

Operating Systems

John SUM
Institute of Technology Management
National Chung Hsing University
Taichung, ROC

October 30, 2019

Contents

1	Introduction	3
2	A Simple Problem	3
3	System Initialization	3
3.1	Instructions from BIOS	4
3.2	Instructions from memory	4
4	Functions of an OS	5
4.1	Program running	5
4.2	Memory management	6
4.2.1	Chopping	6
4.2.2	File allocation table	7
4.2.3	Trash	8
4.3	File management	9
4.3.1	File access	9
4.3.2	File system	10
4.4	Process management – Multitasking	12
4.4.1	Client-server operation	12
4.4.2	Multiple access	13
4.4.3	Watch-dog	14
4.5	Background processes	14

5	System Close	15
6	Machine dependent	15
6.1	From CPU dependent to hardware dependent	15
6.2	Emulator	16
7	Common Operating Systems	16
7.1	Personal computers	16
7.2	Smartphones	16
7.3	Local area networks	17
7.4	Series of OS	17
8	System Software & Application Software	18
8.1	System software	18
8.2	Application software	18
8.3	Application software development	19
9	More on Client-Server Operation	19
9.1	Credit card system	19
9.2	Transportation system	20
9.3	Other client-server systems	21

1 Introduction

Once you have understood that a computer is just a combination of various logic circuits and devices for input/output, what you need to know is the operating system. Operating system is a complicated program. It consists of many lines of machine code. Each line of machine code is actually a special pattern of electrical signals feeding to the CPU or other hardware devices for special purpose. Once a device has been fed with such signals, it performs a certain task in accordance with the instruction specified by the machine code.

Operating system starts running when a computer (respectively a smartphone) is power on and then stops when the power is off. Some of its purposes is to manage the communication amongst the hardware devices, manage the memory system for data storage and retrieval and manage the file system, the input/output devices. Once this infrastructure has been setup, a computer user can interact with the operating system via either command prompt or graphical user interface to use the computer.

2 A Simple Problem

To highlight the difficulties of the tasks to be performed by an operating system, let us have a simple example. It is a memory management problem, see Figure 1. The blank rectangle on top refers to a memory space with no file inside. After the computer has been used for a while, two files have been stored, the File X and File Y. As the initial memory space has nothing, the operating system simply put File X in the first portion and put File Y just after File X. After another while, it is supposed that File X has been deleted. Now, another application system would like to save File Z, the black rectangle in the bottom. The size of this file is larger than File X. **So, what the operating system should do to store File Z?**

3 System Initialization

Once the computer is on, the operating system starts to do the system initialization and preparing everything for the user to use.

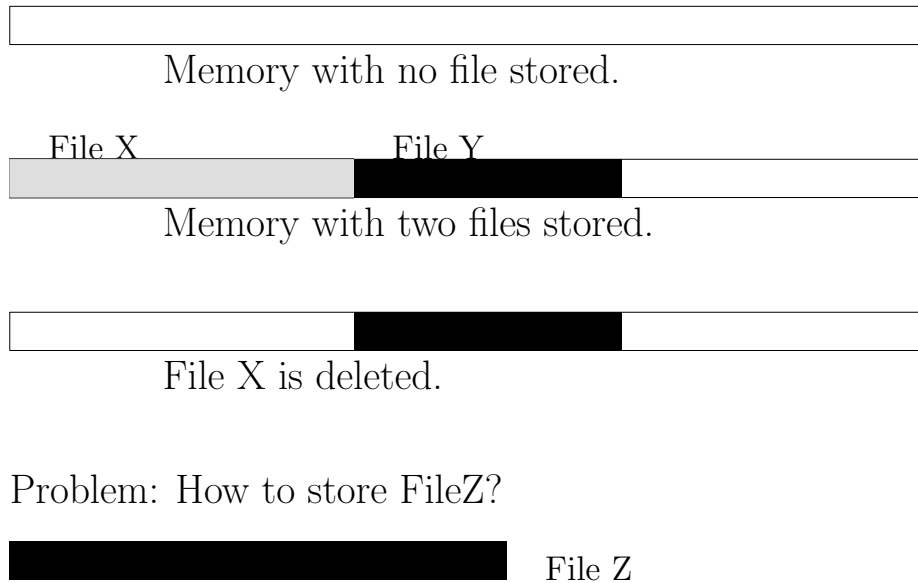


Figure 1: A problem in memory management.

