Assignment 3, Operating Systems

Jay R Bolton

November 7, 2011

Ch 3 problems: 3.1, 3.2, 3.4, 3.5, 3.10, 3.11, 3.12

3.1 **(1)** A process finishes and the OS selects a new one.

(2) Process has reached maximum allowable time for uninterrupted execution.

(3) Process initiates an IO action that it then depends on.

(4) A process gets back the result from IO.

(5) A parent process terminates a child process.

(6) The process is trapped with a fatal condition.

3.2 **(22)**

Process	**State**
P1	Blocked
P3	Blocked
P5	Ready/Suspend
P7	Blocked
P8	Running (?)

(37)

Process	**State**
P1	Ready
P3	Ready
P5	Blocked/Suspend
P7	Blocked
P8	Running (?)

(47)

Process	**State**
P1	Ready
P3	Ready
P5	Ready
P7	Blocked
P8	Running (?)

3.4 The diagram from 3.8 would be modified by adding swapped queues, one for Ready and one for Blocked. An edge from Ready labeled suspend

would lead to the ready swapped queue. An edge from the ready swapped queue to the ready queue would be labeled activate. An edge from each of the blocked queues would go to a corresponding swapped blocked queue with edges labeled suspend and activate.

3.5 The OS could define a priority offset. In other words, swapped jobs would have reduced priority by some offset. For example, if the offset is 5, a swapped job with priority 1 would wait for four ready jobs with priorities 1, 2, 3, and 4 before it is run. 5 might be a bit too extreme.

3.10 It's only useful if we assume that the original process will be the one to resume after the interrupt. But in general, the OS will probably have another process preempted, so the original process' context would need to be saved out anyway.

3.11 Real time applications (according to Wikipedia), must enforce deadlines. If processes in Kernel mode cannot be preempted, then a deadline cannot be enforced to kick it out.

3.12 Either the process ID of the child process or -1 for failure

I accidentally did the review problems first. I'll leave them here.

3.1 An instruction trace is a listing of machine instructions that a process has performed up until it was halted or the trace was read out.

3.2 New batch job, user log-in, OS service, or the branching off of an existing process.

3.4 Preemption of a process is its interruption in favor of another process with higher priority.

3.5 Swapping is the transfer of a process state from memory to disk. It allows more processes to be suspended than would fit into memory alone.

3.10 To protect the kernel from operations that too easily screw things up. Users generally would not need to mess with the OS at the instruction level and could too easily break it.

3.11 (p 136) Assign an id to the process, allocate space in memory for it, allocate the process control block, place process in its proper control list, and create any other necessary data structures.

3.12 Interrupts are caused by the clock, IO, or memory; nothing in the process itself causes it. Traps arise because of some error in the process itself.