

Operating Systems Exercises week 1 – Selected answers.

1 What is the main advantage of multiprogramming?

Answer:

Multiprogramming makes efficient use of the CPU by overlapping the demands for the CPU and its I/O devices from various users. It attempts to increase CPU utilisation by always having something for the CPU to execute, and thus enables more jobs to be processed in a given time period than without multiprogramming.

2. In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation leads to problems with security.

- a) Give 2 examples of the type of security problem you could find?
- b) Can we ensure the same degree of security in a time-shared machine as we have in a dedicated machine? Explain your answer.

Answer:

- a) Shared resources of any sort means that one program can copy or corrupt data/programs that belong to the other program by for example accessing main memory that is allocated to the other program or accessing secondary memory that belongs to the other program (e.g. files on a hard disk).
- b) Probably not. There is an inherent risk in any system which involves sharing of resources that the different parties to the sharing (people or programs) might interfere with each other. On a dedicated machine there is no such inherent risk. Although the operating system would be designed to try and prevent interference between different users/programs, any protection scheme devised by man can inevitably be broken by him, and the more complex the scheme, the more difficult it is to feel confident of its correct implementation. The risk of sharing resources can be reduced but it is not possible to guarantee that it has been eliminated.

3. Define the essential properties of the following types of operating system:

- a) Batch
- b) Interactive
- c) Time-sharing

Answer:

- a) Batch – Jobs with similar needs are batched together and run through the computer as a group by an operator or automatic job sequencer. Performance is increased by attempting to keep CPU and I/O devices busy at all times through buffering, off-line operation, spooling, and multiprogramming. Batch is good for executing large jobs that need little interaction; it can be submitted and picked up later.
- b) Interactive – User (task initiator) is on-line with computer system and interacts with it via an interface. It is typically composed of many short transactions where the results of the next transaction may be unpredictable. Response time needs to be short (few seconds) since the user submits and waits for the result.
- c) Time-sharing – Uses CPU scheduling and multiprogramming to provide economical interactive use of a system. The CPU switches rapidly from one user

to another i.e. the CPU is shared between a number of interactive users. Instead of having a job defined by spooled card images, each program reads its next control instructions from the terminal, and output is normally printed immediately to the screen.

4. We have stressed the need for an operating system to make efficient use of the computing hardware. When is it appropriate for the operating system to forsake this principle and to “waste” resources? Why is such a system not really wasteful?

Answer:

Single-user systems should maximise use of the system for the user. It may strictly speaking be wasteful to have a computer sitting idle on someone’s desk, but the comparative cost of the computer compared to the human user makes it more important to gain maximum benefit and convenience to the human user. From the viewpoint of an organisation it is more beneficial for the user to be productive and have a computer readily available for their immediate use than to maximise the use of the computing resources it has available.

Also such elements as a GUI interface is wasteful of computing power, but it optimises the user’s productivity and eases their interaction with the system. It is not wasteful because the user is nowadays the more expensive resource and the efficiency of the user is more important than that of the computer.

5. Under what circumstances would a user be better off using a time-sharing system, rather than a personal computer?

Answer:

When there are few other users on a time-sharing system and the task is computationally large, time-sharing makes sense (assuming the mainframe is significantly more powerful than a PC). The full power of the system can be brought to bear on the user’s problem. The problem can be solved faster than on a personal computer. Another case is where there are lots of other users needing the same set of resources at the same time. A personal computer is fine when the job is small enough for it to execute the program to the user’s satisfaction.

6. How does the distinction between supervisor mode and user mode function as a rudimentary form of protection (security) system?

No answer – question and answer to be included in portfolio that is to be handed in.

7. What are the differences between a software interrupt (trap) and a hardware interrupt? What is the use of each?

No answer – question and answer to be included in portfolio that is to be handed in.

8. For what types of operations is DMA useful? Explain your answer.

Answer:

DMA is useful for transferring large quantities of data between memory and fast devices such as hard disks. In DMA the data is transferred from peripheral to memory without using the CPU. The CPU is only interrupted when a block of data has been transferred. Without DMA each byte/word of data is transferred to memory via the CPU and thus the CPU is interrupted for every byte/word of data (which is what happens with slow devices like mice and keyboards). Thus the CPU is interrupted much less using DMA (and thus has more time to run application program code) than it would be without it. The transfer of data also completes more quickly.

9. Some early computers protected the operating system by placing it in a memory partition that could not be modified by either a user job or the operating system itself. Describe two difficulties that you think could arise with such a scheme.

Answer:

The data required by the operating system (passwords, access control data, accounting info, etc) would have to be stored in or passed through unprotected memory and thus be accessible to unauthorised users. If the partition was some form of read-only memory, then updating components of the OS would be more difficult.

10. Protecting the operating system is crucial to ensuring that the computer operates correctly. Provision of this protection is the reason behind dual-mode operation, memory protection and the timer. To allow maximum flexibility, however, we would also like to place minimal constraints on the user. The following is a list of operations that are normally protected. What is the minimal set of instructions that must be protected?

- a) Change to user mode.
- b) Change to supervisor mode
- c) Read from OS memory
- d) Write into OS memory
- e) Fetch an instruction from OS memory
- f) Turn on timer interrupts
- g) Turn off timer interrupts.

Answer:

The minimal set of instructions that must be protected are:

b, d, g