3.1 Instructions from BIOS

While the majority of the operating system is stored in the secondary storage, the first few lines code are stored in a memory called BIOS. BIOS is a read only memory (ROM). The content in the BIOS has been fixed once it is manufactured. Once a computer has been on, the CPU will automatically fetch (search and get) the first instruction from this BIOS. For sure, this set up is also specially designed to handle this job. The instructions in BIOS instruct the CPU to start up all the necessary connections amongst the CPU, the RAM, the main memory and other necessary hardware components.

3.2 Instructions from memory

Once the connections have been set up, the CPU are ready to fetch from the main memory other instructions of the operating system for setting up the environment for users to access the computer. It includes running a background process that checks the keystrokes signal of the keyboard. It also includes running a background process checking the file system in the main memory for better file management. It also re-loads the desktop environment (the graphical display) that the user was accessing before last power off. For

some complicated environment, the operating system even checks out all the network connections and logging in the internet accounts that the user was logging on before last power off. Once everything is ready, the operating system will be in an idle status waiting for user commands.

4 Functions of an OS

In summary, an operating system is used for **(i) hardware management, (ii) memory management, (iii) file management, (iv) process management, (v) managing the user control via graphical user interface (like Window or command console)**. It acts as a middleman between the application software and the hardware.

4.1 Program running

Here is an example showing what the operating system does once after an executable program has been invoked.

- User moves the cursor to the icon of *Chrome*.
- User can then invoke the *Chrome* by double-click the left button of the mouse.
- The operating system checks from the executable file **chrome.exe** how much memory space to be allocated as the working space for this application software. Part of the space will be used for the machine codes and part of it is used for temporary data storage.
- If the space (part is from the main memory and part is from the RAM) is enough, the operating system will transfer the control to **chrome.exe**. If the window for the Chrome browser is still active, i.e. the process 'chrome' is active, the user can use *Chrome* to browse web pages. If the window for the Chrome browser is not in active mode, the operating system could put the process 'chrome' in waiting mode. In either cases, the working space and other related information will be kept.
- If the available space is not enough, the operating system could decide to move some inactive processes out of the waiting queue to sleep mode.

Every information related to these inactive processes will then be moved to the main memory. Their working space will be free for the 'chrome'¹. Once the space is ready, the operating system will transfer the control to **chrome.exe**. Again, if the window for the Chrome browser is still active, i.e. the process 'chrome' is active, the use can use *Chrome* to browse web pages. If the window for the Chrome browser is not in active mode, the operating system could put the process 'chrome' in waiting mode. In either cases, the working space and other related information will be kept.

- Once the browsing is finished and the Chrome has been closed, the operating system will then free the memory space as the one before the chrome.exe is invoked.
- Operating system waits for another call or continues to serve other process in the waiting queue.

The above steps are just for illustration. The actual procedure could be a lot of complicated. For instance, the operating system might need to maintain a temporary file system and working memory space for each process no matter it is in active, wait or sleep mode. While a process has been swapped to sleep mode, the content of the file system, information stored in the working space and other information regarding the status of the process have to be saved. These steps are particular important to the operating system in handling multiple tasks, i.e. multi-tasking.

4.2 Memory management

Before talking about how the operating system handles multiple tasks, let us go back to the simple memory management problem as shown in Figure 1.

4.2.1 Chopping

To save File Z, one simple approach is to chop the file into two segments. The first segment is saved in front of File Y and the second segment is save after File Y, as shown in Figure 2.

