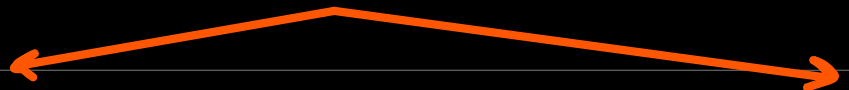
System Software

- A software platform that provides facilities for programs to be run which are benefit to a user

Q- Why operating system should be used?

- Hardware is unusable without an operating system
- Acts as an interface and controls communication between user and hardware
- Provides software platform.

User Interface



Command Line Interface (CLI)

- User types in instructions to launch or open a program
- User is in direct communication with

Graphical User Interface (GUI)

- User interacts with the computer system by using icons.
- User does not need to know where

the computer system

- Usually, a number of instructions are needed to be typed in to open/launch a program
- Linux

the application resides in the comp.

- User lauches applications using pointers
- Windows

Q- Explain key management tasks carried out by operating system.

Memory Management:

- Allocates memory to processes
- Ensure fair usage of memory (Priorities)
- Organise memory by making use of virtual memory
- Keeps processes seperate
- To release memory when process stops.

Process Management:

- Manages scheduling of process
- Allows multi-tasking

- Handles priorities
- Enables process to share information.
- Prevents interference b/w the processes
- Manages which resource the process requires.

Note: Processing only occurs in RAM

Provision of User Interface:

- Allows a user to communicate with hardware by making navigation around the system easier
- Provides facilities for user to input data.
- Provides facilities to output results to user.
- E.g: GUI, CLI

Interface: Communication method.

Security Management:

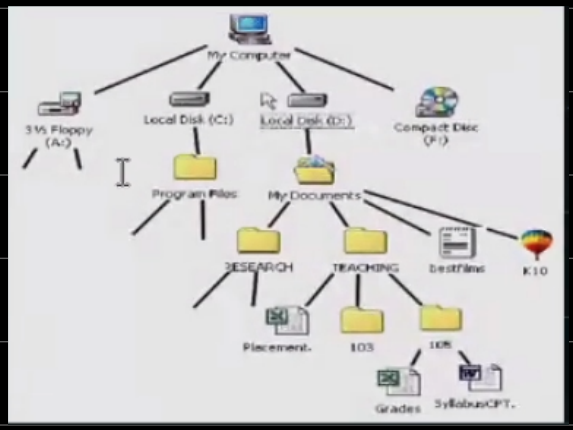
- Sets up user account
- Access rights
- Checks user name and passwords
- Automatic backup
- System Restore

File Management:

- Storage space divided into file allocation unit
- Space allocated for particular files
- Maintains directory structure
- Specifies logical method of file storage (NTFS, FAT 32)
- Provides file naming convention
- Controls file access.
- Specific tasks that can be performed on a file, e.g: open, delete, copy

- Allocation unit: It represents the smallest amount of disk space that can be used to hold a file.

- Directory structure:



Printer Management:

- Installs printer driver
- Sends data to the printer
- Handles error messages
- Sends commands to the printer

Interrupt Handling:

- Identifies priorities to the interrupt
- Saves data on power outage
- Loads appropriate interrupt service routine

Input Output Management:

- Installation of appropriate drivers
- Controls access to data being sent
- Controls access to hardware
- Manages communication b/w devices

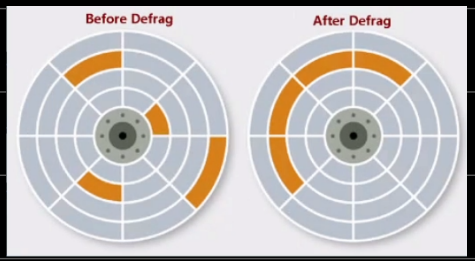
Utility Software

- These softwares analyze and maintain a computer system and make it functional

1) Hard Disk Formatter:

- Makes existing data inaccessible
- Partitions the disk into logical drives
- Prepare the disk for initial use
- Might search for error
- Sets up specified file system

2) Hard Disk Defragmenter



- Re-organises the disk
- Moves split file so they are contiguous
- Creates a large area of contiguous free space.

3) Disk Content Analysis / Disk Repair :

- Check for any error
- Resolves any errors on the disk
- Retrieves files from damaged disk
- Marks bad sectors of the disk.

4) Virus Checker:

- Scans files on the computer for malicious code
- Scans files when they enter the system like when memory stick is inserted.
- Sets up a schedule for virus checking
- Deletes virus
- Regularly updates virus definition (Anti-viruses)

5) Backup Software:

- Creates a copy of the content of the disk
- Can be set up to automatically backup.
- Allows user to decide what is backed up.
- Allows off-site backup
- May encrypt backed up files
- Restores the data if necessary

6) File Compression Utility:

- Compress and decompress files
- Infrequently used files are compressed
- saves space in hard disk

Program Libraries

Q- What is meant by a library routine?

