CS342 Operating Systems

Homework 2

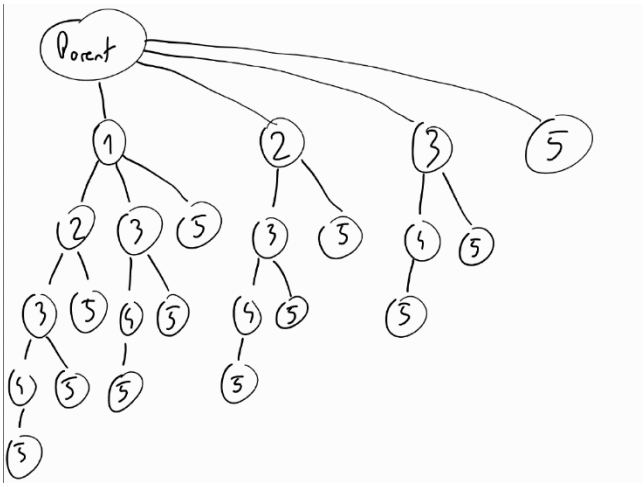
Ata Seren

21901575

Section-2

1)

```
main() {
        fork()  1
        fork()  2
        if (fork() == 0) 3
                fork()  4
        fork()  5
}
```



I gave a number to each fork() to name the process they create on the tree to follow them easily. 23 child processes have been created according to the tree.
If we consider the parent process too, 24 process have been created in total.

2) Speedup <= $\frac{1}{S+\left(\frac{1-S}{N}\right)}$ which N = number of processing cores

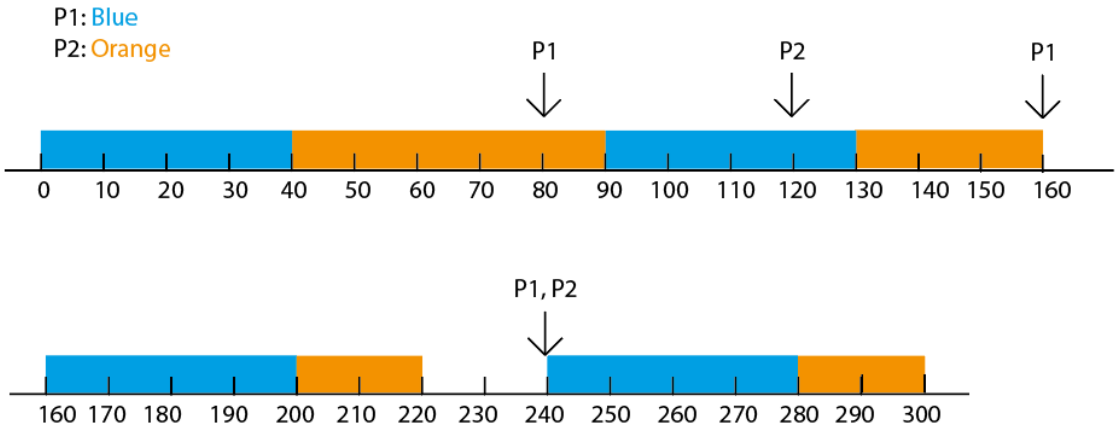
When N = 8, $\frac{1}{S+\left(\frac{1-S}{8}\right)} = 4$. According to this, S = 1/7

When N = 16, $\frac{1}{\frac{1}{7}+\left(\frac{1-\frac{1}{7}}{16}\right)} = \frac{56}{11} \cong 5.09$

Speedup <= 5.09

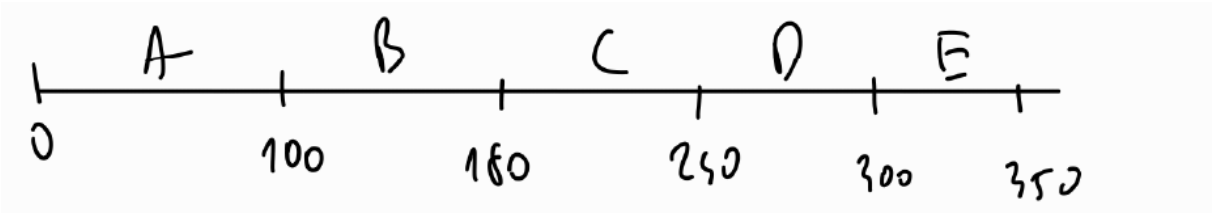
3) One 10, one 20, one 30 and two 50 will be printed. Their order can change if parent is processed faster than child or vice versa.

