Assignment 5

Samantha Licea Dominguez | z122301

Excercise 1

Develop an interval timer program using setitimer.

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

void MyAlarm(int);

int main(){
  struct sigaction sa;
  struct itimerval timer;
  struct itimerval otimer;

  time_t t;
  time(&t);

  printf("Starting time %s\n",ctime(&t));

  memset(&sa,0,sizeof(sa));
  sa.sa_handler = &MyAlarm;
  sigaction(SIGVTALRM,&sa,NULL);

  timer.it_value.tv_sec = 3;
  timer.it_value.tv_usec = 0; //3 seconds

  setitimer(ITIMER_REAL,&timer,NULL);

  while(1){}
}

void MyAlarm(int signum){
  time_t t;
  time(&t);
  static int count = 0;
  printf("timer expired %d times (current time %s) \n",++count,ctime(&t));
}
```

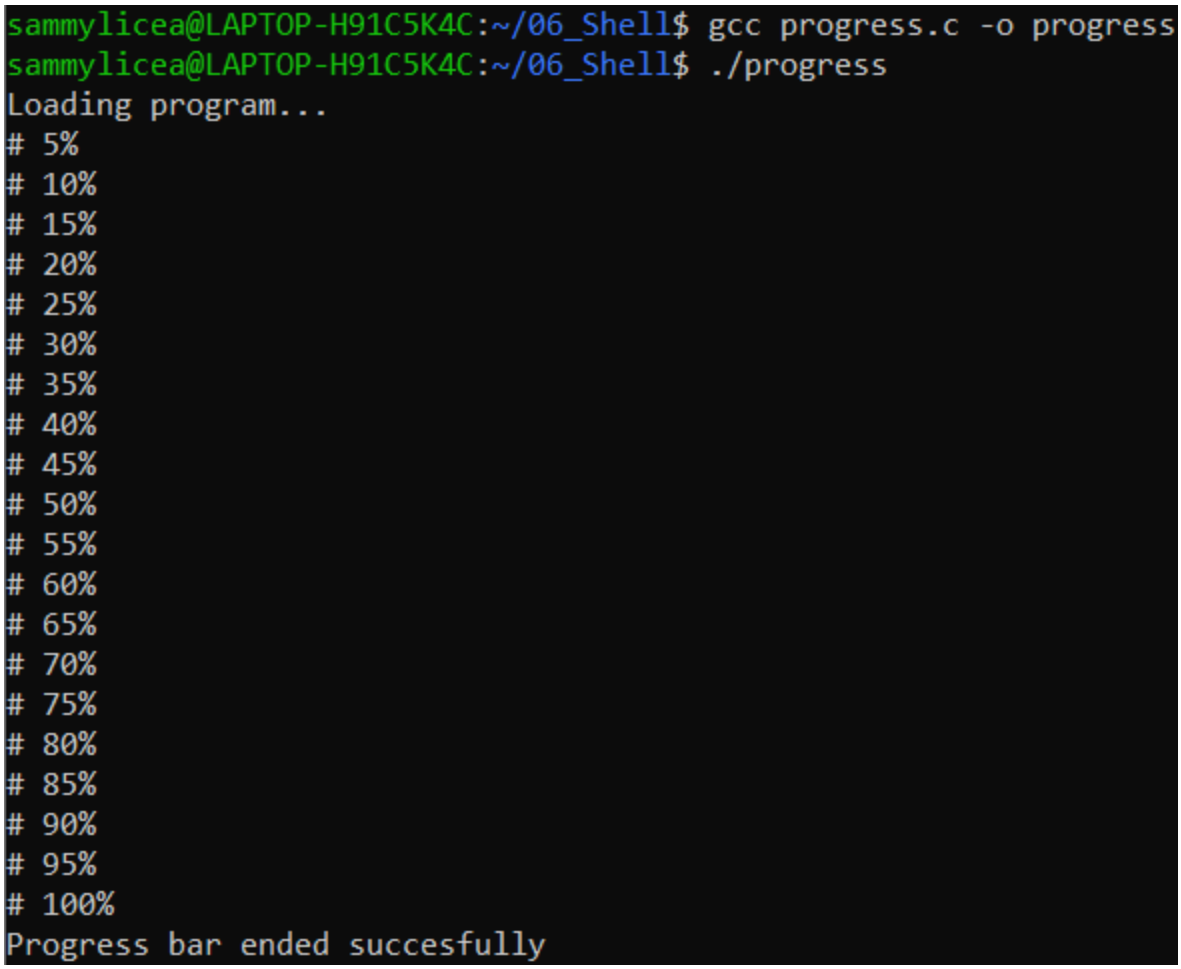
```
sammylicea@LAPTOP-H91C5K4C:~/06_Shell$ gcc -o timeval timeval.c -lc
sammylicea@LAPTOP-H91C5K4C:~/06_Shell$ ./timeval
Starting time Sun Dec  4 11:41:15 2022

timer expired 1 times (current time Sun Dec  4 11:41:18 2022
```

It was a little bit difficult for me this excercise because I could just obtain the result of the timer once 3 seconds after the program started, but it works.

Excercise 2

Let’s make a progress bar

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

void signal_handler(int);

int main(){
  signal(SIGALRM,signal_handler);
  alarm(0.5);
  printf("Loading program...\n");

  int c = 5;
  for(int i = 0;i < 20;i++){
    printf("# %d%c\n",c,37);
    c+=5;
    sleep(1);
  }

  printf("Progress bar ended succesfully\n");

  return 0;
}

void signal_handler(int signum){
  static int cont = 0;
  printf("Inside handler\n");
}
```



