Chapter 8: System Software

Operating System

→ a software platform that provides facilities for programs to be run, which are of benefit to a user

Operating System Activities

→ User - system interface, this allows to get something useful done
  → Command - line interface
  → Graphical user interface (GUI)

→ Program - hardware interface
  → The OS has to make sure that hardware does what the software wants it to do.
  → Program development tools allows a programmer to write a program without needing to know how the processor works.

→ Resource management
  → When a program has started to run it is called a process. No process can run to completion without interruptions (needing access to the resources provided by computer system)
  → it helps achieve efficiency in computer system by:
    → scheduling of processes
    → resolution of conflicts when two processes require same resource.

→ Memory management
  → There are three different aspects
    1. Memory protection - ensures two programs are not trying to use the same memory location
    2. Memory organization scheme is chosen, ex. virtual memory
    3. Memory usage optimisation, decides which process should be in main memory, and where will it be stored

PAGE No.
DATE / /

⇒ Continuing (OS activities)

→ Device Management
  → Every computer system has a variety of devices- keyboard, mouse, e
    the management of these require:
    1. Installation of appropriate device driver software
    2. Control of usage by processes (Program - hardware)

→ File management
  → Three major features:
    → File naming conventions
    → Directory (folder) structures (structure provided by OS)
    → Access Control mechanisms (accessing and controlling made e2)

→ Security management
  → OS helps by
    → provision for recovery when data is lost
    → prevention of intrusion
    → ensuring data privacy

→ Error Detection and recovery
  → Errors can arise, because code was badly written or it was supplied w wrong set of data, other errors can be due to devices not working properly.
  → The OS should have the capability to interrupt a running process and provide error diagnostics.
  → In extreme cases, the operating system needs to be able to shut down the system in a way that there is no loss of data

PAGE No.
DATE / /

⇒ Utility Software

→ a utility program can be provided by the OS or installed seperately
→ it is executed when needed, ex. some utility programs manage hard disks. It can be executed by the OS or user

⇒ Hard disk formatter and checker

→ formatter:
  → removes existing data from a disk that has been used previously
  → sets up the file system on the disk
  → partitioning the disk into logical drivers (if required)

⇒ A utility program, which can be a part of disk formatter, performs disk contents analysis and, if possible, disk repair when needed. The program checks for errors on disk, some arise due to physical defect (bad sector). bad sector can be caused during manufacturing or mishandling. Due to such problems some files on the disk might no longer be useable. A disk repair utility program marks bad sectors and ensure that the file system no longer tries to use them. If integrity of data is lost, utility can recover some of the data, but if not then it has to delete the files.

⇒ Hard Disk Defragmenter

→ A disk defragmenter utility. A perfectly functioning disk will gradually will become less efficient, because of constant creation, editing and deleting of files leaves them in a fragmented state.

→ A defragmenter utility program reorganizes the file storage to return it to a state where all files are stored in one block access a sequence of sectors. For a large disk this will take some time. It will be impossible if the disk is too full.

PAGE No.
DATE / /

⟹ Backup Software

→ its quite likely that you perform a manual backup of your files

→ an easy way to perform backup is to use a backup utility

it will : establish a schedule for backups

: only create a new backup file when there has been a change.

⟹ File compression

→ A file compression utility program can be used regularly by an operating system to minimise hard disk storage requirements. IF the OS doesn't do it the user can. Compression is the best way to send files (.zip)

⟹ Virus checker

→ A virus checking program should be installed as a permanent facility to protect computer system. It is necessary for a virus checker to be regularly updated and for it to scan all files on a computer system as a matter of routine.

⟹ Program libraries

→ it contains programs which are subroutines, created to carry out particular tasks.

→ advantage of using Program Libraries - it saves a lot of time as it is already tried and tested and free of errors.

→ ex. NAG - Numerical Algorithms Group, it contains 1600 procedures for mathematical and statistical operations.

→ If a compiler is used for translation, the compiler produces an executable file (object / machine code), this cannot be executed by itself, it needs to be linked to any subroutine used in the code. This has a major disadvantage, it increases the storage space requirement, also increases memory usage, because every program using subroutine has to have its own copy.

→ alternative is to use dynamic linked library (DLL), DLL requires a small piece of code to be included, this links it to the routine. The advantage is executable files needs less storage. The disadvantage is if routine is not available or DLL is corrupted it wont work

Memory req is also reduced

PAGE NO.
DATE / /

Compilers and Interpreters

→ Interpreter

1. interpreter program begins execution
2. The first line of the source code is read
3. The line is analysed
4. if an error is found, it is reported and interpreter halts execution
5. if no error is found, the line being analysed is converted into intermediate code.
6. The interpreter uses intermediate code to execute required action
7. Steps 3 to 7 repeated, after next line is read.

→ Compiler

1. Compiler program begins execution
2. The first line of source code is read
3. The line is analysed
4. if an error is found, it is recorded.
5. if no error is found, the line is converted into intermediate code
6. The next line of source code is read, steps 3-6 repeated
7. When the whole of source code has been dealt with
   - → if no error is found, the whole intermediate code is converted into object code.
   - → if error/s are found, the list of these is output and object code is not produced.

When to use Interpreter and when Compiler

Interpreter	Compiler
use when program is being developed	use when sure that program contains no errors
Reason	Reason
→ an error can lead to several other errors.	→ an executable file can be created
→ it can detect and correct early errors.	→ this can be distributed for general use
→ it provides debugging facility	→ execution of the program will be faster

PAGE No.
DATE / /

⇒ The advantages and disadvantages to a

→ Programmer

→ interpreter has advantage when program is being developed

→ interpreter has disadvantage while sharing as source code has to be shared.

→ compiler has advantage that an executable file can be distributed to users, so the users have no access to the source code.

→ Users

→ for an interpreted program, the interpreter and source code has to be available

→ for a compiled program, the ~~compiler~~ ~~interpreter~~ ~~and~~ object code has to be available

→ compiled object code will provide faster execution than interpre

→ compiled object code is less secure because it could contain a virus.

⇒ Features in IDE (Integrated Development Environment)

→ Pretty printing - automatically colour codes keywords.

→ Context sensitive prompts - it displays hints or a choice of keywords

→ Dynamic syntax checks - performs syntax checks and alerts programmer

→ Expanding and collapsing code blocks - it saves scrolling if you can collapse blocks of statement.

→ Debugging - finding and correcting errors, often called bugs.