Assignment 2

U21546216

Scott Bebington

Task 1

1.1) B
1.2) A
1.3) B
1.4) C
1.5) D

Task 2

2.1) 10 units

2.2) 38 units

2.3)

P1: Ready state time – 24 units

P2: Ready state time – 18 units

P3: Ready state time – 18 units

P4: Ready state time – 21 units

Average waiting time = (P1's waiting time + P2's waiting time + P3's waiting time + P4's waiting time) /4

Average waiting time = (24 units + 18 units + 18 units + 21 units)/4

Average waiting time = 20.25 units waiting time

2.4)

The total execution time will have to increase to allow for an I/O device to get execution time for P3.

P3 would have to use its running time for I/O device time and move and run after the allocated 38 time units.

Or if no extra execution time is given, P3 would have a shorter run time to allow for the use of an I/O device.

2.5)

P1: Turnaround time = Ready time + Running time = 24 units Ready time + 11 units Running time = 35 time units turnaround time.

P2: Turnaround time = Ready time + Running time = 18 units Ready time + 8 units Running time = 26 time units turnaround time.

P3: Turnaround time = Ready time + Running time = 18 units Ready time + 4 units Running time = 22 time units turnaround time.

P4: Turnaround time = Ready time + Running time = 21 units Ready time + 5 units Running time = 26 time units turnaround time.

Task 3

3.1)

Blocked/Suspended is suspended by the OS and waiting for an external event to happen such as a user input whereas Ready/Suspended is Suspended by the OS but is only waiting for the OS to run the program.

3.2)

Other OS reason: The OS might suspend a process is it thinks the process is going to cause a problem.

3.3)

The process may be awaiting an external I/O request from the user and cannot continue without it causing the process to be blocked.

3.4)

If there are more than 1 process in a queue and the OS admits a new process it will go into the queue but cannot be executed yet as there are other processes that need to be executed first.

Task 4

inherently serial code = 46% = 0.46

speedup from 1 core to 4 cores = 1/ ((1 – f) + (f / n))

= 1/ ((1 – 0.46) + (0.46 / 4))

= 1.527 (I used the unrounded version for the final calculation)

speedup from 1 core to 12 cores = 1/ ((1 – f) + (f / n))

= 1/ ((1 – 0.46) + (0.46 / 12))

= 1.729 (I used the unrounded version for the final calculation)

Relative speedup = speedup from 1 core to 12 cores – speedup from 1 core to 4 cores

= 1.729 – 1.527

= 0.202

Speedup factor from 4 cores to 12 cores is equal to 0.20239 or 20.239%