¹Clearly, how to decide if a waiting process is scheduled to sleeping mode is based on the design of the operating system.

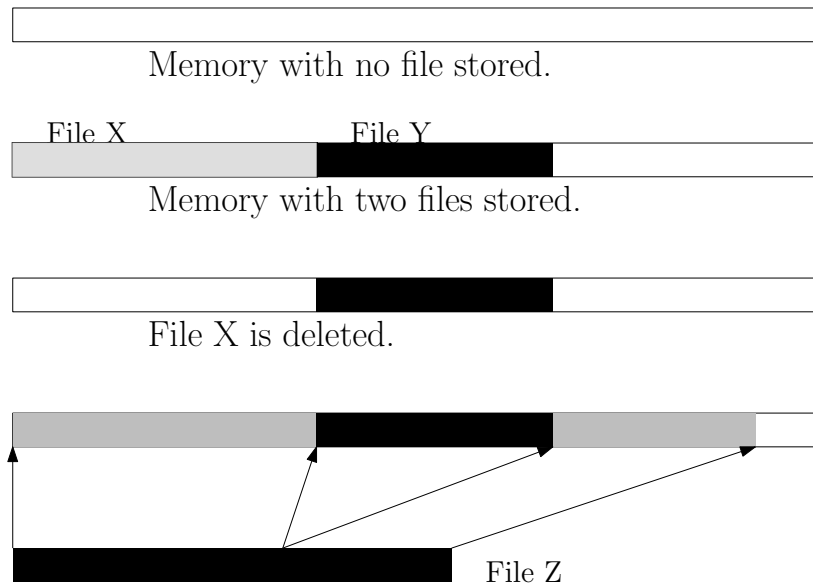


Figure 2: A simple approach to save File Z.

It should be noted that a file normally will consist of two important data to indicate the beginning of a file (let call it file-start) and the end of a file (let us call it file-end). As the file is not chopped into two segments, the operating system needs to add something at the end of the first segment to indicate that (i) the file has not been ended and (ii) the memory address of the next segment. In these regards, the total file size of the stored File Z is larger than the bared File Z.

4.2.2 File allocation table

Once a file has been stored in the memory, it will definitely be retrieved in the future. Thus, an operating system needs to maintain a table containing the locations of all these files. This table is called file allocation table. For the example as shown in Figure 2, the table is initially empty. Once the file File X and File Y have been stored, the locations (i.e. their memory addresses) are updated in the table.

Filename	Memory Address
File X	0
File Y	9

After File X has been deleted, the operating system simply removes the record of File X from the table.

Filename	Memory Address
File Y	9

It should be noted that the actual content of File X is still in the memory. The actual content has yet been erased. In this point of time, File X is still recoverable. The file allocation table is yet another file. It is also stored in the memory. If we can find out the memory location of this table, we can direct access this memory location and change its content.

Once File Z has to be stored, the operating system checks that there is no consecutive memory space for File Z. So, the operating system will have to chop the file into two segments and add overhead to them. Finally, the operating system updates the file allocation table simply by adding starting address of File Z.

Filename	Memory Address
File Y	9
File Z	0

Now, File X can never be recovered as its previous location has been overwritten by File Z.

4.2.3 Trash

In the old day, the operating system will not show you the files that have been deleted. Today, as in Microsoft Windows, deleted files will be put in a trash (or recycle bin). The purpose is to let the user recover accidentally deleted file. To accomplish this, an additional file allocation table for the trash is maintained. For the above example, After File X has been deleted but File Z has not been stored, the contents of the tables are depicted as below.

Normal		Trash	
Filename	Memory Address	Filename	Memory Address
File Y	9	File X	0

As File X is in the trash, the operating system response with an error message 'not enough memory space' while it is requested to save File Z. In the end, the user has to decide if File X has to be deleted from the trash to release memory capacity for File Z.

