HW 6: **Sequential vs Parallel Processing**

Instructor: Prof. Seokin Hong (seokin@knu.ac.kr)

Assigned: November 11, 2019

Due: 11:59pm November 18, 2019

- The goal of this assignment is to write a program to execute multiple tasks in sequential or in parallel. To execute the multiple tasks in parallel, you need to create multiple processes using fork().
  - This program should execute the following two tasks,
    1. Task1: Sort numbers in an array in ascending order
    2. Task2: Sort numbers in an array in descending order
  - The size of the array is 20000, and it is randomly initialized with values from 0 to 19999.
  - **The program has two execution modes: sequential and parallel modes**.
  - In sequential mode, the program executes the tasks one by one in a single process.
  - In parallel mode, the program creates two child processes and executes the tasks in a parent process and three child processes.
    - 1st child process performs the first task.
    - 2nd child process performs the second task.
    - The parent process needs to wait for all child processes to finish.
  - The program should report the elapsed time of the two execution modes.
  - The program should store the sorted numbers in three files.
    - The first file needs to contain the randomly generated numbers.
    - The second file needs to contain the numbers sorted in ascending order.
    - The third file needs to contain the numbers sorted in descending order.

- **Program logic**

```
#include <stdlib.h>
#include <time.h>

int main(int argc, char* argv[])
{
   unsigned array[20000];
   srand(time(NULL));
   for (unsigned int i=0; i<20000;i++)
   {
       array[i] =rand()%20000;
   }

   if ((strcmp(argv[1] ,“0”)==0)
       //execute the four tasks in parallel using fork()
   else
       //execute the four tasks in sequential
}
```

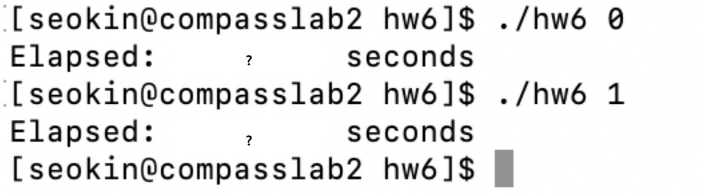
- **How to measure the elapse time**

```
#include <sys/time.h>

struct timeval start_time, end_time;
gettimeofday(&start_time, NULL);

function();

gettimeofday(&end_time, NULL);
double operating_time =
(double)(end_time.tv_sec)+(double)(end_time.tv_usec)/1000000.0-
(double)(start_time.tv_sec)-(double)(start_time.tv_usec)/1000000.0;


printf("Elapsed: %f seconds\n", (double)operating_time);
```

- **Run**

```
$ ./hw6   0        //execute program in parallel mode
$ ./hw6   1        //execute program in sequential mode
```

- **Expected Outputs**



```
[seokin@compasslab2 hw6]$ ./hw6 0
Elapsed:     ?     seconds
[seokin@compasslab2 hw6]$ ./hw6 1
Elapsed:     ?     seconds
[seokin@compasslab2 hw6]$
```

Late Day Policy

All homeworks are due at 11:59pm on the due date. A grading penalty will be applied to late assignments. Any assignment turned in late will be penalized 25% per late day.

Plagiarism

No plagiarism will be tolerated. If the assignment is to be worked on your own, please respect it. If the instructor determines that there are substantial similarities exceeding the likelihood of such an event, he will call the two (or more) students to explain them and possibly to take an immediate test (or assignment, at the discretion of the instructor) to determine the student's abilities related to the offending work.