HW 4: **Sequential vs Parallel Processing**

Instructor: Prof. Seokin Hong (seokin@knu.ac.kr)
TAs: Hoseung Lee (astrohsy@gmail.com) and

Jeong woo Woo (dnwjddn52@gmail.com)

Assigned: October 31, 2018
Due: 11:59pm November 7, 2018

- The goal of this assignment is to make 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 four tasks,
        1. Task1: Counting even numbers in an array
        2. Task2: Counting odd numbers in an array
        3. Task3: Calculating the sum of the even numbers in an array
        4. Task4: Calculating the sum of the odd numbers in an array
    - The size of the array is 2000000, and it is initialized with values from 0 to 1999999.
    - 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 three child processes and executes the tasks in a parent process and three child processes.
        - Parent process performs the first task that counts even numbers in an array.
        - 1st child process performs the second task that counts odd numbers in an array.
        - 2nd child process performs the third task that calculates the sum of the even numbers in an array.
        - 3rd child process performs the fourth task that calculates the sum of the odd numbers in an array.
    - In parallel mode, the parent process needs to wait for all child processes to finish.
    - The program should report the elapsed time of the two execution modes.

- **Program logic**

```
int main(int argc, char* argv[])
{
   int array[2000000];
   for (int i=0; i<2000000;i++)
        array[i] =i;

   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 <time.h>


clock_t tic = clock();

function(); //function we want to measure its elapse time

clock_t toc = clock();
printf("Elapsed: %f seconds\n", (double)(toc - tic) / CLOCKS_PER_SEC);
```

- **How to make the parent process wait for all child processes to finish**

```
.. parent process code ..

while(wait(0)!=-1);
```

wait() system call returns -1 if the current process has no child process!!

<Precautions!!>

- Submit a source code (hw4_StudentID.c) to the LMS (lms.knu.ac.kr)

- If you use the wrong name, NO grade will be given. ex) If your student number is ‘2014123123’, your file name should be hw4_2014123123.c

[Run]

$ ./hw4_2014123123 0 //execute program in parallel mode
$ ./hw4_2014123123 1 //execute program in sequential mode

[We will test your program with the following sequence]

$ gcc –o hw4_2014123123 hw4_2014123123.c

In the parallel mode, the program should show the PID of each process, the result of each task, and the elapsed time as follows.

```
seokin@compasslab1:~/system_programming/hw4$ ./hw4_2014123123 0
parent pid is 5551
pid of parent is 5551, the total even numbers are 1000000
pid of 2nd child is 5553, sum of even numbers is 999999000000
pid of 1st child is 5552, the total odd numbers are 1000000
pid of 3rd child is 5554, sum of odd numbers is 1000000000000
Elapsed: 0.004141 seconds
```

In the sequential mode, the program should show the result of each task and the elapsed time as follows.

```
seokin@compasslab1:~/system_programming/hw4$ ./hw4_2014123123 1
total even numbers are :1000000
total odd numbers are :1000000
sum of even numbers is :999999000000
sum of odd numbers is :1000000000000
Elapsed: 0.013996 seconds
```

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 50% 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.