**Assignment2**

1. Why does the operating system introduce processes? In order to achieve cooperation and coordination among concurrent processes, what should an operating system do in terms of process management?
2. Describe the main differences between processes and programs.
3. Draw the diagram of process state transition and explain when the transition happens
4. Using the program shown below, **explain** what the output will be at LINE A.

#include <sys/types.h>

#include <stdio.h>

#include <unistd.h>

int value = 8;

int main()

{

pid\_t pid;

pid = fork();

if (pid == 0) { /\* child process \*/

value += 15;

return 0;

}

else if (pid > 0) { /\* parent process \*/

/\* parent will wait for the child to complete \*/

wait(NULL);

printf("Pareent: value = %d\n",value); /\* LINE A \*/

return 0;

}

}

1. Describe the actions taken by a kernel to context-switch between processes.
2. Including the initial parent process, how many processes are created by the following program? How do you know?

#include <stdio.h>

#include <unistd.h>

int main()

{

/\*fork a child process\*/

fork();

/\*fork another child process\*/

fork();

/\* and fork another\*/

fork();

}