4.3 File management

In a computer, there are many different types of files. Some of them are created by users. Document files and image files are two typical examples. Some of them are created by operating system. Some of these system files are created temporary for maintaining the statuses of the processes running in a computer. So, the basic operations in file management are (1) to create and delete a file, (2) to open and close a file, (3) to read, write or append a file, (4) to search a file from the main memory and (5) file system management.

4.3.1 File access

Suppose that you have selected 'New File' in MS Word and typed in certain content. After editing, you select the option 'Save'. The MS Word will ask for the file name and the folder where you would like to save the file. Once you have keyed in the information, the operating system will perform the action 'Create'. The file is thus saved in main memory (via the memory management process). If you have not yet exit the MS Word, the file will thus be assigned with statuses 'Open' and 'R/W/A'. Until you have finished editing and exited. The file will be set to 'Close'.

It is strict rule in any operating system on file access. A file can only be read, write or append if it has been open. If a file is in 'Close' state, nothing can be done on it.

For those files which are created by users, the files are usually managed by the users in which folders they should be located. Sometimes, users might need to create a tree of folders for better managing the files. The creation and deletion of files is thus managed by users. For system files, users have no right to access. All operations acting on these system files are managed entirely by the operating systems. That is to say, only the operating system has the right to access the files.

For a computer which is designed for multiple users, the operating system allows the users to set the access right for a file or a folder. Unix operating system for instance, the user can set the access right of file (or a folder) to be (i) owner level R/W/D and (ii) group level R/W/D and (iii) world level R/W/D. If a user is granted to write a file, it implies that the user can append content to a file.

Owner refers to the user who creates the file. Group refers to a specific group of users that the file owner assigns. World refers to everyone in the

world. By default, Unix will set the access right of each file or folder as follows :

Level	Access right
System	Read, write, delete
Owner (File creator)	Read, write, delete
Group (Dedicated users of the computer)	Read
World (Every user of the computer)	None

In all systems, the system account has the highest privilege. It can do anything in a computer. Therefore, the system account has the right to read, write and delete any file of any user.

Two points should be noted for the case that multiple computers are connected as a network. First, there is no super-system account. Still, each computer will have its own operating system and its own system account. Each system account can only be allowed to access the files in its own computer. Usually, the system account of one computer is not allowed to access the files of another computer. Second, world access is granted to all the users in the network. If a file is set to be 'world readable', this file can be read by any user in the network. If the computer is connected to the Internet, this file will be readable for everyone whose computer is connected to the Internet.

An obvious example is the webpages of a website. While we would like to browse the NCHU webpage, we could simply open the browser and key in the URL – <http://www.nchu.edu.tw>. Then, the browser will connect to the web server of the domain 'nchu.edu.tw', download the file 'index.html' and read it. Why our browser is able to read this file is because the file 'index.html' has been set to 'world readable'. If the file has not been set to this condition, the webpage is not accessible.

4.3.2 File system

One more function of the operating system in file management is to maintain the file system information. File system refers to the directories (i.e. folders) where the files are put under and which files are put under such directories. Figure 3 shows a simple example. Here, 'c:\>' refers to the root directory. The root directory has two sub-directories (or simply directories), 'Program Files' and 'Windows'. Under the directory 'Program Files', there are two

```
c:\Program Files\notepad\notepad.exe
    \office\msword.exe
        \msexcel.exe
        \mseaccess.exe
    \explorer.exe
    \draw.exe
c:\Windows\user\johnsum\computer_science.doc
    \final_exam_2016.pdf
    \final_exam_2017.pdf
    \final_exam_2018.pdf
```

Figure 3: An exemplar file system.

directories, namely 'notepad' and 'office'. Under the 'Windows' directory, there is a 'user' directory and then the 'johnsum' directory.

This tree of directories is the file system structure in which the operating system has to maintain. Clearly, users have also been involved in maintaining this file system. The operating system will have to update the status of each file, including its creator, its file type, its modification date and its file size.

