Assignment 3 report

Real-time Operating System - 48450

Student Name: Smit Patel

Student ID: 11819164

**Table of Contents**

[I. Introduction 3](#_Toc483688347)

[II. Theory of operation 3](#_Toc483688348)

[III. Observation of Prg\_1 3](#_Toc483688349)

[IV. Observation of Prg\_2 5](#_Toc483688350)

[V. References 9](#_Toc483688351)

1. Introduction

This assignment involves some program development for memory management and signals as well as CPU scheduling. We were required to create a program that applies few concepts of Real Time Operating System. These programs are implemented In C language.   
  
The first program is little bit complex due to Threads and FIFO involvement. The basic idea of this task is to understand the working of FIFO between multiple threads. on top of that one of the CPU scheduling Round Robin has to be performed to work out the average wait time and turn-around time from the given data.  
Second program is little bit simpler compared to the first one. This has memory management and signals by simulating page replacement for virtual memory management with first in first out algorithm. This management is very important in operating system being efficient and low consumption of physical memory.

1. Theory of operation

The Round Robin algorithms generally uses time slices or time quantum. It is a length of time which are assigned to each process in equal portions and in circular order, to handle all the processes regardless of priority.it is starvation free scheduling so the processor is never idle thus efficiency increases. This algorithm can also be using in data packet scheduling in networks.

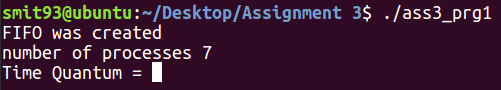
There are many Operating system scheduling algorithms like First-Come-First-Serve, Shortest job Nest, Priority etc. but Round Robin is the eldest simplest, fairest and most widely used algorithm. It is pre-emptive so it is effective oi time sharing environment in which the system needs to guarantee reasonable response for interactive users.

Page replacement algorithm is one of the memory management algorithm. Page replacement have many algorithms but in this assignment FIFO page replacement needed to be implemented. This manages linked list of all pages while maintaining order in which they enter into memory. Page at the from is replaces so it is very easy to implement but sometimes it costs very often used page from the memory so it has to be replaced again with whatever the first page is.

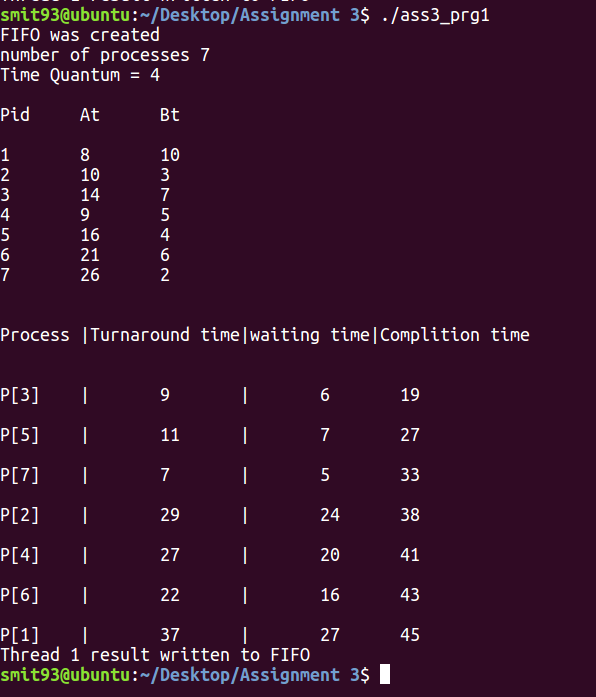
1. Observation of Prg\_1

The primary task of this program was to create a FIFO and share in between 2 different threads which has different tasks to perform. I commence to the code from the demo code provided on UTS online. First of all I created threads. Along with the fifo by feeding it by dummy data. Create() and put() are the two methods in which threads are being created. FIFO is opened twice in thread! One on read mode to check if there is existing FIFO in memory and write to write it inside the fifo. Which was allotted to writeinfifo() which was called inside Thread1 as requested in assignment specification. I also put file creation in thread2 in the beginning itself so it is easy to implement once the round robin subroutine is written. It has dummy name to make sure everything is working fine.

RoundRobin() method has the primary logic of the task which is CPU scheduling algorithm Round Robin.it starts with the structure of a process which contains Process Id , Arrival time of the process, burst time to know how long process will take to execute. Waiting time will initially be 0 and determined at the end of round robin routine. Same with turn around time which will be known after completion time is calculated because turn around time is completion time subtracted by arrival time of particular process . the most important part of the round robin algorithm is time quantum or time slice which is provided with console input by the use following is the screenshot of the output where it asks for time quantum to calculate the values.



This is where user puts the value for quantum. Above that request program shows the number of processes it is going to be calculated with round robin. And on top of everything a message indicates that FIFO has been created and ready to take values from thread1. Now program checks the arrival time of every process and then arrange it in ascending order so they can be served. Below is the output of this program which will give all the remaining values from the structure and calculate the average values of turn around time and waiting time. Then it will be written in the final output.txt file created with file creation code written in advance.

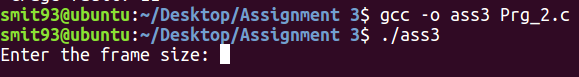


1. Observation of Prg\_2

Second program in this assignment is about memory management and signals. This simulates page replacement for virtual memory management by the FIFO algorithm. This page segment have 4 frame in it which will also have to entered by the user in the command line argument.

