

SIT315 M1.S2P: Real-Time Systems – Part 2

Codey Funston | S222250824

No Group

Activity 1 – Process Memory Segments

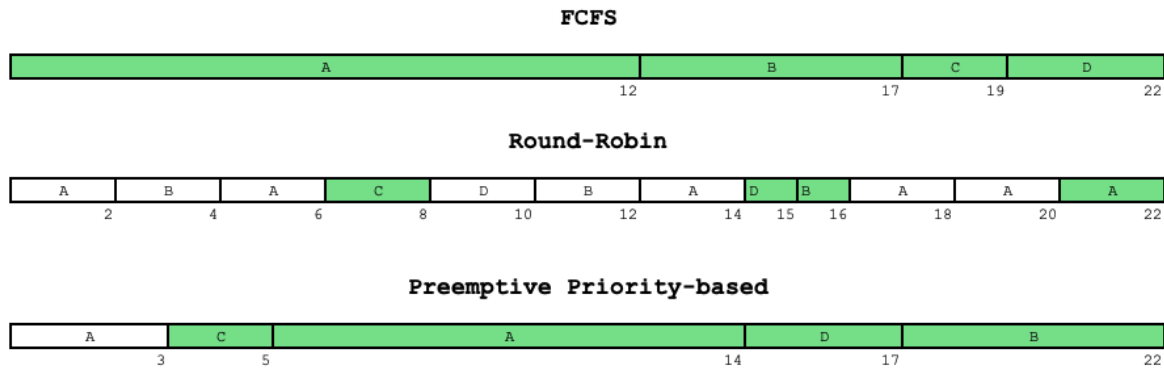
1. In the text segment there is the compiled code. There will be compiled code for the `sum()` function, `main()` function, and from the `iostream` library there will be compiled code for `cin` and `cout`.
2. The data segment is used for global variables and static local variables. It will include `my_size` as an initialised to 4 global variable, and `result` as an initialised to 0 static variable.
3. When line 12 is executed for the third time there will be the integer `n` stored in the stack as this isn't a pass by reference parameter and it is local to the `sum()` function, it will have value 8. There will also be the local variables in `main()` which consists of the pointer `p`, incrementor `i` with a value of 2, and `total` with a value of 17. The heap will have `p` which is an integer array that was dynamically allocated with the "new" keyword, it will have the first three user input numbers 4, 5, and 8.
4. When line 33 is being executed (only once) there will no longer be the array pointed to by `p` since it was deleted in the previous statement. However, the pointer `p` is still in the stack as there is no need to delete it as its lifetime corresponds to the `main()` function. The other variables in the stack will be `i` and `total`, with values of 4 and 20 respectively.

Activity 2 – Interrupts

The main problem in the code is that it does not account for debouncing. To fix this I included a time delay check like in my other code projects in this module. It prevents multiple alerts from the button cancelling out the user's intention. Also, the delay is reduced, and it results in the ability for the user to hold down the button to have it lit up and depress the button and have it switched off. The switch now behaves like a simple raised/lowered switch in a basic electric circuit.

Activity 3 – Process Scheduling

1. Gantt Charts



2. Individual and average waiting times (question 2 and 3)

The waiting time for a process is *finish* – *arrival* – *burst*.

FCFS:

$$A = 12 - 0 - 12 = 0$$

$$B = 17 - 1 - 5 = 11$$

$$C = 19 - 3 - 2 = 14$$

$$D = 22 - 4 - 3 = 15$$

$$\text{Average} = (0 + 11 + 14 + 15) / 4 = \mathbf{7.5}$$

Round-Robin:

$$A = 22 - 0 - 12 = 10$$

$$B = 16 - 1 - 5 = 10$$

$$C = 8 - 3 - 2 = 3$$

$$D = 15 - 4 - 3 = 8$$

$$\text{Average} = (10 + 10 + 3 + 8) / 4 = \mathbf{7.75}$$

Pre-emptive Priority-based:

$$A = 14 - 0 - 12 = 2$$

$$B = 22 - 1 - 5 = 16$$

$$C = 5 - 3 - 2 = 0$$

$$D = 17 - 4 - 3 = 10$$

$$\text{Average} = (2 + 16 + 0 + 10) / 4 = \mathbf{7}$$

4. The shortest remaining time first (SRTF) selects the next process to run based on how long each process has left.

$$A = 22 - 0 - 12 = 10$$

$$B = 8 - 1 - 5 = 2$$

$$C = 5 - 3 - 2 = 0$$

$$D = 11 - 4 - 3 = 4$$

$$\text{Average} = (10 + 2 + 0 + 4) / 4 = \mathbf{4} > \{7.5, 7.75, 7\}$$