To better manage the files, especially the system files, each operating system will create directories (i.e. default directories) for system files once the operating system is first installed. Different operating systems have different default setting on these default directories. For instance, in Unix system, the directory name for the system programs is called 'bin', instead of 'Program Files'. Figure 4 shows the default directories of Windows system. The sub-directories 'Program Files', 'Program Files(x86)', 'Programs' and 'Windows' are prepared for the system files. These directories are pre-set as read only directories. Normally, user is not allowed to delete any file in these directories. Whenever a new system is going to be downloaded and installed, the operating system will by default store the system under the directory 'Program Files', unless the user has specified other directory for the system. As a common practise, users should not store any user file in these directories.

Apart from the visible file system which can be browsed by Microsoft Explorer, there is another invisible-to-user file system – the file system for system files (the files for handling the processes running in the computer). This special file system is maintained entirely by the operating system.

File management and memory management work together to speedup the

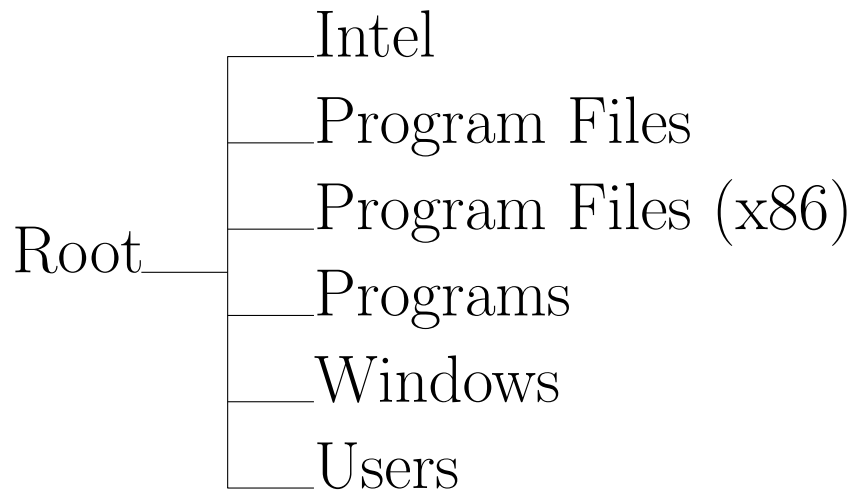


Figure 4: The root directory of Windows.

file search and the retrieval of a file from the memory.

4.4 Process management – Multitasking

It should point out that the operating system handles more than one process at a time. This is called multi-tasking. Therefore, an operating system always needs to frequently update the status of many tasks and their related information. The CPU needs to swap from one process to another. Even when you are interacting with IE for Internet search, the CPU could be instructed to do something else which are not for iexplorer.exe.

4.4.1 Client-server operation

To handle multiple tasks, one simple idea is illustrated in Figure 5. Once an application system has a request to the operating system, it will put a 'service request' to the service queue of the operating system. Based upon the design, it could have multiple service queues to indicating different priorities. Without loss of generality, it is assumed that there is only one service queue here. Once the request has been put in the queue, the application system will only have to wait and check if there is any message appeared in its service queue.

The operating system from time to time will check its service queue and

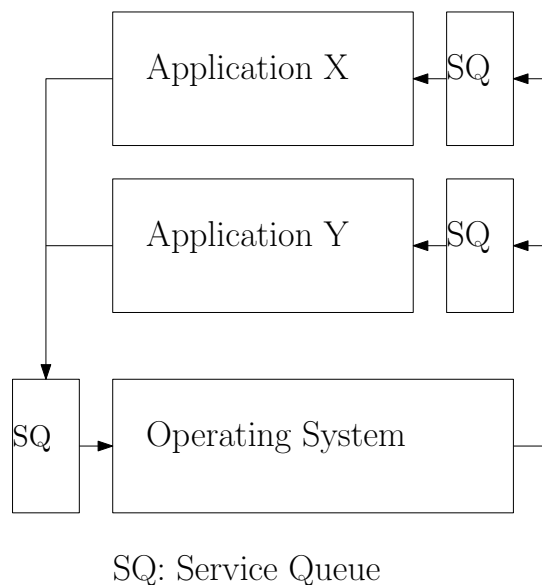


Figure 5: Use of service queue to handle multiple tasks.

complete the requests one by one. After it has completed one request, the results will thus be send to the service queue of the application system which submitted the request. Then, the operating system checks its queue and completes another request.

