MTL458: Operating Systems

Midterm (2023)

Time: 2 Hr, Total marks: 30.

Instructions:

- Write your answers neatly and to the point.
- Remember that you will be graded on what you write and not what you intend to write.

Questions:

Q1 (a) Segmentation is a generalization of base-and-bounds. Which advantages does segmentation have as compared to base-and-bounds?

(b) List the types of events that cause a process to switch from user mode to kernel mode.

(c) Explain whether starvation is possible for the following scheduling policy (i) Shortest Job First, (ii) Round Robin, (iii) MLFQ. *(2+2+2=6)*

Q2 (a) Consider an operating system that uses a paging-based memory management scheme. Logical addresses are 32 bits long. Each page table entry occupies 4 bytes. Calculate the minimum and maximum page sizes for which exactly 3 levels of paging will be required.

(c) You are now given some new information about a particular system. Specifically, this system has 4 MB linear page table size (per process), and has a 1KB page size. Assuming page table entry size is 8 bytes, how many bits are in the virtual page number (VPN) on this system? *(4+4=8)*

Q3 Write a program to fork exactly 10 children. Each process should print its own pid and exit. (5)

Q4 Explain how many distinct output(s) the following code fragment can generate if the buffer is a character array containing the string "PQRST". Assume for this question that rand() generates a truly random, uniformly distributed positive integer.

(a) Code fragment 1:

```
.....
  fork();
  for (i = 0; i < 4; i++) {
      printf( "%c", buffer[i]);
      sleep(rand() % 5);
  }
....
```

(b) Code fragment 2

1 by

```
.....
 for(j=0;j<2;j++){
  fork();
 }
 for (i = 0; i < 4; i++) {
     printf( "%c", buffer[i]);
     sleep(rand() % 5);

 }
....
```

(3+4=7)

Q5 TLB misses can be nasty. The following code can cause a lot of TLB misses, depending on the the values of S and M. Assume that your system has a 32-entry TLB with a 8KB page size.

```
int value=0;
int data[M];

for(int j=0;j<1000;j++){
   for(int i=0;i<M; i+=S){
        value=value+data[i];
   }
}
```

What should you set M and S to so that you can achieve a TLB miss upon pretty much every access to the array "data"?

(4)