CSCE-313 Final Exam Fall 2015

Student Name ……

Student ID …….

Instructions:

1. This exam is worth 90 points and contains 9 questions.
2. Write your answers below each question in the space provided.
3. If you are unsure about your answer to a multiple choice or True/False statement, please write one sentence legibly explaining your assumption. Note that this is not required unless explicitly stated in the question itself.
4. Please take the first 5 minutes to read through the entire exam and strategize how you will proceed. The exam contains a mix of T/F, multiple choice, and problem solving questions. It is critical that you are able to assess their relative complexity and time-manage the completion of your exam.
5. **We are Aggies. We don't lie, cheat or steal, or tolerate those who do! Sign below showing your commitment and support for our code of honor.**

Signature ……

Question 1. [10 points, 1 point each] Match each term in the left column to the definition and /or description in the right column that fits best. Do this by filling in the void entries on the left:

( )	Pthread	(A)	Lives in Kernel space and entries contain inode pointer among other attributes
( )	Unix Pipe	(B)	Waits till all id'd signals are caught
( )	FIFO	(C)	Mechanism for IPC via message passing
( )	UDP	(D)	Synchronization primitive
( )	Socket	(E)	Integer number identifying a file connection
( )	Mailbox	(F)	Represented by address and port number
( )	SIGWAIT	(G)	Thread that corresponds to POSIX standard
( )	vnode table	(H)	Named IPC structure that resides in Kernel
( )	Mutex	(I)	Protocol for Network Layer in Internet Programming
( )	File descriptor	(J)	Mechanism for inter-process via the use of file descriptors

Question 2. [10 points, 1 point each] Circle TRUE (T) or FALSE (F) for statements noted below:

a) T F Doubling the block size of a Unix file system will double the max file size
b) T F A refcnt of 2 in a vnode table implies a file of size 2 blocks
c) T F Blocking a signal means to delay receipt of the signal until later
d) T F CIA triad in Security refers to Confidentiality, Integrity, and Availability of Data
e) T F A signal mask is a set of signals a process is ignoring
f) T F One major difference between datagram (UDP) sockets and stream (TCP) sockets is stream sockets are not reliable
g) T F A program can create a named pipe by calling "pipe" system call
h) T F Shared memory is faster for communication than pipes because the kernel stores shared memory segments in the CPU cache
i) T F Datagram sockets would be a good choice for transmitting background music for elevators instead of automated teller machine transactions
j) T F A critical section is a portion of memory that cannot be shared amongst processes.

Question 3a. [5 points] Consider the following C program. (For space reasons, we are not checking error return codes, so assume that all functions return normally.)

```c
#include <stdio.h>
#include <signal.h>


pid_t pid;
void handler1(int sig) {
  printf("zip");
  fflush(stdout); /* Flushes the printed string to stdout */
  kill(pid, SIGUSR1);
}
void handler2(int sig) {
  printf("zap");
  exit(0);
}
main() {
  signal(SIGUSR1, handler1);
  if ((pid = fork()) == 0) {
    signal(SIGUSR1, handler2);
    kill(getppid(), SIGUSR1);
    while(1) {};
  }
  else {
    pid_t p; int status;
    if ((p = wait(&status)) > 0) {
      printf("zoom");
    }
  }
}
```

What is the <u>output string</u> that this program prints? EXPLAIN YOUR ANSWER. Correct answers WITHOUT explanation will only receive partial credit.

Question 3b. [5 points, 2.5 points each] Consider the following C program. (For space reasons, we are not checking error return codes, so assume that all functions return normally).

```c
#include <stdio.h>
#include <signal.h>

int counter = 0;
void handler(int sig)
{
  counter ++;
}

int main()
{
  int i;
  signal(SIGCHLD, handler);
  for (i = 0; i < 5; i ++){
    if (fork() == 0){
      exit(0);
    }
  }
  /* wait for all children to die */
  while (wait(NULL) != -1);
  printf("counter = %d\n", counter);
  return 0;
}
```

EXPLAIN YOUR ANSWERS. Correct answers WITHOUT explanation will receive partial credit.
Note: When a child process stops or terminates, SIGCHLD is sent to the parent process. The default response to the signal is to ignore it. The signal can be caught and the exit status from the child process can be obtained by immediately calling wait.