4)



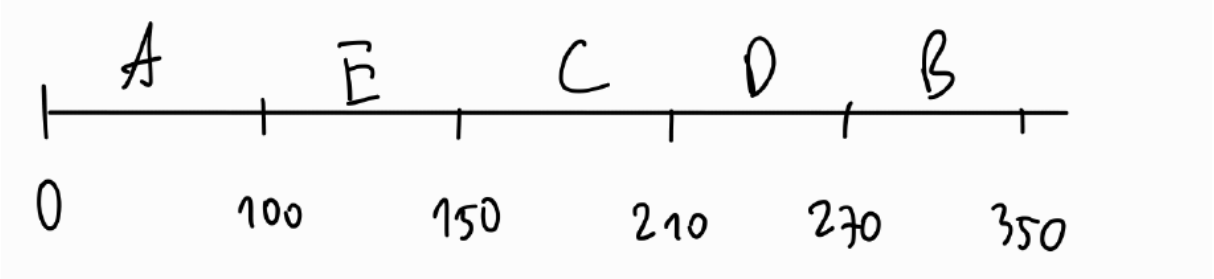
5)

For FCFS:



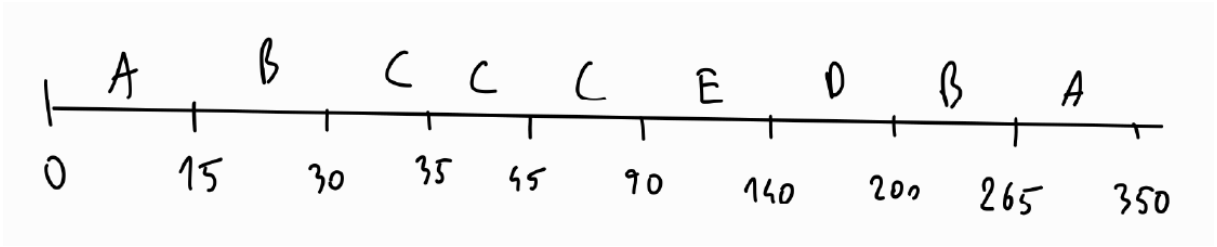
	Arrival	CPU Burst	Finish	Turnaround	Waiting
A	0	100	100	100	0
B	15	80	180	165	85
C	30	60	240	210	150
D	35	60	300	265	205
E	45	50	350	305	255

For SJF:



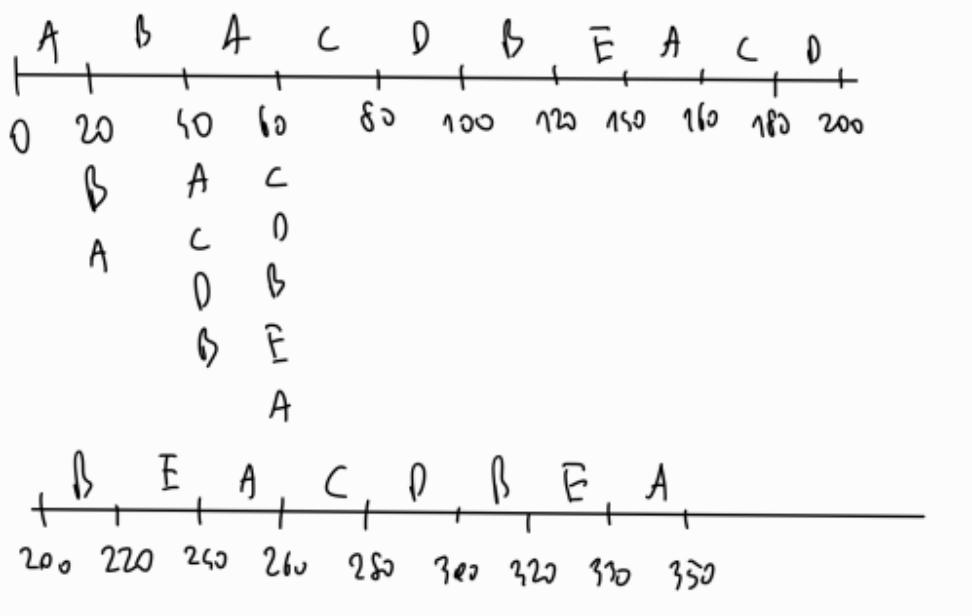
	Arrival	CPU Burst	Finish	Turnaround	Waiting
A	0	100	100	100	0
B	15	80	350	335	255
C	30	60	210	180	120
D	35	60	270	235	175
E	45	50	150	105	55