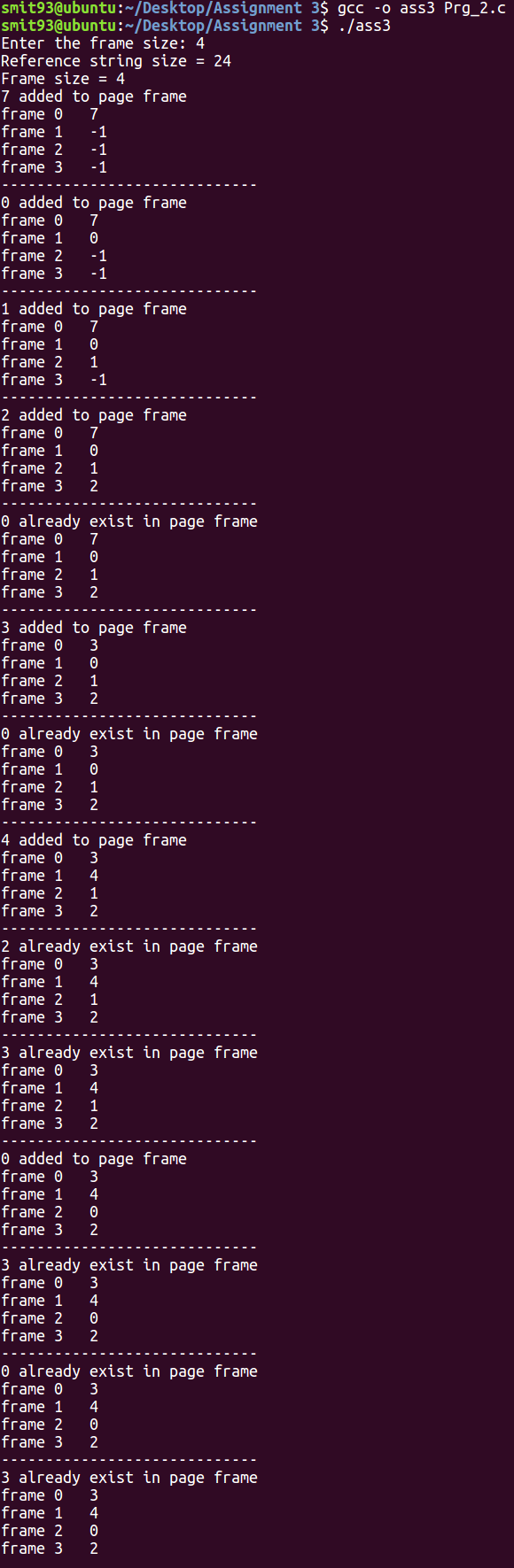
**7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1,7,5**

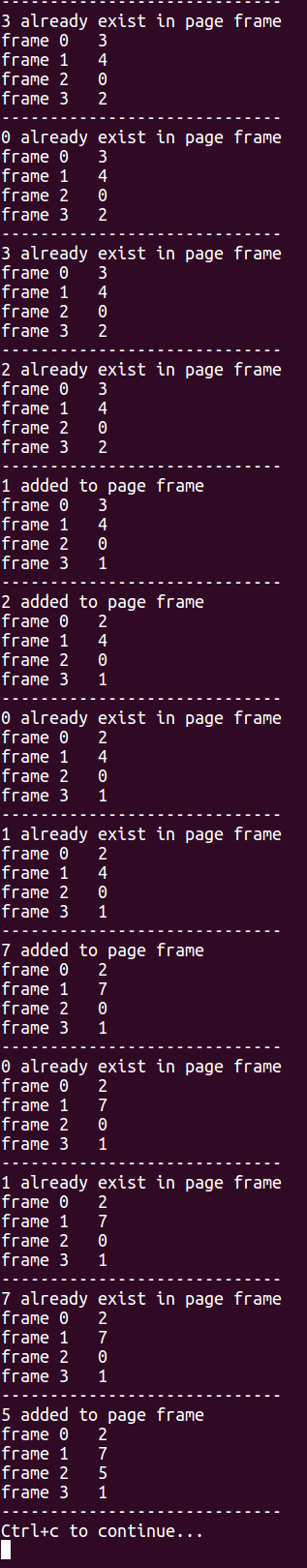
are the values or reference string given in assignment specification. Now the programme is expected to check if there is a page fault and output the current state of the frame as well this is the first message appears when program is run



It asks for size of the frame which is 4 for this particular problem but it can also calculate for different number of frames for the same reference string.

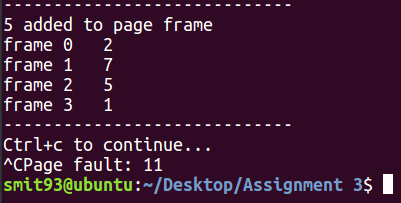
The structure frame\_t has page and its count to keep track of faults and outputs .first of all number of pages inn reference string is measured by simple for loop. Then main loop starts and it counts 0 to 3 due to number of frames for this problem. Then the internal loop runs exactly as number of processes in that it checks the state of page frames and prints the status after every number fed to loop from reference string.  
following is the output of how this program handles the pages its faults and output.





After the last number from the string memory management method stops and wait for a particular signal which was specified in the assignment specification which is Ctrl + C. so program waits for that specific input. It is one of the control signals in the terminal. The signal handler is predefined process in signal.h header file.

After the signal is received number of page fault are presents in the consol. Below is the screenshot of it



And then program is terminated and terminal is ready for next command.

1. References

If you have references, please referee it correctly.

A. Silberschatz, P. B. Galvin & G. Gagne, 2012, Operating System Concepts, 9 th edn, John Wiley & Sons, New York.