Solution - Mid Term Exams

CPS-410: Operating Systems

Instructor: **Dr. Hammad Afzal**

Q1: Explain interrupts and traps, and provide a detailed account of the procedure that an operating system handles an interrupt. [5]

An interrupt is an event that requires the attention of the operating system. These events include the completion of an I/O, a key press, the alarm clock going off, division by zero, accessing a memory area that does not belong to the running program, and so on.
A trap is an interrupt generated by software.

When an interrupt occurs, the following steps will take place to handle the interrupt:

The executing program is suspended and control is transferred to the operating system. Mode switch may be needed.
A general routine in the operating system examines the received interrupt and calls the interrupt-specific handler.
After the interrupt is served, a context switch transfers control back to a suspended process. Of course, mode switch may be needed.

See p. 8 of our textbook and class notes.

Q2: What is a context? Provide a detail description of all activities of a context switch. [5]

A process needs some system resources to run successfully. These system resources include process ID, registers, memory areas (for instructions, local and global variables, stack and so on), various tables (i.e., process table), and a program counter to indicate the next instruction to be executed. They constitute the environment or context of a process. The steps of switching process A to process B are as follows:

- Suspend A's execution
- Transfer the control to the CPU scheduler. A CPU mode switch may be needed.
- Save A's context to its PCB and other tables.
- Load B's context to register, etc.
- Resume B's execution of the instruction at B's program counter. A CPU mode switch may be needed.

Q3: Explain the concept of Dual-Mode, elaborating the differences between modes. [3]

Some instructions designated as privileged, only executable in kernel mode. For this purpose, dual modes are used: User mode and kernel mode. Dual-mode operation allows OS to protect itself and other system components. When a user application runs, system is in user mode.

When a user application requests a service from operating system (via a system call), it must transition from user to kernel mode.

Mode bit provided by hardware. Provides ability to distinguish when system is running user code or kernel code.

System call changes mode to kernel, return from call resets it to user.

Q4. Organize the following concepts in order of (i) increase in Size, (ii) Access Time.
Cache, Registers, Disk Storage, Main Memory

Increase in Size: Registers, Cache, Main memory, Disk Storage.
Access Time: Registers, Cache, Main memory, Disk Storage.

Q5. What are the two approaches to implement commands (in context of Command Line Interpreters (CLI)). Also highlight the impact of the two approaches on the size of CLI) [3]

Command Line Interface (CLI) or command interpreter allows direct command entry. Sometimes implemented in kernel, whereas others (Windows, Xp) treat command interpreter as a special program that is running when user first logs on. Sometimes multiple flavors implemented – shells (E.g. Bourne shell, C shell, Bourne-Again shell, Korn shell in Unix and Linux)

Two approaches to implement commands.

1st Approach: Command interpreter itself contains code to execute command. Number of commands determine size of command interpreter since each command has its own code.

2nd Approach: Implement commands through system programs (in UNIX). Command interpreter does not understand the command. It only identifies the file to be loaded in memory and execute. E.g. rm file.txt: The code to implement rm command will be in file rm.

Q6 .What are the methods to pass parameters to the Operating System during System Call? [2]

Three general methods used to pass parameters to the OS.
Simplest: pass the parameters in registers. In some cases, may be more parameters than registers

Parameters stored in a block, or table, in memory, and address of block passed as a parameter in a register. This approach taken by Linux and Solaris

Parameters placed, or pushed, onto the stack by the program and popped off the stack by the operating system

Q7. List any 6 of the attributes that an operating system maintains to keep track of information about a process. [3]

1. Process state
2. Program counter
3. CPU registers
4. CPU scheduling information
5. Memory-management information
6. Accounting information
7. I/O status information

Q8. In the context of Process State Diagram, explain the transition from "Waiting" to "Suspend" and "Suspend to Ready". You can go into the further detailed division of Suspended state.

Waiting → Waiting/Suspend: If there are no ready processes, then at least one blocked process is swapped out to make room for another process that is not blocked. This transition can be made even if there are ready processes available, if the OS determines that the currently running process or a ready process that it would like to dispatch requires more main memory to maintain adequate performance.

Waiting /Suspend → Ready/Suspend: A process in the Blocked/Suspend state is moved to the Ready/Suspend state when the event for which it has been waiting occurs.

Q9: Briefly describe the "Long term scheduling" and "Short Term Scheduling". Which scheduling is not performed in modern Microsoft Windows based OS and why? [3]

Long-term scheduler (or job scheduler) – selects which processes should be brought into the memory (ready queue)

Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU.

Long Term scheduling is not performed in Modern MS Windows.

Q10: Elaborate the difference between Single and Multi threaded processes in terms of the constituents of Process Control Block. (only draw diagram) [2]