CS 350 : COMPUTER SYSTEM CONCEPTS Student's Name:
SAMPLE TEST 2 (OPERATING SYSTEMS PART)

MAXIMUM MARK: 100
Time allowed: 70 minutes

NOTE:
Unless otherwise stated, the questions are with reference to the C Programming Language and the UNIX operating System.

Q1 (30 marks)

(a) Give **two** reasons why we may need to use files in our programs.

(b) Give **two advantages** of sequential access file

(c) Give **three limitations** of sequential access files

(d) What is the **difference** between each of the following:

(i) **fprintf()** and **fwrite()**

(ii) **fwrite()** and **write()**

(iii) **system call** and **library call**

(e) (i) Given the following simple program, explain what the program does.

(ii) Give a sample command line usage for the program.

```
/*
*  sample test program  - testProgram.c
*/

#include <stdio.h>
int main( int argc, char *argv[] )
  {

   FILE *fptr1, *fptr2;
   int c;

   if ( argc != 3 )
      printf( "wrong usage \n" );

   else
    if ((fptr1 = fopen(argv[ 1 ], "r")) != NULL)
       if ((fptr2 = fopen(argv[2],"wr")) != NULL)
          {

           fseek(fptr2, 0, SEEK_END);
            while ((c = fgetc(fptr1)) != EOF)
               fputc(c, fptr2);
          }

       else
         printf("Can't open file \"%s\"\n", argv[ 2 ] );
     else
       printf("Can't open file \"%s\"\n", argv[1]);

   return 0;
}
```

Q2 (12 marks).
For each of the following, indicate whether the assertion is TRUE or FALSE.

s/n	**Assertion**	**True or False**
1	The functions **rewind()** and **fseek()** are similar, in that, they are both used to reposition the file position pointer. Also, **rewind()** can be implemented with appropriate use of **fseek()**, and vice versa.	
2	Pipes can be used to communicate between processes that are not related	
3	Signals can be used to communicated between processes on different machines	
4	Semaphores can be used to solve the mutual exclusion problem. They can also be used to provide synchronization between processes.	
5	Multiprogramming and timesharing are one and the same concept, in that, both aims at a more efficient utilization of the same computer system resource - the main memory	
6	Pipes could be blocking or non-blocking, full duplex or half-duplex, but they can only be message-boundary non-preserving (and not message-boundary preserving)	
7	At the time of creation, a process has access to the standard files: **stdin, stdout** and **stderr.** But these standard files can be redirected to other files in the course of program execution.	
8	The **exec** family of system calls is used to overlay one process with another. Thus, the only way to get a child process to perform a different task from the parent is by use of an **exec** call.	
9	With **fread()** (**fwrite()**), we typically read (or write) one element of an array of data structures at a time. However, we can actually read (write) multiple elements with one call to **fread()** (**fwrite()**).	
10	The operating system and other system programs operate in kernel (protected) mode, while all other programs operate in user mode.	
11	A zombie process is a major problem in operating systems, since a zombie process consumes system resources, and the zombie cannot be removed from the system.	
12	The system calls **wait()** and **waitpid()** are similar in the sense that both of them can be used to wait for a process. Therefore, one can always be used in place of the other.	

Q3 (30 marks)

(a) Given that we have a program file imageServer.c. We compile this to produce an executable file, called **imageServer**. At some other stage, we talk of the "***imageServer process***".

(i) Give two differences between **imageServer** (the executable program), and ***imageServer***, the process.

(ii) What is the relationship between the two?

(b) (i) Using a diagram explain the possible states that a process can be, indicating the transition between states.

(ii) Give one reason for each of the allowable transitions in your diagram.

(c) When the following program is executed, the result will be a process tree, which may vary from one run to the other.

(i) Using a diagram, sketch two of the possible process trees than can result if the program were to be executed.

(ii) Very briefly, explain why we could have different possible process trees for the same program.
[You can assume process ID's are integer numbers from 5, 6, 7, …. Assume no error during the fork() system call.]

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
   int k =0;
   pid_t PID;

   PID=fork();

   if  (PID %2 != 0 )
     break;
   else
     printf("I am %ld, the child of %ld, (long)getpid(), (long)getppid() );

   return 0;

}
```

Q4 (28 marks)
(a) Signals and pipes are two mechanisms that can be used to communicate between processes.

(i) List **two advantages** of using **signals** for inter-process communication

(ii) List **two limitations** in using **signals** for inter-process communication

(iii) List **two advantages** of using **pipes** for inter-process communication

(iv) List **two limitations** in using **pipes** for inter-process communication

(b) (i) Give **one** difference between SIGSTOP (the **stop** signal) and SIGKILL (the **kill** signal)

(ii) Give **one** similarity between SIGSTOP (the **stop** signal) and SIGKILL (the **kill** signal)

(d) Two major problems with most solutions to the critical section problem are **busy waiting** and **priority inversion**.

(i) Very briefly, explain what busy waiting means, in the context of operating systems.

(ii) Very briefly, explain what priority inversion means, in the context of operating systems.

(ii) How can these problems be resolved ?