For SRTF:



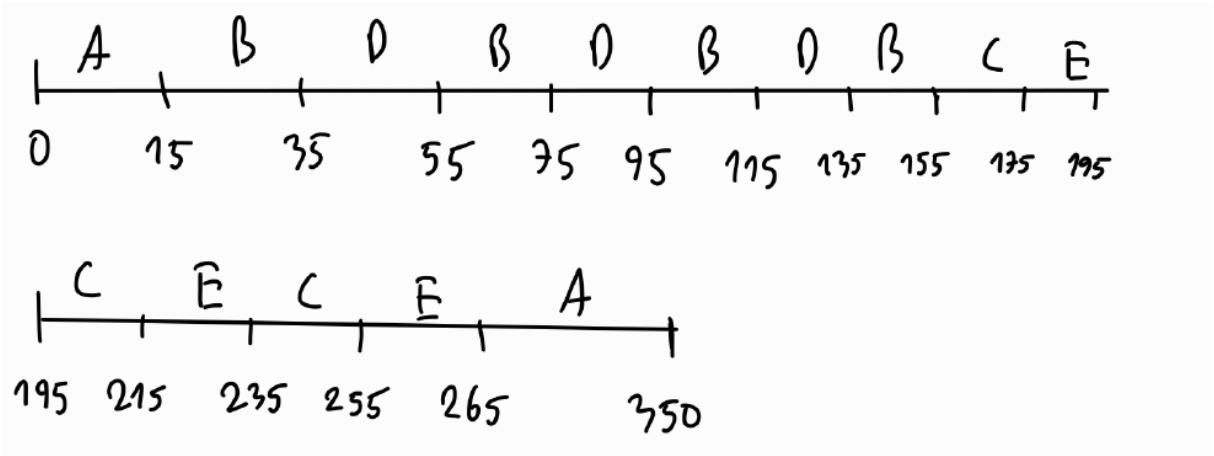
	Arrival	CPU Burst	Finish	Turnaround	Waiting
A	0	100	350	350	250
B	15	80	265	250	170
C	30	60	90	60	0
D	35	60	200	165	105
E	45	50	140	95	45

For RR(q = 20):



	Arrival	CPU Burst	Finish	Turnaround	Waiting
A	0	100	350	350	250
B	15	80	320	305	225
C	30	60	280	250	190
D	35	60	300	265	205
E	45	50	330	285	235

For preemptive priority scheduling with RR(q=20) applied on equal priority:



	Arrival	CPU Burst	Finish	Turnaround	Waiting
A	0	100	350	350	250
B	15	80	155	140	60
C	30	60	255	225	165
D	35	60	135	100	40
E	45	50	265	220	170

6) For process A:

$T_{n+1} = \alpha * t_n + (1 - \alpha) * T_n$
$\alpha = 0.8 \quad T_0 = 10$ ms

$T_1 = 0.8 * 20 + 0.2 * 10 = 18$
$T_2 = 0.8 * 10 + 0.2 * 18 = 11.6$
$T_3 = 0.8 * 40 + 0.2 * 11.6 = 34.32$
$T_4 = 0.8 * 30 + 0.2 * 34.32 = 30.864$
$T_5 = 0.8 * 20 + 0.2 * 30.864 = 22.1728$
Predicted value for process A after the mentioned bursts were executed is 22.1728 ms.

For process B:

$T_1 = 0.8 * 30 + 0.2 * 10 = 26$
$T_2 = 0.8 * 20 + 0.2 * 26 = 21.2$
$T_3 = 0.8 * 40 + 0.2 * 21.2 = 36.24$
$T_4 = 0.8 * 10 + 0.2 * 36.24 = 15.248$
Predicted value for process B after the mentioned bursts were executed is 15.248 ms.

If process A and B have arrived at the same time or in the queue at the same time, B will be picked because its predicated CPU burst time is less than A's.

7) Function that will be run by the thread will print 1000 which is value of x after initialized and incremented in the function, 2200 which is value of global variable y and 300 which is value of z transferred to the function as an argument during thread creation.

   In main(), 100 which is value of global variable x, 2200 which is value of global variable y and 300 which is value of global variable z are printed.

   Output:
   1000 2200 300
   100   2200 300