ALevel CS C08 System Software(1)

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

Operating system: a software platform that provides facilities for programs to be run which are of benefit to a user
why need OS:
1. The hardware is unusable without an OS // hides complexity of hardware from user
2. Acts as an interface/ controls communications between user and hardware / hardware and software // or by example 1
3. Provides software platform / environment on which other programs can be run
User-system interface: Allow the user to get the software and hardware to do something useful
1. a commend-line interface(**CLI**)
2. a graphical user interface(**GUI**)
Program-hardware interface: Program development tools associated with a programming language allow a programmer to write a program **without needing to know the details of how the hardware**, particularly the processor, actually works.
Resource Management:
1. **Scheduling of processes**
2. resolution of conflicts when **two processes require the same resource**
Memory Management: Handles the allocation of memory to processes
1. Memory protection ensures that one program does not try to use the same memory locations as another program
2. The memory organisation scheme is chosen to achieve the best usage of limited memory size, for example, virtual memory involving paging or segmentation
3. 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
tasks:
1. Allocates / deallocates RAM to programs/tasks/processes
2. Keeps track of allocated and free memory locations
3. Swaps data to and from the hard drive
4. Handles virtual memory
5. Paging // segmentation
6. Memory protection, preventing a process accessing memory not allocated to it
Device(peripheral) management: Include Monitor screen, keyboard, printer, webcam etc.
tasks:
1. Install/manage device drivers
2. Control of hardware usage by processes // allocation of devices to processes // inter process communication
3. Device detection
4. Power Management
5. Keep track of device status (free or busy)
6. Buffer management
File management:
1. File naming conventions
2. directory (folder) structures
3. access control mechanisms
Security Management: Provides user accounts and passwords
1. Provision Tor recovery when data Is loss
2. prevention of intrusion
3. ensuring data privacy
tasks:
1. Sets up user accounts
2. Checks usernames, passwords // Authentication
3. Implements access rights
4. Automatic backup
5. System restore / roll back (to previous stable state)
Error detection and recovery: Errors can arise in the execution of a program either because it was badly written or because it has been supplied with inappropriate data.
tasks:
1. Deals with interrupts
2. Deal with run time errors generated by software
3. Deal with hardware faults
4. Error diagnostic messages
5. Deadlock detection and recovery
6. Safe-mode boot-up routines
7. System shutdown
8. Saves system restore points
Interrupt processing: Handles the signals sent when the attention of the processor is required elsewhere
Provision of a software platform: Provides an environment within which programs can be run

Disk formatter tasks: setting up a disk so it is ready to store files
1. **removing existing data** from a disk that has been used previously
2. **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
3. **partitioning the disk** into logical drives if this is required.
Disk repair: scans for errors in a disk and corrects them
purpose:
1. **Checks for any errors** / inconsistencies / bad sectors on the disk
2. **Resolves any errors on the disk**
3. **Retrieves files** / data from a damaged disk // re-constructs directory // recovers disc when data corrupt
4. Marks bad sectors on the disk // marks bad sectors as unusable
Hard disk defragmenter: move parts of files so that each file is contiguous in memory
fragmented state: the constant creation, editing and deletion of files
A defragmenter utility program reorganises the file storage to return it to a state where all files are stored in one block across a sequence of sectors.
step:
1. **Re-organises** the disk contents
2. **Moves split files** so they are **contiguous**
3. **Creates a large area of (contiguous) free space**
Backup software: creates a copy of data in case the original is lost
1. establish a schedule for backups
2. only create a new backup file when there has been a change.
File compression:
used regularly by an operating system to minimise hard disk storage requirements.
Virus checker:
1. installed as a permanent facility to protect a computer system.
2. regularly updated and for it to scan all files on a computer system as a matter of routine.

Program libraries
The 'programs' in a program library are usually **subroutines** created to carry out particular tasks. A programmer can use these within their own programs.
how to use program libraries:
1. Program libraries store pre-written functions and routines
2. The program library can be referenced/imported
3. the functions/routines can be called in her own program
DLL(dynamic linked library):
1. When a DLL routine is available the executable code just requires a small piece of code to be included.
2. This allows it to link to the routine, which is stored separately in memory, when execution of the program needs it.
3. Many processes can be linked to the same routine.
DLL advantages:
1. the executable files for all programs need **less storage space**.
2. **Memory requirement** is also minimised.
3. if a new version of the routine becomes available it can be loaded into memory so that any program using it is **automatically upgraded**.
DLL disadvantages:
1. **relying on the routine** being available and performing the expected function.
.2. If for some reason the DLL becomes **corrupted** or **a new version has bugs** not yet discovered the program will fail or produce an erroneous result.
3. The user running the program will find it **difficult to establish** what needs to be done to get the program to run without error.

ALevel CS C08 System Software(2)

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

language translator purpose:
To convert a (higher level) programming language to a different form
Assembler: Assembler language
Interpreter steps:
1. The interpreter program, the source code file and the data to be used by the source code program are **all made available**.
2. The interpreter program begins execution.
3. The first line of the source code is read.
4. The line is analysed.
5. If an error is found, this is reported and the interpreter program halts execution.
6. If no error is found, the line of source code is converted to an intermediate code.
7. **The interpreter program uses this intermediate code to execute the required action.**
8. The next line of source code is read and Steps 4–8 are repeated.

Compiler steps:
1. The compiler program and the source code file are made available but no data is needed.
2. The compiler program begins execution.
3. The first line of the source code is read.
4. The line is analysed.
5. If an error is found this is recorded.
6. If no error is found the line of source code is converted to an intermediate code.
7. The next line of source code is read and Steps 4–7 are repeated.
8. When the whole of the source code has been dealt with one of the following happens.
    1. If no error is found in the whole source code the complete intermediate code is **converted into object code**.
    2. If any errors are found a list of these is output and no object code is produced.

programmer of creating interpreted or compiled programs:
1. An interpreter has advantages when a program is being developed because errors can be identified as they occur and corrected **immediately** without having to wait for the whole of the source code to be read and analysed.
2. An interpreter has a disadvantage in that during a particular execution of the program, parts of the code which contain syntax errors may not be accessed so if **errors are still present**, they are not discovered until later.
3. An interpreter has a disadvantage when a program is error free and is distributed to users because the **source code has to be sent to each user**.
4. A compiler has the advantage that an executable file can be distributed to users, so the **users have no access to the source code**.

user of interpreted or compiled programs:
1. For an interpreted program, the interpreter and the source code have to be available each time that an error-free program is run.
2. For a compiled program, only the object code has to be available each time that an error free program is run.
3. Compiled object code will provide faster execution than is possible for an interpreted program.
4. Compiled object code is **less secure** because it could contain a virus.

why choose interpreter:
1. one error in a program can **lead to several other errors** occurring
2. an interpreter can **detect and correct an early error** so limiting subsequent ones
3. the debugging facilities provided in association with the interpreter speed this process.

why choose compiler:
1. an executable file can be created
2. this can be **distributed** for general use
3. execution of the program will be **faster** than if an interpreter were used.

IDE features:
Prettyprinting: Prettyprint refers to the presentation of the program code typed into an editor.
Context-sensitive prompts: This feature displays hints (or a choice of keywords) and available identifiers that might be appropriate at the current insertion point of the program code.
Dynamic syntax checks: When a line has been typed, some editors perform syntax checks and alert the programmer to errors.
Expanding and collapsing code blocks:When working on program code consisting of many lines of code, it saves excessive scrolling if you can collapse blocks of statements.
Debugging: finding and correcting errors, often called ‘bugs’, in a program
debug tools:
1. Breakpoints
2. Single stepping
3. Report windows