

# COMP 2432 Operating Systems

## Mid-term Answer

### 1. I/O and Operating Systems.

Slide 5, Lecture 2; Slide 27, Lecture 2.

### 2. CPU Scheduling.

SJF 1111111115533322222244444444							(a) SRT 1223332255222444444411111111						
Pid	Burst	Arr	Prior	Wait	TR		Pid	Burst	Arr	Prior	Wait	TR	
1	9	0	3	0	9		1	9	0	3	19	28	
2	7	1	2	13	20		2	7	1	2	5	12	
3	3	3	4	8	11		3	3	3	4	0	3	
4	7	5	1	16	23		4	7	5	1	8	15	
5	2	8	5	1	3		5	2	8	5	0	2	

FCFS 1111111112222222333444444455							(b) RR (q=3) 1112221113334442225511144424						
Pid	Burst	Arr	Prior	Wait	TR		Pid	Burst	Arr	Prior	Wait	TR	
1	9	0	3	0	9		1	9	0	3	14	23	
2	7	1	2	8	15		2	7	1	2	19	26	
3	3	3	4	13	16		3	3	3	4	6	9	
4	7	5	1	14	21		4	7	5	1	16	23	
5	2	8	5	18	20		5	2	8	5	10	12	

(c) For any non-preemptive scheduling algorithm, there will be 15 context switchings for 16 processes. That will generate an overhead of **15** time units. The total burst time for the 16 processes is **93**. Therefore, the final completion time when all processes complete execution is  $93+15=108$ .

### 3. Process Management.

Slide 5, Lecture 4; slide 32, Lecture 4; that special process is process **1** in Linux (**systemd** or **init**).

### 4. Memory Management.

Alg.	70 hole			100 hole			60 hole			Unfilled	Util.
FF	50	20		49	29	22	45	15		30	93.0%
BF	49	21		29	45	22	50	10		30	93.0%
WF	49	21		50	45	5	29	30	1	22	94.6%
Best	45	22	3	50	49	1	29	30	1	-	99.0%

(e) Frame size of 256 bytes means an offset of 8 bits. The first 4 bits for the logical address **010011010010** is the page number, i.e.  $0100_2 = 4$ . Frame number for page 4 is 10011. Therefore, the physical address in the main memory is **100111010010**, which is of 13 bits for a main memory of 8KB.

### 5. Programming with Processes.

There are **five** processes and **three** possible output sequences (below).

Sequence 1	Sequence 2	Sequence 3
bye welcome welcome	welcome bye welcome	welcome welcome bye

### 6. Shell Programming.

This is just a sample script. You could have your own way of programming.

```
#!/bin/bash
list=(`grep $1 trans*.txt | grep $2`)
count=(0 0 0 0 0 0 0 0 0 0 0 0)
i=4
while [ $i -lt ${#list[*]} ]; do
    if [ ${list[i]} == 'A+' ]; then let count[0]++
    elif [ ${list[i]} == 'A' ]; then let count[1]++
    elif [ ${list[i]} == 'A-' ]; then let count[2]++
    . . .
    elif [ ${list[i]} == 'D+' ]; then let count[9]++
    elif [ ${list[i]} == 'D' ]; then let count[10]++
    elif [ ${list[i]} == 'F' ]; then let count[11]++
    fi
    let i=i+5
done
echo $1 in $2
if [ ${count[0]} -gt 0 ]; then echo A+: ${count[0]}; fi
if [ ${count[1]} -gt 0 ]; then echo A: ${count[1]}; fi
if [ ${count[2]} -gt 0 ]; then echo A-: ${count[2]}; fi
. . .
if [ ${count[9]} -gt 0 ]; then echo D+: ${count[9]}; fi
if [ ${count[10]} -gt 0 ]; then echo D: ${count[10]}; fi
if [ ${count[11]} -gt 0 ]; then echo F: ${count[11]}; fi
if [ ${#list[*]} -eq 0 ]; then echo No student in subject ; fi
```