HOMEWORK 1
SOLUTIONS
CSE 120 OPERATING SYSTEMS
FALL 2006

1) [Silberschatz 7th Edition - Chapter 1 - Q 1.10] What is the purpose of interrupts? What is the difference between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?

[1+1+1+1 points]

Ans:

- Interrupts are used to signal the occurrence of an event or a change in the normal flow of the program.
- A trap (or an exception) is a software-generated interrupt caused by either an error (for example, division by zero or invalid memory access) or by a specific request from a user program that an operating-system service be performed.
- The latter is an example of when traps are generated intentionally by a user program.

2) Compare and contrast between the two common models of inter-process communication.

[4 points]

Ans:

- Shared memory & Message passing are the 2 common models of IPC.
- Message passing is useful for exchanging smaller amounts of data, because no conflicts need be avoided. It is also easier to implement than shared memory.
- Shared memory allows maximum speed and convenience of communication, as it can be done at memory speeds when within a computer.
- Shared memory is faster than message passing as message passing systems are usually implemented using system calls and thus require kernel intervention which is more time consuming.
- In contrast, in shared-memory systems, system calls are required only to establish memory regions. Once shared memory is established, all accesses are treated as routine memory accesses, and no assistance from the kernel is required.

3) [Silberschatz 7th Edition - Chapter 3 - Q 3.2] Describe the actions taken by a kernel to context-switch between processes.

[3 points]

Ans:

- When a context switch occurs, the kernel saves the context of the old process in its PCB and loads the saved context of the new process scheduled to run.
- The system needs to save the current context of the process currently running on the CPU so that it can restore that context when its processing is done, essentially suspending the process and then resuming it.
- The context is represented in the PCB of the process; it includes the value of the CPU registers, the process state and memory management information.

4) List two challenges an OS faces when passing parameters between user and kernel mode. Describe how an OS can overcome them.

[2+2 points]

Ans:

- The simplest approach to pass parameters to the OS is to pass them through registers.
- Sometimes, there can be more parameters than registers.
- In these cases, the parameters are stored in a block, or table, in memory, and the address of the block is passes as a parameter in the register.
- This can also sometimes limit the number or length of the parameters being passed.
- To overcome this problem, parameters can also be pushed onto the stack by the program and popped off the stack by the OS.

5) Give 2 reasons why memory protection is so critical for an operating system. What kinds of protection must a typical operating system provide?

[3 points]

Ans:

- If a computer system has multiple users and allows the concurrent execution of multiple processes, then access to data must be regulated.

- Protection provides a mechanism for controlling the access of processes or users to the resources defined by a computer system.
- Protection can improve reliability by detecting latent errors at the interfaces between component subsystems.
- A protection-oriented system provides a means to distinguish between authorized and unauthorized usage.
- A typical OS must be able to protect itself from user programs; protect programs from each other; and it may or may not protect user programs from itself.

6) What would be the expected behavior if a privileged instruction is executed in user mode? In a typical operating system, should there be a privileged instruction to switch to kernel mode?

[1+1 points]

Ans:

- If an attempt is made to execute a privileged instruction in user mode, the hardware does not execute the instruction but rather treats it as illegal and traps it to the OS.
- In a typical OS, there should not be a privileged instruction to switch to kernel mode.