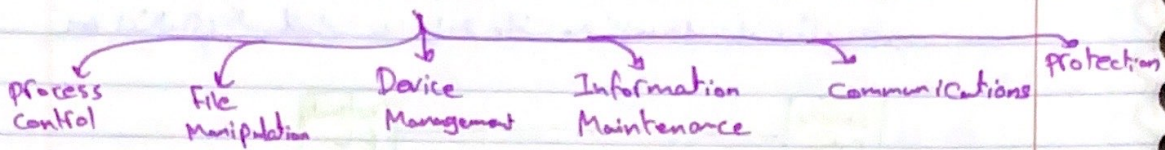
- Open input File — يفتح الملف الأول (بده يجيبه من الهيرد)
- If File doesn't exist, ABORT — إذا الملف مش موجود يطلع من البرنامج

- create output file — ينشيء الملف الجديد
- If file exists, ABORT — إذا اسم الملف موجود لازم يطلع من البرنامج

loop
- Read from input file — قراءة من الملف الأول
- write to output file — نسخ على الملف الثاني لحد ما يخلص الملف الأول

- close outputfile — يسكّر الملف
- write completion message to screen — يطبع على الشاشة إنه خلص
- Terminate normally — ينهي البرنامج بشكل طبيعي

* Types of System Calls



① Process Control

- end, abort (إنهاء البرنامج بشكل طبيعي أو غير طبيعي)
  - Normal
  - Error
- load, execute (يتم تحميل العملية ثم تنفيذها)
- create process, terminate process
- Get process attributes, set process attributes
- Wait for time (في عمليات تحتاج وقت معين)
- wait event, signal event
- allocate and free memory (بعد ما البرنامج ينفذ العملية البرنامج بيطلب مساحة، وبعد ما يخلص بيفضي هاي المساحة)

② File Manipulation

- create file, delete file
- open, close
- read, write, reposition
- get file attributes, set file attributes

③ Device Management

- request device, release device
- read, write, reposition
- get and set device attributes
- logically attach or detach devices (بالربط الجهاز بشبكة أو بشيء OS بفهم إنها عليها جهاز)

④ Information Maintenance (المعلومات اللي عنا في السيستم تاعنا)

- get and set time or date
- get and set system data
- get and set process, file, and device attributes

⑤ Communications

- create, delete communication connection
- send, receive messages
- transfer status information
- attach or detach remote devices

⑥ Protection

- Control access to resources
- Get and set permissions
- Allow or deny user access

* System programs

- System programs provide a convenient environment for program development and execution
- Some of them are simple UI to system calls

⇒ System programs be divided into the following categories:

① File Management

- Create
- Delete
- Copy
- Rename
- Print
- Dumb
- List and manipulate files directories

② Status Information

Ask the system for ⇐ جميع البرامج اللي بتعامل مع حالة النظام بتكون من هاد الكاتجوري

① Date, Time
② Amount of available memory or disk place
③ Number of users
④ Detailed performance
⑤ logging and debugging, etc.

③ File modification

⇐ الفرق بينه الفايل متجنت و الفايل مودفيكيشن إنه الأول بتعامل مع الطبقة الخارجية للفايل (بدون ما يطلبالي جواه) أما الفايل مودفيكيشن بنطلع تعديل المحتوى الداخلي للملف.

- It can be done by several text editors
- There a special commands to search contents of files or perform transformations of the text.

④ Programming language support

- Compilers
- Assemblers
- Debuggers
- Interpreters

⇐ System program توفر بيئة مناسبة لإنشاء البرامج
المكتوبة بلغات مختلفة

⑤ Program loading and execution

- Absolute loaders
- Relocatable loaders
- Linkage editors
- overlay loaders
- Debugging

⇐ بعد ما البرنامج يتعمله compile لازم يحمل على الجهاز
عشان ينفذ

⑥ communications

- System programs help incommunications between processes or users or allow user to surf the web
- Allow user to send messeges to another's screen
- Browse web pages
- send e-mails
- To log in remotly
- Transfer files

In addition to systems programs, most operating systems are supplied with programs that are useful in solving common problems or performing common operations (web Browsers, word processors, spreadsheets, Database syst...) Application programs

④ Microkernals

⇐ في هذا التركشر بدلاً ما يكون الكيرنال فيه كتير فنكشنليتز، بكون عنا كيرنال صغير بنزل فيه بس الأشياء الضرورية والأساسية، وبكون موجودة برة بعده باقي الأشياء الثانية على شكل user level و system programs، الوظيفة الأساسية للكيرنال هيك هو إنه يوفّر اتصال بين صفوف البرنامج وهاي الخدمات.

