Assignment no.2

Camlin Page
Date 10/10/22

Q.1) What are the services provided by OS?

→ • Operating system provide an environment for execution of programs and service to program & users.
• One set of operating system services provides functions that are helpful to user:
• User Interface:- Almost all operating system have a user interface. It varies between Command-Line Interface (CLI) & Graphical user Interface (GUI).
• Program execution:- The system must be able to load a program into memory & to run program, end execution, either normally or abnormally.
• I/O operation:- A running program may require I/O, which may include a files or an I/O device.
• File-system manipulation:- The file system is a particular interest, program need to read and write and directories create & delete them, search them, permission management.
• Communication:- Processes may exchange information on the same computer or between computer over a network.
• Error detection:- OS needs to be constantly aware of possible errors:-
- may occur in CPU and memory, hardware in I/O devices, in user program.
- For each type of error, OS should take

appropriate action to ensure correct and consistent computing.

- Resource allocation:- When multiple user or multiple jobs running concurrently, resources must be allocated to each of them.
- Accounting:- To keep track of which user use how much and what kinds of computer resources.

Q.2) Explain :-

1) Graphical User Interface :-

- User friendly desktop metaphor interface.
- Usually mouse, keyboard and monitor.
- Icons represents files, program action, etc.
- Various mouse buttons over objects in the interface cause various actions.
- Invented at Xerox PARC.
- many system now includes both CLI & GUI interface.
- microsoft Windows is GUI with CLI command shell.
- Apple mac OS is Aqua GUI interface with UNIX kernel underneath and shell available.

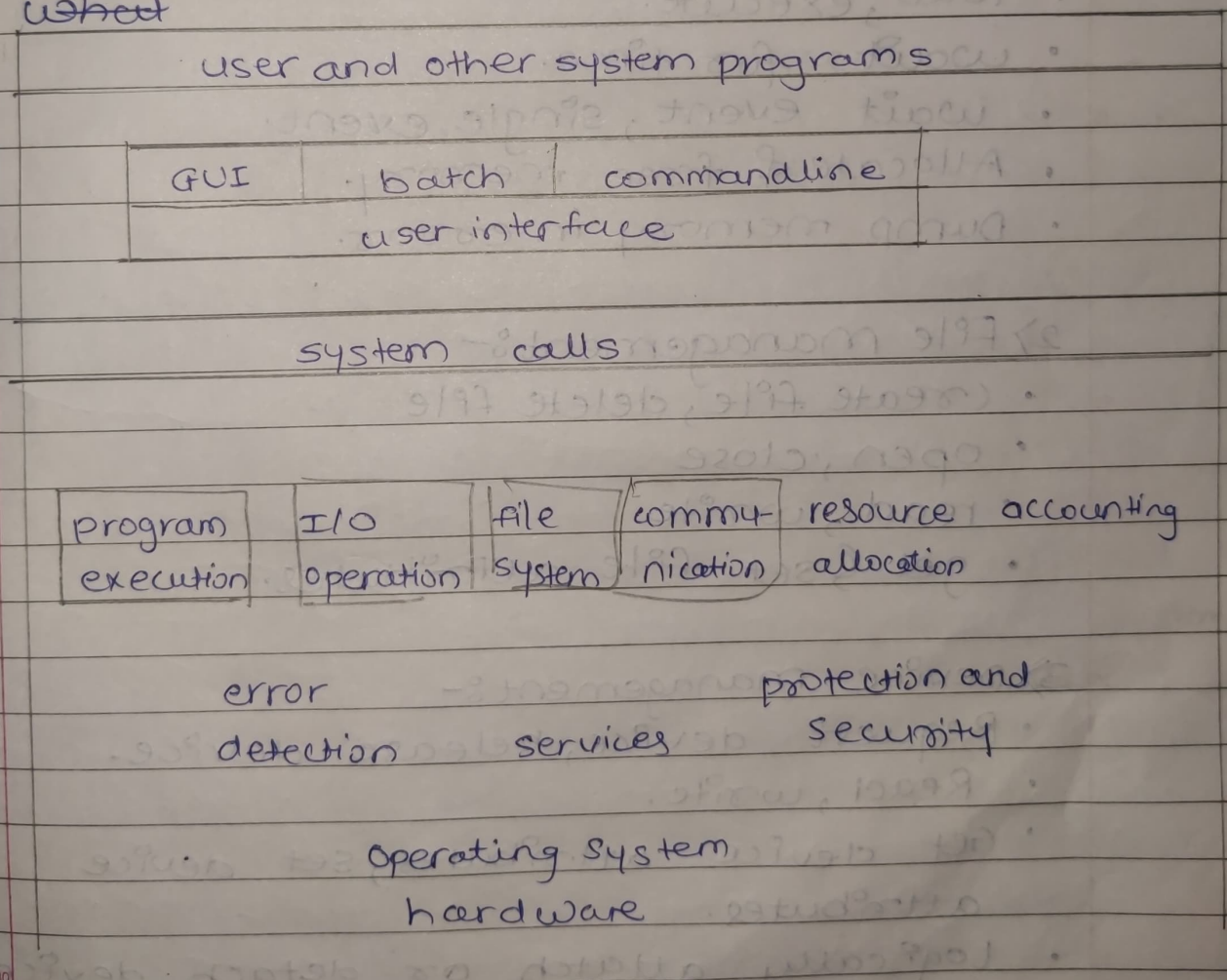
2) Command Line Interface :-

- Sometimes implemented in Kernel, sometimes by ~~Operating~~ System. programs.
- Primarily fetches a command from user

and executes it.
. Sometimes commands built-in, sometimes just names of program.

