Chapter 3 Questions

Questions may be asked in class, on quizzes, and on exams

* + What is a process?

Process is a program in execution

* + 1. what is the difference between a process and a program?

Program is a passive entity, such as a file containing a list of instructions stored in disk. Process is an active entity with a program counter specifying the next instruction to execute and a set of associated resource.

* + What does a process include?
    1. Stack which stores temporary data when invoking functions
    2. Heap is memory which is dynamically allocated in program runtime
    3. Data is about global variables
    4. Text section is the executable code
    5. What is a Process Control Block (PCB)?

It is a database of all the data needed to start, restart a process, along with some accounting data.

is a data structure in the operating system kernel containing the information needed to manage the scheduling of a particular process.

* + 1. What does a PCB do?

Provide information for specific process

* + What states can a process be in?

New, Ready, Running, Waiting, End

* + 1. be able to analyze and/or describe either Silberschatz’s 5-state or Stallings’ 7-state models
  + What important part of the process switch operation is not shown in Figure 3.4?

terminal

* + What does it mean that the long-term scheduler controls the degree of multiprogramming?

The degree of multiprogramming is the number of processes currently in memory. this scheduler dictates what processes are to run on a system, and the degree of concurrency to be supported at any one time – whether many or few processes are to be executed concurrently, and how the split between I/O-intensive and CPU-intensive processes is to be handled.

* + What is the difference between a CPU-bound and an I/O-bound process?

I/O-bound spends most of the time doing I/O while CPU-bound process spends most of time doing computing

* + What is happening in the computer and/or OS during a context switch?

Context switch means CPU switch from a process or thread to another.

Kernel save the old process in its PCB and loads the saved context of the new process to run.

Context-switch time is pure overhead cause system does no useful work.

* + What are the different ways resources can be shared between a parent and a child process?

When a sub-process is spawned using fork, it inheritates the parent's process opened handles, files, console input/output, and anonymous pipes.

it is by using named pipes, and named files. A file can be memory mapped so this effectively just creates a shared buffer space.

In general, when a process creates a child process, that child process will need certain resources (CPU time, memory, files, I/O devices) to accomplish its task. A child process may be able to obtain its resources directly from the operating system, or it may be constrained to a subset of the resources of the parent process. The parent may have to partition its resources among its children, or it may be able to share some resources (such as memory or files) among several of its children. Restricting a child process to a subset of the parent’s resources prevents any process from overloading the system by creating too many child processes.

In addition to supplying various physical and logical resources, the parent process may pass along initialization data (input) to the child process.

* + What are the different ways parents and children can execute (with respect to each other)?

**1.** The parent continues to execute concurrently with its children.

**2.** The parent waits until some or all of its children have terminated.

* + What is the difference between an independent and a cooperating process?

Independent doesn’t share any data with any other processes executing in the system

Cooperating can affect or be affected by other processes in the system.

* + What might be the value of interprocess communication (IPC)?

IPC allows cooperating processes to exchange data which is send data to and receive data from other processes without shared address. IPC has two fundamental models: shared memory and message passing.

* + Is sharing a resource such as a common set of data considered IPC? Why or Why Not?

Only cooperating processes need to share a resource.

* + What makes message passing block or non-blocking?
  + Direct or indirect communication
  + Synchronous or asynchronous communication
  + Automatic or explicit buffering
  + What makes either one of these operations synchronous or non-synchronous?
  + End of Chapter
    1. Exercises: 3.1, 3.2, 3.8, 3.11, 3.12, 3.13, 3.16
    2. Programming Problems: none