Chapter 7: System software

> System softwares are softwares responsible for making the computer functional so that you can use it for specific task using application software.

- Operating system is an example.

Operating System

> Operating system manages resources available and creates environment for user to run application programs.

- Functions are:
  - Provide user-interface: CLI or GUI.
  - Hardware management
  - Resource management
  - Memory management
  - Device management
  - File management
  - Security management.

Utility software

> Softwares used to maintain the computer so that it runs in good condition.

- Hard disk formatter and checker: Remove existing files, set up file system, partition disks.
- Hard disk defragmenter: Reorganises the file storage to return it to state where files are stored in one block across a sequence of sectors.
- Backup software: Creates copy so that you can retain your data in case the original can't be accessed.
- File compression: Reduces file size to save space.
- Virus checker/ Antivirus: Protects from malware.

Program libraries

> Collection of programs that perform particular tasks

- Advantages:
  - Can develop software faster.
  - Usually well tested
  - Generally optimized
- Disadvantages:
  - Each program creates own copy. So, increases storage requirements.
  - Less customizable.

Dynamic linked libraries

> Instead of having own copy, programs share DLL.

- Advantage:
    - Less storage space used
    - If new version is available, only one needs to be changed.
- Disadvantages:
    - Reliance on DLL. If its corrupted or has bugs.

Language translators

Compilers and Interpreters

- Compiler:
    - Execution is line by line too
    - If there is error, it is recorded
        - else converted to intermediate code
    - If whole source code has no error, complete intermediate code is converted into object code.
    - If there are errors, errors are displayed.
    - Program doesn't need to be compiled before every use if there is no change.
    - Advantage:
        - Executable file can be distributed to users without source code.
        - Compiled object code provide faster execution and dont need to be generated from source code.
    - Disadvantages:
        - Compiled object code could contain virus and it would be hard to know.
        - Debugging without complete program is hard.
- Interpreter:
    - Code is executed line by line
    - If no error, convert to intermediate code and execute
        - else report error and halt.
    - Program need to be interpreted every time it is executed.
    - Advantages:
        - Easier debugging since only relevant part can be executed and testted without analysing whole program.
    - Disadvantage:
        - Errors are discovered one at a time sequentially. This is bad if there are many syntax error in program.
        - Source code needs to be sent to end user.

> Java takes a different approach. Computer that has to run program needs a Java Virtual Machine which interprets Java Byte Code which is obtained by compiling Java program.

IDE and Features

> Integrated Development Environment(IDE) is software that can be used to write source code and compile/interpret it with some other features.

- Features:

- Prettyprinting: Color coding different parts of program
- Context-sensitive prompt: Hints while typing
- Dynamic syntax check and highlighting
- Expanding and collapsing code blocks
- Debugging:
    - Set `breakpoint`: Point in code where program stops.
    - Program can be `Stepped through`: Execute one line at a time.
    - `Variable watch`: Watch values of variable as program runs.

bishaltwr@gmail.com