From that, it is easy to see that an operating system is just like a server. The application systems are clients. Once the computer has been power on, the operating system stands by and waits for the requests from the application systems. Once a request has been arrived, it serves the client to complete the tasks specified in the request and responses by sending the results back to the client. This mode of operation is called client-server mode.

As a matter of fact, the concept of client-server operation (or client-server computing) is not new and not limited to the field of computer science. We have encountered this mode of operation almost in everyday. We will come back this issue later in the text.

4.4.2 Multiple access

As long as multiple application systems are running at the same period of time, it could happen that two applications are accessing the same file at the time.

- A PDF file is open to be read by EDGE while it is being read by an Acrobat Reader.
- The user would like to change the filename of a PDF file while it is being read by an Acrobat Reader.
- A file is open to be edited by NOTEPAD while it is being edited by WinEDIT.

For the first case, both applications are reading the files. Normally, the operating system will allow to do so. For the second case, Windows will not allow to do so. An error message will be shown on the screen. For the third case, it is allowed in the Windows environment. However, it is not advisable to do so. Why?

An operating system has not just to access users' files, but also has to access many 'system files'. If any one of these system files has been miss-updated, the operating system could thus be malfunction. For the worst case, user needs to re-install the operating system. How to solve this problem, it is up to the design of the operating system.

4.4.3 Watch-dog

Now, one problem the operating system has to solve is to know when a service request has arrived. To solve this problem, there is no any intelligent method. One method is to use a watch-dog. The operating system simply checks the service request queue from time to time, in mini-second basis for instance. Once the operating system has found that a service request has arrived, it checks its content and complete the task as requested.

4.5 Background processes

Watch-dog program is just one of many background processes. If you press 'CTRL-ALT-DEL', you are able to invoke the 'Task Manager'. In it, you will find a lot of background processes that are running in your computer. Some of these processes are initiated by the operating systems and some of them are initiated by application systems.

5 System Close

Once the power is going to be off (i.e. normal power off), the operating system will check the statuses of all the processes which are in active, wait or sleep mode. Their corresponding information will thus be saved and then the power is off. If the system is shut down abnormally, these information will likely be lost.

6 Machine dependent

It should be noted that the lines of code in the BIOS and the program instructions of an operating system are **machine dependent**. The 'machine' embraces all the hardware devices in the computer. They include the CPU, the memory devices, the RAM, the keyboard, the panel, the mouse, the network card and etc. Today, many of these devices have common standards. While keyboards could be manufactured by different manufacturers, their standards for communication between the keyboard and the mother board are the same. In the past few decades, computer industry did put a lot of effort to unify those standards. Therefore, the term 'machine dependent' could not be interpreted as 'processor dependent' if we are talking about computer operating systems.

6.1 From CPU dependent to hardware dependent

In 1990s, most computers are equipped with Intel CPU. So, the operating system installed in these computers was usually Microsoft Windows. At the same time, Apple used Motorola processor as the CPU for Macintosh and thus MacOS was developed. Something changed in the middle of 1990s. As Motorola shut down her processor manufacturing, Apple had to use Intel processor as the CPU of her computers. MacOS was thus re-designed owing to fit for Intel processors. So, it comes to an interesting phenomena – one machine two operating systems. Computer is no more a good example to describe the term 'machine dependent'. Luckily, in the smartphone industry, we still can see this 'machine dependent'.

