Name: Arthur Chen

UIN: 327003368

|  |  |  |
| --- | --- | --- |
|  | User-level threads | Kernel-level threads |
| To create | Easier and faster | Harder and slower |
| Can be run on any OS | Yes | No |
| Thread switching | No kernel mode privileges required | Required to switch to kernel mode to transfer control from one thread to another during a process |
| Blocking | The entire process is blocked if one of the user-level threads performs a blocking operation | While one of the threads is blocked, other threads can be scheduled on a different processor |
| Multi-threading | Not an advantage to use multiprocessing for user-level threads | Multiple threads of the same process can be scheduled on different processors |

The green color indicates which is better under each condition.

1. FCFS:

Unfavorite of short jobs since any short jobs right after long jobs will have to wait for a long time.

RR:

Since RR treats all jobs equally by giving them equal amount of time to use the CPU, short jobs will be able to leave faster than long jobs.

Multilevel feedback queues give higher priority for jobs which wait for a longer time. Therefore, this algorithm is quite similar to the RR, and the shorter jobs can leave the CPU faster.

1. In this code, fork() has being called 4 times. Therefore, by the end of the code, there are 2^4 processes in total, which is 16.
2. When fork() is called and a new child process is created, the copy of heap and stack will also be created. Therefore, heap and stack do not share between parent and child process, but and only the shared memory segment will be shared between them.