CS4650 – FALL 1999 – MIDTERM EXAM

Malek Mouhoub
(10/21/99)

Exercise 1

1. An I/O device interupts when it needs service. When an interrupt occurs, what are the different methods used by the operating system to determine which I/O device caused the interrupt ?
2. Why is job scheduling called *long-term scheduling*?
3. How does the job scheduler select the processes to be load into memory for execution ?
4. What are the different ways of communication between processes ?
5. What is a thread ? what are the differences between multi-treading and multi-processing ?
6. What advantage is there in having different time-quantum sizes on different levels of a multilevel queueing system ?

Exercice 2

Consider the following set of processes, with the length of the CPU-burst time given in milliseconds.

Process	*Burst time*	*Priority*
P_1	10	3
P_2	1	1
P_3	5	3
P_4	1	4
P_5	2	2

The processes are assumed to have arrived in the order $P_1,\ P_2,\ P_3,\ P_4,\ P_5$, all at time 0.

1. Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.
2. What is the turnaround time of each process for each of the scheduling algorithms in part a ?
3. What is the waiting time of each process for each of the scheduling algorithms in part a ?
4. Which of the schedules, in part a, results in the minimal average waiting time (over all processes) ?

Exercice 3

1. What are the requirements a solution to the critical section problem must satisfy ?
2. For each solution (program in C) presented in pages 2 and 3, check if the requirements are satisfied.

Solution 1

```
turn =1;
p1()
{
    for(;;) {
        while (turn ==2);
        Crit1();
        turn =2;
        Remainder1();
    }
}
```

```
p2()
{
    for(;;) {
        while (turn ==1);
        Crit2();
        turn =1;
        Remainder2();
    }
}
```

Solution 2

```
c1 = c2 =1;
p1()
{
    for(;;) {
        while (c2 ==0);
        c1 = 0;
        Crit1();
        c1 =1;
        Remainder1();
    }
}
```

```
p2()
{
    for(;;) {
        while (c1 ==0);
        c2 = 0;
        Crit2();
        c2 =1;
        Remainder2();
    }
}
```

Solution 3

```
c1 = c2 =1;
p1()
{
    for(;;) {
        c1 = 0;
        while (c2 ==0);
        Crit1();
        c1 =1;
        Remainder1();
    }
}
```

```
p2()
{
    for(;;) {
        c2 = 0;
        while (c1 ==0);
        Crit2();
        c2 =1;
        Remainder2();
    }
}
```

Solution 4

```
c1 = c2 =1;
turn = 1;
p1()
{
    for(;;) {
        c1 = 0;
        while (c2 ==0)
            if (turn == 2) {
                c1 = 1;
                while (turn == 2);
                c1 = 0;
    }
        Crit1();
        turn = 2;
        c1 =1;
        Remainder1();
    }
}
```

```
p2()
{
    for(;;) {
        c2 = 0;
        while (c1 ==0)
            if (turn == 1) {
                c1 = 1;
                while (turn == 1);
                c2 = 0;
    }
        Crit2();
        turn = 1;
        c2 =1;
        Remainder2();
    }
}
```