**Chapter 8.System Software**

Operating system: a software platform that provides facilities for programs to be run which are of benefit to a user

-System software-

* User-system interface

-a command-line interface (CLI)

-a graphical user interface (GUI)

* Program-hardware interface

-The software uses the hardware

-The operating system has to ensure that the hardware does what the software wants it to do

-Allow a programmer to write a program without needing to know the details of how the hardware

* Resource management

-process: a program that has begun execution

-(time) scheduling of process

-resolution of conflicts when two processes require the same resource

* Memory management

-Memory protection ensures that one program does not try to use the same memory locations as another program

-The memory organisation scheme is chosen to achieve the best usage of limited memory size, for example, virtual memory involving paging or segmentation

-Memory usage optimisation involves decisions about which processes should be in main memory at any one time and where they are stored in this memory

* Device management

-installation of the appropriate device driver software

-control of usage by processes

-power management

* File management

-file naming conventions

-directory (folder) structures

-access control mechanisms

* Security management

-provision for recovery when data is lost

-prevention of instrusion

* Error detection and recovery

-causes of an error

-operating system should have the capability to interrupt a running process

-provide error diagnostics where appropriate

-in extreme cases, the operating system needs to be able to shut down the system without loss of data

-placed in a different category

-make an abbreviated list

-Utility software-

Hard disk formatter and checker

* removing existing data from a disk that has been used previously
* setting up the file system on the disk, based on a table of contents that allows a file recognised by the operating system to be associated with a specific physical part of the disk
* partitioning the disk into logical drives if this is required

Disk repair

* can mark bad sectors and ensure that the file system no longer tries to use them
* integrity of files has been affected, the utility might be able to recover some of the data but otherwise it has to delete the files

Hard disk defragmenter

* Find files that are not stored in contigous blocks and move them together
* create more free disk space
* it will be impossible if the disk is too full because of the lack of working space for the rearrangement

File compression

* Reduce the size of the file

Backup

* Making a copy of data in case of loss

Virus checker

* scan a file when the file initially entered the system
* regularly updated
* scan all files on a computer system as a matter of routine

-Program libraries-

ADVANTAGES:

* relatively free from errors and more robust
* already tested
* saves time and programming easier

dynamic linked library (DLL)

* precompiled and better performance
* executable files for all programs need less storage space

shortcomings:

* if for some reason the DLL becomes corrupted, the program will fail
* there may be a malicious code in the DLL

-Language translators-

Compliers(编译器), interpreters(解析器) and assemblers(汇编器)

|INTERPRETERS|

* The interpreter program, the source code file and the data are all made available
* Interpreter program begins execution
* first line of the source code is read
* line is analysed
* error is found, this is reported and the interpreter program halts execution
* no error is found, the line of source code is converted to an intermediate code
* interpreter program uses this intermediate code
* Steps 4-8 are repeated

ADVANTAGES:

* When a program is being developed because errors can be identified as they occur and corrected immediatedly

SPECIAL POINTS:

* For an interpreted program, the interpreter and the source code have to be available

DISADVANTAGES:

* Source code is distributed to users because the source code has to be sent to each user, which would cause copyright problems

|COMPLIERS|

* The complier and the source code file are made available
* - the whole of the source code-
* If no error is found in the whole source code the complete converted into object code
* If any errors are found a list of these is output and no object code is produced

ADVANTAGES:

* Code will provide faster execution that is possible for an interpreted program
* An executable file can be distributed to users, so the users have no access to the source code

SPECIAL POINTS:

* For a complied program, only the object code has to be available each time that an error-free program is run

DISADVANTAGES:

* Code is less secure because it could contain a virus

Java

when a programmer writes a Java program this is complied first of all to create what is called Java Byte Code.When the program is run, this code is interpreted by the Java Virtual Machine. The Java Byte Code can be transferred to any computer that has a Java Virtual Machine installed

IDE

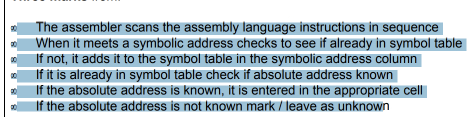
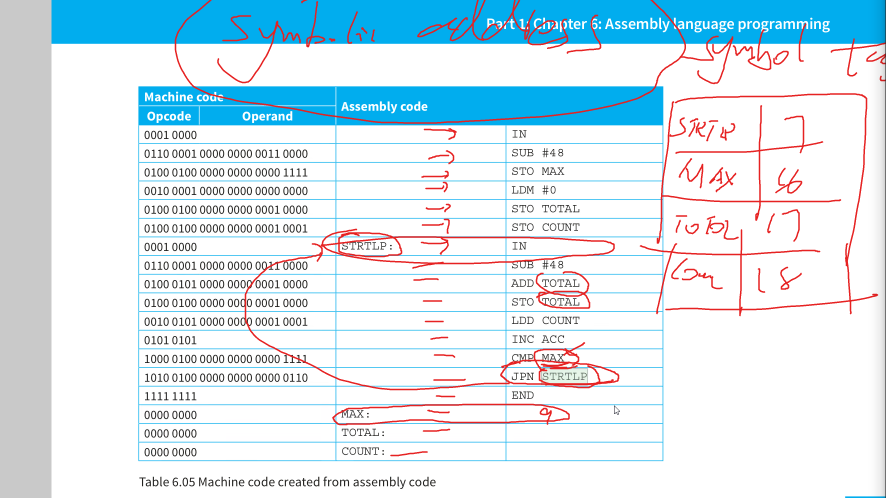
* Prettyprinting
* Context-sensitive prompts
* Dynamic syntax checks
* Expanding and collapsing code blocks
* Debugging
  + break point
  + single stepping
  + watch window (SOME)

-----finding and correcting errors

|ASSEMBLERS|

-two-pass assembler-

-FIRST-PASS-

* removal of comments
* replacement of a macro name used in an instruction by the list of instructions
* uses a symbol table
* 
* read the assembly language program one line at a time
* check the opcode is in the instruction set
* 

-SECOND-PASS-

* uses the symbol table and a lookup table that contains the binary code for each opcode
* the output from the second pass will be a machine code program
* read the assembly language program one line at a time

