Operating Systems: Chapter 6

1-15	priorities of Variable class in windows
16-31	priorities of real time class in windows
Aging	as time progresses, increase the priority of the waiting processes
Asymmetric multiprocessing	one processor accesses the systems data structures which eliminates the need for data sharing
convoy effect	short process behind long process will be waiting for the long process to finish
CPU Scheduler	Selects from processes in memory that are ready to execute and allocates to the CPU
deterministic modeling	takes a workload and defines the performance of each algorithm for that workload
deterministic modeling	Processes that run day to day vary, There is no static set of processes to use for ____,
fixed priority scheduling	do all from foreground, then all from background), possibility of starvation
hard affinity	Linux supports that
idle thread	If no thread is ready, dispatcher will execute a special thread
Linux Scheduling	Real-time are assigned value range 0-99, Normal tasks are assigned priorities from 100-139, CFS assigns a proportion of processing time to each normal task
Max CPU utilization	scheduling criteria: keep CPU as busy as possible
Max Throughput	number of processes that complete their execution per time unit
memory management	thread running at priority 0 in windows
Min Response time	amount of time it takes from when a request was submitted until the first response is produced, time it takes to start responding, not to output the response
Min Turnaround time	amount of time to execute a process
Min waiting time	amount of time a process has been waiting
multilevel queue	Ready queue is partitioned into separate queues: foreground (interactive) background (batch), Processes are permanently assigned to one queue, - Both queues use different scheduling algorithm
priority scheduling	Priority number is associated with each process, CPU is allocated to the process with the highest priority (smallest number)
process bouned to one queue	Multilevel feedback queue: Low scheduling overhead, but not flexible
Process can move between queues	Multilevel feedback queue: Flexible but very complex, aging can be implemented this way
processor affinity	process has affinity for processor on which it is currently running
response	when a request is submitted until the first response is produced
Round Robin	Each process gets a small unit of CPU time, after this time has elapsed, the process is pre-empted and added to the end of the ready queue
Shortest Job First	Associate each process to the length of its next CPU burst, use these lengths to schedule the process with the shortest time
Sort Affinity	OS attempts to keep a process running on the same processor but no guarantees
Starvation	low priority processes may never execute, waiting forever
Symmetric multiprocessing (SMP)	each processor is self-scheduling, all processes are in a common ready queue or each processor has its own private queue
time slice	each queue gets a certain amount of CPU time that it can schedule among its processes, EX: 80% for foreground, 20% for background
turnaround	submission of a process to the time of completion
windows scheduling	Priority based, pre-emptive scheduling, 32 level priority scheme to determine the order of thread execution