- Pre existing or Pre compiled code
- Can be linked to other programs
- To perform common complex tasks

Q- Benefits of Library Routines

- Less code needs to be written, so saves time
- Pre tested so reduces testing time
- Can be written in different programming languages that enables the developer to use special features of that language.
- Simplifies the program, since just the name of the function is included.
- Can be complex algorithms and no need to work out on how to write them.

Q- Drawbacks of Library Routines

- Compatibility issues; may not work with other code
- Not guranteed through testing; may contain unknown unexpected bugs.
- The code may not meet exact needs; may give unexpected results.

Dynamic Link Library

- A collection of self-contained library program
- They are already compiled
- Linked to main program execution
- Library program code is seperated from ".exe" file.
- Library file only loaded into memory when required at run-time.
- A DLL (.dll) file can be made available to several applications at the same time.
- If a change is made in DLL code then all the programs that use the DLL will get changed.

Q- Benefits of Dynamic Link Library

- The executeable file is smaller
- DLL file only loaded into memory when required at run-time.
- Changes to DLL code are done independantly of the main program, so there is no need to recompile the main program.
- A single dll file can be made available to several applications saving memory space.

Q- Drawbacks of Dynamic Link Library

- The executeable code is not self-contained
- The dll file needed to be included at run-time.
- Appropriate linking software must be available at run-time to import the DLL file.
- The DLL file must be present or else error.
- Unexpected changes to DLL mean program will stop working as expected.
- Malicious changes to DLL file could install a virus on user's computer.

Language Translators

1. Assembler
2. Compiler
3. Interpreter

Assembler

- Programs written in assembly language are translated into object code by an assembler program
- Each instruction in the source code consists of an operand and opcode.
- The software translates low level language into object code for the processor to execute
- The source code uses instructions from the processor's instruction set.

Compiler

- Compiler translates high-level language into machine code for the processor to execute
- Compiler creates an executeable file
- No need to give access to source code, so makes it more difficult for user to modify the code.

Q- Benefits of Compiler

- Produces an executeable file
- User does not have access to source code
- It will probably be faster to run the executeable file
- Code does not have to be compiled each time it is run/executed.
- Does not need the compiler to be present at run-time

Q- Drawbacks of Compiler

- The source code must be re-compiled every time the programmer changes the program.
- Finding error is difficult as error messages are given at the end.
- The source code must be 100% correct for executeable file to be produced

Interpreter

- Interpreter translates high level language into object code for processor to execute line by line.
- The interpreter reads each statement and checks it before running it
- The interpreter halts when it encounters an error.

- The interpreter analyses and checks each line before executing it

Q- Benefits of Interpreter

- Errors can be corrected as they occur
- Can run a partially developed program when developing
- The effect of any change made to the code can be seen immediately.

Q- Drawbacks of Interpreter

- No executable file is produced
- User has access to source code.
- Need to translate the source code again and again, so consumes time.
- Interpreter should be present at the time of execution.

Q- Differences b/w compiler and interpreter.

- Compiler creates an executeable file
- Interpreter does not create an executeable file.

- The compiled program can be independantly distributed. (without source code)

- Interpreter executes each statement immediately after decoding
- Compiler checks the whole program for errors.

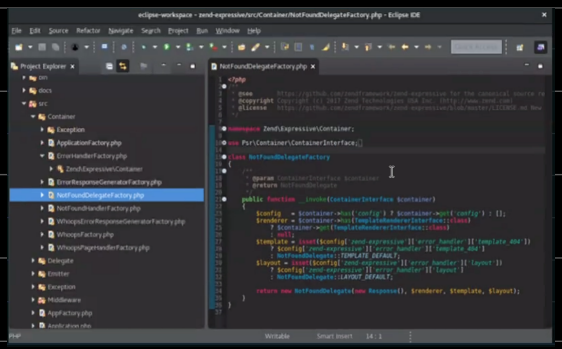
- The interpreter software must be present in main memory every time the program is executed.
- The compiler is not required for a compiled program.

- Cross compilation is possible (compile on one hardware to run on another)

Q- Explain the purpose of language translators.

- To convert higher level or low level programming languages into object code.

Integrated Development Environment (IDE)



- Software application that combines all of the features and tools needed by a software developer

Q- What are the features of an IDE?

- Prettyprint
- Automatic Indentation
- Syntax Checking
- Type Checking
- Highlights any undeclared variables.

Q- What are the features provided by an IDE that assist in initial error detection.

- Identification of unused variables
- Dynamic Syntax Checking
- Type - Checking

Debugging tools that IDE can provide

1. Breakpoint: Run the code to set-point to find errors
2. Report Window: Errors are displayed in a seperate window in IDE
3. Single-stepping: Execute the code line by line
4. Variable watch: Checks the content of variables at specific points.