3) Touchscreen Interface:-
. Touchscreen device require new interface.
. mouse not possible or not desired.
. Action and selection based on gestures.
. virtual keyboard for text entry.
. voice commands.

~~Q.3)~~ ~~What~~



Q.3) What is system call? Explain types of system calls.

→ • System call is the programic way in which a computer program requests a service from the kernel of operating system on which it is executed.

• Types of System calls:-

1) Program Control:-
- creating a program, terminate program.
- End, Abort.
- load, execute.
- wait for time.
- wait event, single event.
- Allocate & free memory.
- Dump memory in error.

2) File management:-
- create file, delete file
- open, close
- read, write
- get & set file attributes.

3) Device management:-
- Request device, Release device.
- Read, write.
- Get device attributes, set device attributes.
- Logically attach or detach device.

4) Information maintainance:-
- Get & set date & time.
- Get & set system data.
- Get & set process, file & device.

5) Communication:-
- Create, delete communication connection.
- Send, Receive.
- transport status information.
- Attach & ~~det~~ detach remote device.

6) Protection:-
- Control access to resource.
- get & set permission.
- Allow & deny user access.

Q.4) Explain system program and types of system program.

→ • System program provides convenient environment for program development and execution.
- most users view of system is defined by system program not the actual system call.
- Provides a convenient environment development and execution.
- Some of them are simply user interface system.

Types of system program:-
1) file manipulation:- create, delete, copy, rename, print, dump, list and generally

manipulating files and directories.

2) Status Information:-
some ask the system for info, date, time, memory, disk-space, other provided details information about logging and debugging.

3) File modification:-
- Text editors to create and modify files.
- Special command to search content of files or perform transformation of ~~pro~~ text.

4) Programing language support:-
Compilers, assemblers, debuggers.

5) Program loading & information:-
Absolute loaders, relocated loaders.

6) Communication:-
provide the mechanism for creating virtual connection among processes, users and computer system.

Q.5) Explain OS structure.

→ - General purpose OS is very large program.
- various ways to structures ones.

1) Simple structure --MS-DOS:-
- MS-DOS- written to provide most functinality in least space.

Date / /

- Not divided into modules.
- Its interface and level of functionality are not well seperated.

2) More Complex UNIX :-
UNIX limited by hardware functionality the original UNIX operating system had limited structuring. Have two separable parts :-
- System programs.
- The Kernel :-
Consists of everything below system call interface and above physical hardware.
Provides file system, CPU scheduling, memory management, and other operating system functions; a large number of functions for one level.

3) Layered Approach :-
- The operating system is divided into no. of layers, each builts on top of lower layers & the bottom layer is hardware. The highest (layer n) is user interface.
- With modularity, layers are selected such that each uses functions and services of only lower-level layers.

Date / /

4) microkernel :-

- moves os much from kernel into user space.
- mach example of microkernel
- Communication takes place between user modules using message passing.
- Benefits :-
  - Easier to extend a microkernel.
  - Easier to port operating system to new architecture.
  - more reliable, more secure.
- Detriments :-
  - Performance overhead of user space to kernel space communication.

Q.6) How OS can be designed and implement?

→ • Design and Implementation of OS not 'solvable', but some approaches have proven successful.
• Internal structure of different OS's can vary widely.
• start the design by defining goals and specifications.
• OS should be convenient to use, easy to learn, reliable, safe and fast.
• OS should be easy to design, implement, and maintain.

• Implementation :-
  • much variation :
    Early OSes in assembly language.
    Then system programming languages like: Algo.
    Now C, C++.
  • Actually usually a mix of language.
    Lowest levels in assembly.
    main body in C.
    system programming in C, C++.
  • More high-level language easier to port to other hardware but slower.
  • Emulation can allow an OS to run on non-native hardware.

Q.7) Define:

1) Debugging:-
Debugging is the process of finding and fixing the error, in OS.. Both hardware and software problem can be fixed with the help of debugging.

2) log file:-
A log file is computer generated data file that contain information about usaage patterns, activities, and OS applicat. with an OS and is the primary data source from network observability..

3) Core Dump:-
A core dump is a file that gets automatically generated by the Linux Kernel after program crashes. This file contains memory register values and call stack of an application at the point of crashing.

4) Crash Dump:-
A crash dump also known as bug check occurs when window can run correctly. The dump file that is produced from this event is called system crash dump.

Camlin Page
Date / /

⇒ Performance Tunning:-
Performance Tunning improves price to performance ratio for a system or set of services by reallocating the variable computing network or storage resources.

Q.8) Explain System Boot program.
→ • When power initialized on system, execution starts at fixed memory location.
  • Firmware ROM used to hold initial boot code.
• Operating system must be made available to hardware so hardware can start it.
  • Small piece of code - bootstrap loaders, ~~and~~ stored in ROM or EEPROM locates kernel, loades it into memory, starts it.
  • Sometimes two-step processes, where boot block at fixed location loaded by ROM code, which loads bootstrap loader from disk.
• Common bootstrap loaders, GRUB, allows selection of kernel from multiple disks, versions, kernel options.
• Kernel loads and system is then running.