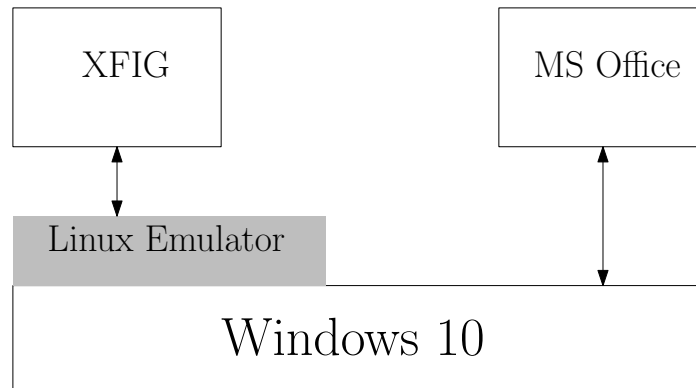


Figure 6: The Linux emulator for XFig.

6.2 Emulator

Occasionally, user of Windows system might need to use an application system, say XFig, that are running on Linux system. In such case, user could install a Linux emulator. Then, the system XFig can be installed. Emulator is a program running on the Windows operating system. It acts as an interface for the XFig. While XFig is running, it seems like interacting with the Linux operating system, Figure 6.

7 Common Operating Systems

7.1 Personal computers

For a computer with Pentium CPU, Microsoft Windows OS (developed by Microsoft) and Linux (developed by Linus Torvalds and first released 5 October 1991) are two common operating systems. For Apple Macintosh, the early operating system is called MAC OS, and the latest version is called OS X.

7.2 Smartphones

For smart phones, like Google Phone and HTC One, the operating system is called Andriod which is developed by Google. It is built upon the Linux. For Apple tablet PCs and iPhone, the operating system is iOS in which the design is based on OS X. Clearly, iOS is developed by Apple. For Nokia

Windows phones, the Lumia 800 and Lumia 710, the operating system is called Windows Phone which is developed by Microsoft. Its latest release is Windows 8 Mobile. This OS is compatible with Windows operating systems.

7.3 Local area networks

Network Operating System is the software for managing the resources (like file server access, database server access, email server access, Internet connection and etc) sharing amongst users on a local area network. The most popular network operating systems are Microsoft Windows Server, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

7.4 Series of OS

From the above, it should be clear that the operating systems could be categorized in four series.

1. Unix/Linux: Linux (for computer) and android (for smartphone).
2. MacOS: MacOS, OS X (for Apple computers) and iOS (for iPhone).
3. Windows: Windows (for computers) and Windows Mobile (for Lumia).
4. Harmony: Harmony OS (for both computers and smartphone).

There is one important reason to motivate the development of computer OS and smartphone OS under the same series. It is for compatibility. Today, users of computers and smartphones always need to transfer data, either via USB cable or WiFi, in between two devices. If both devices are running the operating systems of the same series, users would have no need to worry about the incompatibility problem.

In the future, the smartphone will not just be connected to computer, but it will also be connected to home devices to form a wireless home network. If the operating systems running in these devices are under the same series, communication problems amongst such devices should be minimized.

8 System Software & Application Software

An operating system is the key of a computer. It manages all the resources in a computer and provides services to the application systems. Operating system is a special type of software called **system software**.

8.1 System software

There are other system software, normally called servers, which are particular designed to support users over the network. Web server (Apache, Microsoft IIS and NginX), database server (MySQL and SQL Server), email server and FTP server are four common examples. These system software are designed to support multiple users (equivalently, multiple clients). Their operating principles are the same as the operating system in handling multi-tasks. Each client (respectively task) sends request to the server (respectively operating system) asking for completing certain tasks. Once the server has completed the tasks, it sends back the results in a form of message to the client via the Internet.

