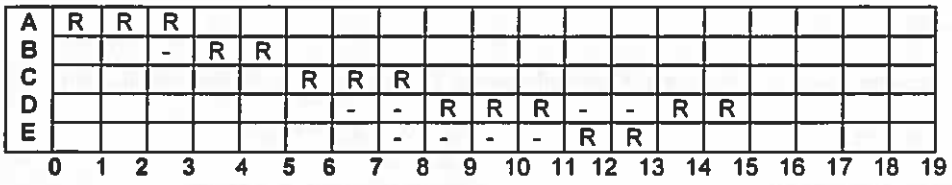
E1.9 – section B: Operating Systems- - Sample model answers to exam questions 2014

Question 1

(a) *New computed example*

[R= Running, - = waiting]

Round Robin with time slice of 3 ms



AWT= (0+1+0+4+4)/5= 1.8ms; ATT= 24/5 = 4.8ms

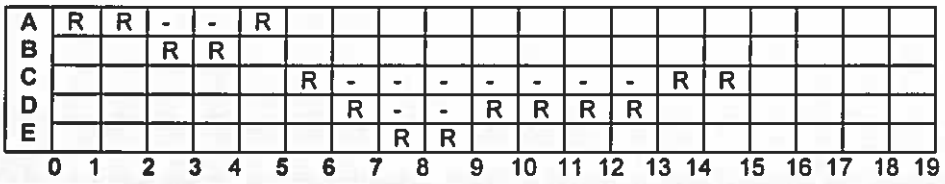
Advantages: Simple to implement and fair

Disadvantages: Difficult to determine appropriate time quantum:

- Too small: good response time, but large overheads (scheduler is called too often)
- Too large: bad response time

[3] [2 for correct solution, + 1 for adv/disadvantages]

Priority scheduling with pre-emption



AWT=10/5= 2ms ATT= 26/5 = 5.2ms

Advantages:

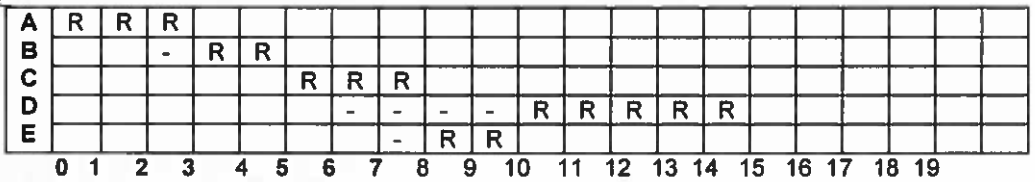
Takes into account external factors regarding the importance of the various processes

Disadvantages:

Might result (quite probable) in starvation of low-priority processes – this can be avoided using an *aging* procedure; processes that wait for too long have their priorities gradually increased.

[2] [2; 1 for correct solution, + 1 for adv/disadvantages]

SJF



Avg waiting time: (0+1+0+4+1) / 5 = 1.2 ms, Avg turnaround time: 22/5 = 4.4 ms

[2; 1 for correct solution, + 1 for adv/disadvantages]

Advantages: Allows new short jobs to get a good service; Good handling of interactive processes since it results in a short response time

Disadvantages: Knowing the length of next CPU burst is difficult; frequently this is predicted utilising previous lengths as estimates, or is user-specified; can result in long waits (including starvation) for long processes.

(b) [new computed example]

Optimal page replacement algorithm (7 page faults)

	1	4	5	6	9	4	1	5	4	7
Frame1	1	1	1	1	1			5		7
Frame2	-	4	4	4	4			4		4
Frame3	-	-	5	6	9			9		9

There are multiple correct ways of doing the last two replacements [2]

FIFO page replacement algorithm (9 page faults)

	1	4	5	6	9	4	1	5	4	7
Frame1	1	1	1	6	6	6	1	1		1
Frame2	-	4	4	4	9	9	9	5		5
Frame3	-	-	5	5	5	4	4	4		7

[2]

LRU (Least recently used) page replacement algorithm (9 page faults)

	1	4	5	6	9	4	1	5	4	7
Frame1	1	1	1	6	6	6	1	1		7
Frame2	-	4	4	4	9	9	9	5		5
Frame3	-	-	5	5	5	4	4	4		4

[2]

(c) [Bookwork]

Turn is a character variable; Interested_A and Interested_B are boolean variables initially set to FALSE;

```
Interested_A = TRUE;
Turn = 'B';
while (interested_B = TRUE
        AND Turn = 'B')
              Do_nothing;
Critical_A;
Interested_A = FALSE;
```

```
Interested_B = TRUE;
Turn = 'A';
while (interested_A = TRUE
    AND Turn = 'A')
          Do_nothing;
Critical_B;
interested_B = FALSE;
```

[1 point for setting the interest variables, 1 point for setting the turn variables, 1 point for setting the loop conditions correctly and 1 point for setting the interest variables to false after [4]

(d) [Bookwork]

First-fit: allocate first memory section that is big enough. Advantages: fast allocation method. Disadvantages: can be very inefficient

Best-fit: allocate the smallest section that is enough. Advantages: less inefficient in terms of space that first-fit. Disadvantages: tends to produce lots of remaining tiny fragments, and requires search through the entire list of memory sections

Worst fit: allocate the largest section that is available. Advantages: after allocating the request, the remainder of that section might still be usable. Disadvantages: requires search through the entire list of sections

[1 point per correct algorithm, and its advantages/disadvantages] **[3]**