**CS3103 Project B Report**

**Group Information**

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Name | SID | Email |
| #1 (contact person) |  |  |  |
| #2 | Mahir Labib | 56749556 | mlabib2-c@my.cityu.edu.hk |
| #3 |  |  |  |

Note: Please specify each team member's contribution if not all members make significant contributions to this project.

**Problem 1**

1) Have you successfully implemented the priority scheduling into the basekernel? Have your test programs executed properly? If not, please provide potential reasons for the issues.

Yes, we have implemented the priority scheduling into the base kernel and have properly tested out the programs and confirmed that they are executing properly.

**2) Abstract idea and mechanism design.**

3) Implemented functions.

1. **Schedulertest.c:** This code **automates** the creation of processes from specified executable files, assigning each a priority. It **opens each file**, **tries to run it as a process with the given priority**, and **reports success or error**. After setting up these processes, it initiates their execution.
   1. First, it prints the name of the process & forks a new process using **‘syscall\_process\_fork’.**
      1. **If it is the child process,** it runs the **function runForSeconds(int seconds**) is implemented with a different time for each process, i.e. ‘process5’ waits for 4 seconds, ‘process4’ waits for 1 second and so on.
      2. **If it is in the parent process,** it prints the child’s PID and waits for the child process to finish with the **“syscall\_process\_wait”**
2. **Makefile**
   1. In the Makefile, we have added additional executable targets to the “**USER\_PROGRAMS**” variables, as these are the new user programs that we have developed and added to the build process. and they are as follows. ***‘process1.exe’, ‘process2.exe’, ‘process1.exe’, ‘process2.exe’, ‘process3.exe’, ‘process4.exe’, ‘process5.exe’, ‘schedulertest.exe’, ‘named\_pipe\_test.exe’.***
3. **Syscall\_handler.c**
   1. **Addition of New System Calls:**
      1. **sys\_process\_prun**: Added to handle process creation with a specific priority, extending the functionalities of process management with priority considerations.  
         A computer screen with white text

         Description automatically generated
   2. **Changes in existing functions** 
      1. Enhanced error handling in ‘sys\_process\_run”  
         A computer code on a black background

         Description automatically generated
   3. **Prioritizing process creation in ‘sys\_process\_prun”**

A black background with white text

Description automatically generated

* 1. **Additional Error Checking**

**A black screen with white text

Description automatically generated**

1. **runForSeconds() function** 
   1. It is almost the same as it is provided in the question. The **runForSeconds** function waits for a specified number of seconds using a **do-while** loop. It repeatedly checks the system time until the elapsed time matches the input parameter **seconds**.
   2. **list.c file** 
      1. **print\_list Function:** We introduced a function to print the priorities of the
      2. nodes in the list.
      3. **List\_push\_head\_priority function**: Allows pushing a node to the head of the list with a specified priority. This adds functionality that was not present in the older snippet.  
         A computer screen with text and symbols

         Description automatically generated
      4. **Change in Priority Ordering Logic:** In the older snippet (list\_push\_priority function), the condition for inserting based on priority was **pri > n🡪 priority**. In the newer snippet, this has been corrected to pri < n-> priority, implying the list maintains a sort order from lowest to highest priority, which was mentioned to be done in the question.  
         Before:   
         A black background with white text

         Description automatically generated  
         After:   
         A black background with white text

         Description automatically generated
2. **Process.c file:** Here, to accomodatre with the change from FIFO scheduling to priority scheduling, we needed to change the implementations of some of these files.
   1. Change from **blocked\_list** to **ready\_list** for the priority launch

**Before:**

**A black background with white text

Description automatically generated**

**After:**

**A black background with white text

Description automatically generated**

* 1. **Process creation and Management**

We introduced priority during the process creation  
A black screen with white text

Description automatically generated

**Problem 2**

1) Have you successfully implemented the named pipe into the basekernel? Have your test programs executed properly? If not, please provide potential reasons for the issues.

2) Abstract idea and mechanism design.

3) Implemented functions.