One kind of system software is called **device driver** which is used for controlling the I/O devices, like mouse, keyboard, USB memory, DVD, digital camera and printer. Today, many device drivers have already been installed in a computer. Some of them have even been included in the operating system. Thus, we normally have no need to install such drivers by ourselves. However, there are still some conditions that we need to download and install them. For example, after purchasing a new HP laser printer, the computer user will have to download the device driver of this laser printer from HP website and install it in the computer. Otherwise, the laser printer will be useless.

8.2 Application software

In contrast to system software, **application software** is built on top of the these system software (including operating system, network operating system, device drivers and other servers). Application software is used for user to use in daily life. MS Office, Skype and XFig (a software for drawing diagrams) are common PC application software. The so-called APPs that many smartphone users downloaded Apple APP Stores or Google Play are application software. Some application software are designed particular

to support the business operations within an enterprise. They include supply chain management systems, customer relationship management systems, course registration systems and etc.

8.3 Application software development

To develop a simple application software, such as a sorter and a calculator, programming language like C language is already good enough. For some complicated programs, such as browser and media player, programmer would need to know the function calls provided by the operating system. Together with C programming, such an application software could be developed. For programs involving network communication, programmer would need to know much more.

9 More on Client-Server Operation

As mentioned early, the concept of client-server operation is not new and not limited to the field of computer science or electrical engineering. In simple words, a server is simply a system providing services (function calls) for its client. Special system, like operating system, provides many function calls (more than your imagination) for the application systems to solve complex problems. But, there are simple systems which provide simple services for their clients. In fact, almost every one of us has nowadays been involved in some client-server operations. Here, let us give a few examples.

9.1 Credit card system

Every time you pay by credit card for a gift purchased in a department store, you have already initiated a client-server operation. The ultimate goal (equivalently the high level purpose) of pay by credit card is to let the department store receiving the money from you. In operations level, the detail steps could be described in the following. Here, we let Bank A be the bank issuing your credit card and Bank B be the bank handing the cash flow of the department store. Moreover, we assume that the amount to be paid for the gift is NTD5000. All information communicated in the process are correct. If any one of these information is wrong, the transfer will be aborted.

Step 1: You need to key in the device your credit card password and the amount NTD5000 to be transferred.

Step 2: The information will then be sent to Bank A (your bank) for verification.

Step 3: Once the bank has received such request, (i) it verifies the password, (ii) checks the balance of your bank account and (iii) checks the validity of Bank B.

Step 4: If everything is alright, Bank A sends a bank-transfer message to Bank B.

Step 5: Bank B grants the bank-transfer.

Step 6: NTD5000 is then transferred from your account in Bank A to the account of the department store in Bank B.

Step 7: Both Bank A and Bank B update the balances of the accounts.

Step 8: Bank A sends message to the device indicating the completion of bank-transfer.

In the above procedure, the card holder has only been involved in **Step 1** and **Step 8**. From **Step 2** to **Step 7**, the process is conducted by the interaction between Bank A and Bank B. Precisely, the process is conducted by the banks in the *credit card system*. In this regard, the card holders are the clients of the credit card system which serves its clients in money transfer.

9.2 Transportation system

While you would like to go from NCHU to NTU, one approach is to take bus and train.

Step 1: Take a bus at the main gate of NCHU to the Taichung train station.

Step 2: Take a train from Taichung train station to Taipei train station.

Step 3: At the Taipei train station, get a taxi to the main gate of NTU.

In this example, the transportation system acts like a server. The passengers are clients. The system serves many clients at a time.

9.3 Other client-server systems

There are many other systems acting like a server serving their clients everyday. Some of these systems provide complex services. Some of them provides simple services.

Services	Server	Clients
Postal service	Post office	Citizens and firms
Logistics	DHL	Citizens and firms
Customer acquisition	SOGO	Merchants
Passenger acquisition	Uber	Drivers
Search	Google	Searchers
Data communication	Internet	Computers
Design and fabrication	TSMC	Apple, Qualcomm, etc.