• فائدة هذا التركشر إنه أغلب الفنكشنز الموجودين بتتم تنفيذهم بال User mode وبس الكيرنال عنده Kernal mode فبكون الاستخدام قليل جداً.

• السلبيات: الأداء بضعف كلما زادت الأشياء والخدمات اللي فوق.

⑤ Modules

⇐ أفضل طريقة وبتتم استخدام الـ OOP عشان يتم إنشاءها.

- Core Kernal: Only have the core functionalities of the Kernal
- The other functionalities are present in the form of modules which will be loaded to the Kernal either at boot time or run time.

⇐ بكون موجود الـ Core Kernal وجواتيه الخدمات الثانية، إذا لزم شي منهم بتم تحميله لـ Core Kernal واستخدامه هناك

⇐ شابهة بين الـ Micro Kernal و الـ layered

- It resembles the layered system in such a way that each Kernal section has defined protected interfaces which is protected from the things that we don't want them to access but it's more flexible than a layered system in that any module can call any other module through the core Kernal
- Here we have only the core Kernal with only the core functionalities and then the other functions are loaded into it whenever necessary but its advantage as compared to the microkernal approach that a microkernal approach we need to have message passing in order to communicate between the modules because they are implemented as system or user level programs above the Kernal but in this one they are loaded dynamically directly into the core Kernal asswhen needed so they don't need that message passing and hence the system overhead is not there

* Hybrid systems

* أغلب أنظمة التشغيل مش نموذج واحد، الفكرة من البايرد سيستم إنه بيشيل حسب approach عشان يحقق أداء عالي وأمنه عالي ويكون الجهاز قابل للاستخدام بنفس الوقت

- Linux & Solaris ⇒ Monolithic + Modular
- Windows ⇒ Monolithic + Microkernal
- Apple ⇒ Hybrid + layered

* IOS

- Runs on different CPU architecture.
- Cocao Touch ⇒ developing Apps
- Media services ⇒ Graphic, Audio and video
- Core services ⇒ cloud computing, databases
- Core OS ⇒ Based on Mac OS X kernal

* Android

⇒ Open source
⇒ Based on Linux kernal
⇒ Apps developed in java plus Android API

* Operating System debugging

Debugging: Finding and Fixing errors or bugs.

ع نظام التشغيل هو اي بعمل ملفات بتحتوي على معلومات الأخطاء

* Operating System Generation & System Boot

⇐ هنا بنوصف عمل الـ OS، النوع الأول بنصمم وبنبني جهاز معين، والنوع الثاني اي بشتغل على عدة أجهزة مختلفة، وهاد هو النوع اللي بنتعامل معه أكثر وما بغلب.

⇐ السؤال اللي بنطرحه هنا إنه إذا الأجهزة بتختلف عن بعضها في الهاردوير كيف رح نثبت النظام بشتغل على أجهزة مختلفة؟

- If the machines are having different hardware configurations then how will the operating system be installed in these different systems if their configurations are different??

Answer: There is something on a system generation helps in generating the operating system in such away it will be compatible with the other machine

⇐ في برنامج اسمه (SysGen) بكون موجود بنظام التشغيل فبليه بقدر يشتغل على الجهاز اللي بدنا نثبته عليه

* The following kinds of info. must be ~~used~~ determined by the SYSGEN program?
  - what CPU to be used?
  - How much memory is available?
  - what devices are available?
  - what operating system options are desired?

* System Boot

- Booting the system: The procedure of starting a computer by loading the kernal.
- On most ~~computer~~ computer systems a small piece of code known as the bootstrap program or bootstrap loader locates the kernal
  ⇒ The function of it is to locate the kernal and help it to load it to the memory because the hardware itself doesn't know how to load it.

⇐ الـ Bootstrap بكون موجود على الـ ROM وكتبته هنا سبب إنه عدم حاجتنا لتنزيله وبتأثر بالفايروسات + الـ ROM ما بنسمح إنه أي شي يكتب عليها لو راحت الحماية

⇐ أو بكون مخزنة على الـ EPROM ⇐ هاي ROM بس إذا كتبنا أوامر معينة بنقدر نمسح اللي كتب عليها

Erasable Programmable ROM