As a process executes, it changes state. There are four general states of a process: (1) ready, (2) running, (3) waiting, and (4) terminated. Discuss at least two of these four states and the importance of each.

As a process runs, it may create other processes. The creating process is named the parent while the new process is known as the child process.

When processes are ready for execution, they are placed in a “ready” queue, that is stored as a linked list. The ready queue plays an important role in pointing to process control blocks (PCB) and CPU scheduling. The ready queue’s header points to the first PCB, which in turn points to the next and the next…etc. The PCB carries valuable information about a process such as process state, memory management, scheduling information and I/O status information. Processes that are ready are picked up by the CPU scheduler, to which the schedule will assign a CPU core.

A process terminates when its final statement has been executed. As a process terminates, it’s resources such as memory are deallocated and reclaimed by the OS. Processes can also be terminated by other processes, for example, a parent process can terminate its children processes. A parent process can terminate its children for several reasons, such as a child process overconsuming resources, a child process is no longer needed or parent process has terminated, therefore its child must also be terminated.

Parallelism:

1. Task parallelism
2. Data parallelism

Multithreading: threads running in parallel

Mutltithreading models:

1. Many to one: Many user threads to one kernel thread. No concurrency, one thread at a time to use kernel, no multiprocessors
2. One to one
3. Many to many

Scheduling algorithms: assigns CPU time to threads and processes. Also which process in queue is allocated to CPU next

Program counter: activity status

Activation record

Process control block PCB

Process scheduling

* I/O bound processes
* CPU bound processes
* Scheduling queues: ready processes are stored in a linked list queue. IT i/O is needed, process moved to wait queue
* CPU scheduling: swapping when necessary. Swap process out to disk from RAM and then swap back in when ready
* Context switch: process context must be saved when process interrupted. Context switch: CPU switches from one process to another. Current process state saved

Interprocess communication (IPC)

* Independent and cooperating processes
* Why cooperation: Info sharing, modularity and comp speed up
* Mechanisms: memory sharing, message passing

IPC in shared memory

* Used in producer – consumer problems

IPC in message passing

* Proccesses do not share address space
* Operations: send and receive
* Message link direct implementation

Threads

* A thread belongs to a process
* Unit of CPU utilization