Project 2

- **110** points
- **Description:**
  -
    - Write a new system call int my_set_process_priority(int x) so that a process *P* can use this new system call my_set_process_priority(int x) to set the priority of the process as x **every time** when a context switch (i.e. process switch) transfers CPU to process *P*.
    - The return value of this system call is either 0 or 1.
    - 0 means that an error occurs when executing this system call.
    - 1 means the system call completes successfully.
    - Values of parameter x should be between 101 and 139. Other parameter values result in an error. In other word, the system call returns 0 when the system call receives an error parameter value.
    - function rand()
      - functionality: pseudo-random number generator
      - prototype of function: int rand(void)
      - head file: #include <stdlib.h>
  -
    ```
    //prototype of the new system call is as follows:

    int my_set_process_priority(int x)
    ```
  - Write and execute the following program to show how the priority of a process influences the execution time of a process.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <time.h>
#include <sys/time.h>


#define TOTAL_ITERATION_NUM  100000000
#define NUM_OF_PRIORITIES_TESTED  40



main()
{
  int          index=0;
  int          priority,i;
  struct timeval   start[NUM_OF_PRIORITIES_TESTED], end[NUM_OF_PRIORITIES_TESTED];

  gettimeofday(&start[index], NULL);         //begin
  for(i=1;i<=TOTAL_ITERATION_NUM;i++)
    rand();
  gettimeofday(&end[index], NULL);           //end

  /*============================================================================

  for(index=1, priority=101;priority<=139;++priority,++index)
  {

    if(!my_set_process_priority(priority))
      printf("Cannot set priority %d.\n", priority);

    gettimeofday(&start[index], NULL);         //begin
    for(i=1;i<=TOTAL_ITERATION_NUM;i++)
       rand();
    gettimeofday(&end[index], NULL);           //end
  }
```

```
   /*==============================================================

   printf("The process spent %ld uses to execute when priority is not adjusted.\n",
    ((end[0].tv_sec * 1000000 + end[0].tv_usec) - (start[0].tv_sec * 1000000 + start[0].tv_usec)));

   for(i=1;i<=NUM_OF_PRIORITIES_TESTED-1;i++)
     printf("The process spent %ld uses to execute when priority is %d.\n",
     ((end[i].tv_sec * 1000000 + end[i].tv_usec) - (start[i].tv_sec * 1000000 + start[i].tv_usec)), i+100);
 }
```

- Hint:
    1. You can add a new field, int my_fixed_priority, in the struct task_struct.
        - Do not forget to add this new field, int my_fixed_priority, after the original last line of struct task_struct.
    2. You can initialize the value of int my_fixed_priority in function copy_process. The initialize value of int my_fixed_priority is 0.
    3. NEW (updated: 23rd Dec.) The priority of a CFS process is determined by the static_prio field of its process descriptor.
        - However, directly changing the value of static_prio may not significantly influence the execution time of a process.
        - Thus, in this project, you only need to use the new system call to find the execution time of the tested program when you do not change the static_prio value of a tested process or set the priority of the tested process as 101, 102, ..., and 139, after the tested process obtains the usage of a CPU again.
        - After obtaining the 140 execution time results, analyze the results to see what trend you can find or what relationship between the priority and process execution is. (P.S.: No trend or no close relationship is also a possible answer.)
        - You need to find the right way and location to adjust the priority of a process after the process obtains the CPU again through a context switch.
        - Bonus Credits (20 points)
            - If directly changing the value of static_prio value cannot directly influence the execution time of the tested process and you can find the right data and the right location to change the data so that the execution time of the tested process can be adjusted regularly, you can obtain this 20 bonus credits.
            - The vruntime field of a process descriptor may influence the priority of a CFS process.
    4. The parameter value x of system call int my_set_process_priority(int x) is used to change the value of int my_fixed_priority.
    5. Assume when a process switch occurs, process *P* is the process that obtains the right to use CPU.
        - If process *P* is process *0*, you do not need to change the priority of the process.
            - Creation of process *0* does not use function copy_process.
        - If the value of the my_fixed_priority field is 0, you do not need to change the priority of the process.
        - If the value of the my_fixed_priority field is between 101 and 139, change the priority of the process as my_fixed_priority.

- **Project Submission:** (posted: 9th Dec. 2023)
    - The due day of report submission is **23:55 7th Jan. 2024**
    - The demo will be held from **8th Jan. 2024** to **9th Jan. 2024**
    - Please book your project 2 demo time using this form before 11:59 7th Jan. 2024.
        - 如果表單上列出的兩天都有事，請告知助教，找出其他 demo 時間。
        - 清大的同學可採取線上 demo（前3天去表單看 webex 連結)，填寫組別請用紅色的字體標記。
        - demo時需全員出席。
        - demo會問一些實作細節。
        - hackmd的筆記完整度會影響你的成績，盡量詳細。
    - On site demo of this project is required.
    - During on site demo, the TAs will execute several programs written by them to check the correctness of your system calls.
    - When demonstrating your projects, the TAs will ask you some questions regarding to your projects. Part of your project grade is determined by your answers to the questions.
    - You need to submit an electronic version your project report to the TAs.
        - The electronic versions could be sent to the TAs through e-mails.
        - Do not forget writing the names and student IDs of all members in your team.
        - Your report should contain:
            - Your source code
            - the execution results
    - Late submission will **NOT** be accepted.