**Assignment II**

**Problem Bank 60**

**Assignment Description:**

The assignment aims to provide deeper understanding of Pipelining Architecture, Scheduling and Multithreading using CPU- OS Simulator. The assignment has three parts.

* Part I deals with Pipeline Architecture
* Part II deals with scheduling algorithm(FCFS, RR)
* Part III deals with Multithreading

**Submission:**

You will have to submit this documentation file and the name of the file should be GROUP-NUMBER.pdf. For Example, if your group number is 1, then the file name should be GROUP-1.pdf.

Submit the assignment by **9th August 2020, through canvas only**. File submitted by any means outside CANVAS will not be accepted and marked.

In case of any issues, please drop an email to the course TAs, Ms. Michelle Gonsalves

([michelle.gonsalves@wilp.bits-pilani.ac.in](mailto:michelle.gonsalves@wilp.bits-pilani.ac.in)).

**Caution!!!**

1. Assignments are designed for individual groups which may look similar and you may not notice minor changes in the assignments. Hence, refrain from copying or sharing documents with others. Any evidence of such practice will attract severe penalty.
2. **Marks will not be awarded for individual submissions**

**Evaluation:**

* The assignment carries 12 marks
* Grading will depend on
  + Contribution of each student in the implementation of the assignment
  + **Plagiarism or copying will result in -12 marks**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FILL IN THE DETAILS GIVEN BELOW\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Assignment Set Number:**

**Group Name:**

**Contribution Table:**

**Contribution** (This table should contain the list of all the students in the group. Clearly mention each student’s contribution towards the assignment. Mention “No Contribution” in cases applicable. If the contribution is equal the write 100%)

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Name (as appears in Canvas)** | **ID NO** | **Contribution (%)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Resource for Part I, II and III:**

* Use following link to login to “eLearn” portal.
  + <https://elearn.bits-pilani.ac.in>
* Click on “My Virtual Lab – CSIS”
* Using your canvas credentials login in to Virtual lab
* In “BITS Pilani” Virtual lab click on “Resources”. Click on “Computer Organization and software systems” course. Refer to Lab Capsule 4, Lab Capsule 5, Lab Capsule 6.

**Part I: Pipeline Processor**

Consider the following program:

program pipeline1

x=5

y=2

z=10

z=z + 1

y=y - 1

z=z + x

x=x + y

z= z + 1

end

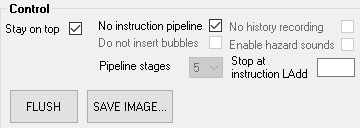
Compile the code and load it in CPU-OS simulator. Perform the following:

**Execute the above program using non-pipelined processor and pipelined processor and answer the following questions.**

**Note: Every time flush the pipeline before running the code**

1. **Non-pipelined Processor:**

To enable non-pipelined processor, check “No instruction pipeline” check box in control panel.



1. How many stages are there in non-pipelined processor? List them

|  |
| --- |
|  |

1. Fill in the following after executing of above program using non-pipelined processor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clocks | Instruction Count | CPI | Speed up Factor |
| Non Pipelined processor |  |  |  |  |

1. What are the contents of General purpose registers after the execution of the program?

|  |
| --- |
|  |

1. **Pipelined processor:**

To use, enable pipelined processor, uncheck “No instruction pipeline” check box in control panel.

1. Fill in the following table with respect to pipelined processor execution of the above program:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pipelined processor conditions | Clocks | Instruction Count | CPI | Speed up Factor | Data hazard  (Yes/No) | Contents of registers used by the program |
| Check “Do not insert bubbles” check box |  |  |  |  |  |  |
| Uncheck “Do not insert bubbles” |  |  |  |  |  |  |

1. Is there a way to improve the CPI and Speed up factor? If so give the solution.

|  |
| --- |
| Solution: |

Fill in the following table for the above solution.

|  |  |  |  |
| --- | --- | --- | --- |
| Clocks | Instruction Count | CPI | Speed up Factor |
|  |  |  |  |

**Part II: Process Scheduling**

Consider the following 4 source codes:

**Source Code 1:**

program My\_Pgm1

i = 2

for n = 1 to 10

x = i + n

next

end

**Source Code 2:**

program My\_Pgm2

i = 10

for n = 1 to 8

x = i + n

next

end

**Source Code 3**:

program My\_Pgm3

i = 5

for n = 1 to 12

x = i + n

i = i + 2

n = n + 2

next

end

**Source Code 4**:

program My\_Pgm4

i = 10

for n = 1 to 5

x = i + n

next

end

Create 4 processes P1, P2. P3 and P4 from source codes 1, 2, 3and 4 respectively with following properties. Fill up the following table:

|  |  |  |
| --- | --- | --- |
| Scheduling Algorithm : FCFS | | |
| Process | Arrival Time | Waiting time |
| P1 | 0 |  |
| P2 | 2 |  |
| P3 | 5 |  |
| P4 | 4 |  |
| Average waiting time | |  |
| Scheduling Algorithm : Round Robin with time quantum 5 | | |
| Process | Arrival Time | Waiting time |
| P1 | 0 |  |
| P2 | 2 |  |
| P3 | 5 |  |
| P4 | 4 |  |
| Average waiting time | |  |
| Scheduling Algorithm : Round Robin with time quantum 10 | | |
| Process | Arrival Time | Waiting time |
| P1 | 0 |  |
| P2 | 2 |  |
| P3 | 5 |  |
| P4 | 4 |  |
| Average waiting time | |  |
| Out of three cases, which one is better and why? | | |

**Part III: Multi-Threading**

program ThreadTest

total = 0

sub thread1 as thread

for i = 1 to 5

total = total +i

next

end sub

sub thread2 as thread

for i = 5 to 10

total = total - i

call thread1

next

end sub

sub thread3 as thread

call thread2

for i = 10 to 15

total = total + i

next

end sub

sub thread4 as thread

call thread3

for i = 15 to 20

total = total - i

next

end sub

call thread4

wait

writeln (“Total =”, total)

end

Compile the above source code and load it in the main memory. Create a single process, choose RR scheduling algorithm with time quantum of 3 ticks. Run the Process.

Answer the following questions:

1. What is the value of “Total” ?
2. How many processes and how many threads are created?
3. Identify the name of the processes and threads.
4. What is the PID and PPID of the processes and threads created?
5. Represent the parent and child relationship using tree representation