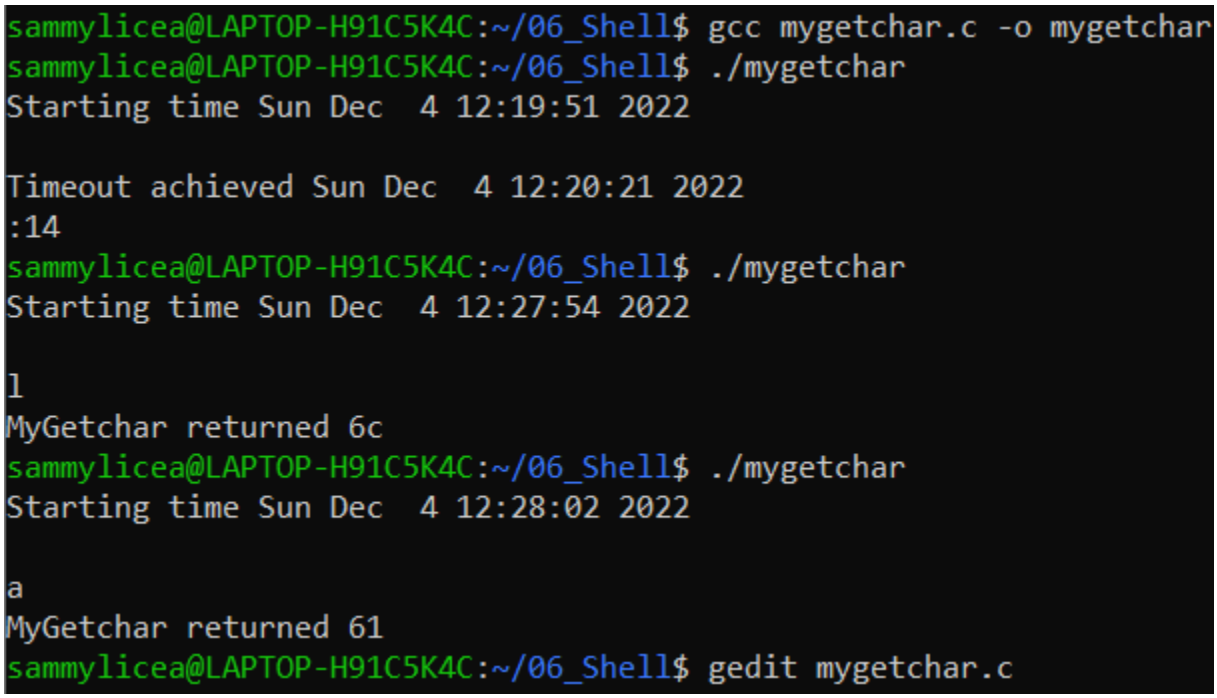
In this case, the progress bar is printed downwards.

Excercise 3

Develop a function mygetchar that adds a timeout function to the function getchar that reads one-character input from the keyboard.

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

void MyAlarm(int);

int main(){
  time_t t;
  time(&t);
```

```
  printf("Starting time %s\n",ctime(&t));

  alarm(30); //timeout after 30 s
  signal(SIGALRM,MyAlarm);
  int c = getchar();
  printf("MyGetchar returned %x\n",c);
  return 0;
}

void MyAlarm(int signum){
  time_t t;
  time(&t);
  printf("Timeout achieved %s:%d\n",ctime(&t),signum);
  exit(1);
}
```



In my program, I added an exit function when the alarm is reached. Otherwise, it prints the character.

Excercise 4

Develop a program that allows child processes to implement different commands as a parallel manner

1. Work A input/output (copy)

2. work B implements accordance with the following:
    a. No dependency
    b. Dependency

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

int fork_exec(const char *);

int main(int argc, char *argv[]){
  if(argc == 1){
    printf("Usage: %s...\n",argv[0]);
    return(-1);
  }
  int i;

  for(i = 0; i < argc;i++){
    fork_exec(argv[i]);
    return 0;
  }
}

int fork_exec(const char *com){
  int status;
  pid_t pid;

  pid = fork();

  if(pid == 0){
    //child process
    system(com);
    //when child process is completed
    exit(EXIT_SUCCESS);
  }else if(pid < 0){
    printf("Fork failed: %s\n",com);
    status = -1;
  }else if(waitpid(pid,&status,0) != pid){
    status = -1;
    return status;
  }
}
```

```
sammylicea@LAPTOP-H91C5K4C:~/06_Shell$ gcc parallel.c -o parallel
sammylicea@LAPTOP-H91C5K4C:~/06_Shell$ ./parallel
Usage: ./parallel...
```

In this code, the loop iterates through the command arguments and calls the function ***fork_exec()*** that uses fork to create a new process and then it uses the ***system()*** function to execute the command. The program ends when it calls the ***fork()*** on all the arguments given in the command line.

However, I haven't understood quite good how to make parallel programs.

References

Parallel processing using fork()

Good day everyone. I have a project that asks us to execute multiple shell commands in parallel using fork(). e.g. shell ls psthe program should creates two childs and execute them in parallel. the question is how can I create multiple proecesses and execute them in parallel.

https://programmersheaven.com/discussion/348344/parallel-processing-using-fork