

Operating Systems Exercises week 2 – Complete answers.

Tutorial questions.

1. What is the purpose of the command interpreter? Why is it usually separate from the kernel?

Answer:

It reads commands from the user or from a file of commands and it initiates execution, usually by turning them into one or more system calls. It is usually not part of the kernel since the command interpreter is more subject to change and users may also want to choose between several different ones that may be available.

2. What is the purpose of a scheduler?

Answer:

To schedule resources to user programs i.e. to decide in which order the various user programs will be allowed to use a given resource. All resources in the system need some form of scheduler, from CPU and memory, to I/O devices and secondary memory.

3. What is an operating system model?

Answer:

It is a general framework, which indicates how the various components/functions of an operating system can be structured together to produce the complete OS. Each different general approach has their own strengths and weaknesses.

4. What is the purpose of system calls? Why are they necessary?

Answer:

NOTE – this question will be included in the portfolio so don't show them the answer!

System calls allow user-level processes to request services from the operating system.

Because of the need to enforce protection on the system to prevent user programs from interfering with each other, user programs do not normally have direct access to hardware, except CPU and some memory. OS has direct access to I/O and disks, etc., so user program must ask OS if it wants some I/O, etc, performing.

5. What is the purpose of system programs?

Answer:

System programs can be thought of as bundles of useful system calls. They

provide basic functionality to users so users do not need to write their own programs to solve common problems to do with the management/maintenance of the computer system (e.g. disk defragmenter) and application program development (compilers, etc.).

6. What is the main advantage of the layered approach to system design?

Answer:

As in all cases of modular design, designing an operating system in a modular way has several advantages. The system is easier to debug and modify because changes affect only limited sections of the system rather than touching all sections of the operating system. Information is kept only where it is needed and is accessible only within a defined and restricted area, so any bugs affecting that data must be limited to a specific module or layer.

7. Why is the separation of mechanism and policy a desirable policy?

Answer:

Mechanism and policy must be separate to ensure that systems are easy to modify. No two system installations are the same, so each installation may want to tune the operating system to suit its own needs. With mechanism and policy separate, the policy may be changed at will while the mechanism stays unchanged. This arrangement provides a more flexible system. You do not want to have to change the code of the OS in order to change a policy decision.

Practical exercises.