1. Does the program output the same value of counter every time we run it? Circle: Yes No

2. If the answer to the above is a Yes, indicate the value of the counter variable. Otherwise, list all possible values of the counter variable.

Answer: counter = _____

Question 4. [10 points, 1 point each] Circle the best answer for each of the 10 multiple choice questions below. ***If in doubt about your answer, use the space below to state your assumption in 1 sentence.***

- A Unix directory contains
    a. Names and inode number of files
    b. Names and sizes of files
- Unix uses indirect blocks to store
    a. symbolic links
    b. lists of block numbers of data blocks
- The size of an inode table determines
    a. the maximum number of files in the file system
    b. the maximum size of a file in the file system
- A web server sends output of programs to the client by
    a. opening a new socket for the program
    b. using dup2 to redirect standard output
- One thing a server does that a client does not do is
    a. a server loops
    b. a server waits for a connection
- A server can handle several requests at once by
    a. using fork to create a new process for each request
    b. using socket to create a new socket for each request
- To prevent race conditions, threads should use
    a. signals
    b. mutex objects
- The input redirection symbol "<" is processed by
    a. the program that reads the data
    b. the shell
- The thread that accepts calls in a web server does not close the socket
    a. because that would close it for the worker thread, too
    b. because it is closed automatically when the worker thread finishes
- The shell notation "prog1 | prog2" is a request to
    a. send all output from prog1 to the standard input of prog2
    b. send the standard output of prog1 to the standard input of prog2

Question 5a [6 points] Consider a UNIX filesystem with the following components: Disk blocks are 4K Bytes. All pointers are 4 Bytes long. An inode has 12 direct block pointers, one indirect block pointer and one double-indirect block pointer. The total inode size is 256 Bytes. Both indirect and double indirect blocks holding pointers take up an entire block. How much disk space, including metadata and data blocks, is needed to store a 4 GB DVD image file? You can leave the answer in symbolic (e.g., 6MB + 3KB, or powers of 2) form– you must show your calculation for credit. For metadata, make reasonable assumptions.

Q5b [4 points] Please draw the directory structure showing the result of executing the following stream of UNIX commands.

mkdir: creates a directory with the name supplied in the argument
rmdir: removes a directory with the name supplied in the argument
mv: moves a file or directory to new location (or name)
cd: change directory

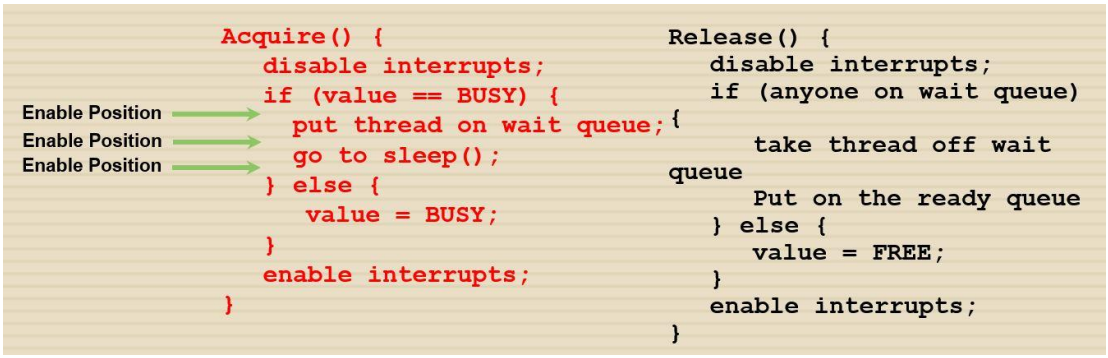
Eg.1 User View of a File System - Working example

```
% mkdir demodir
% cd demodir
% mkdir b oops
% mv b c
% rmdir oops
% cd c
% mkdir d1 d2
% cd ../..
% mkdir demodir/a
```

<space for drawing directory structure>

Question 6a. [5 points] A binary semaphore is initialized to TRUE. Then 5 P() operations are performed on it in a row followed by 8 V() operations. Now 5 more P() operations are performed on it. What is the number of processes waiting on this semaphore? Show your work.

Question 6b. [5 points] In class, we learnt about how we can protect a critical section by encasing it between *lock acquire* and *lock release* primitives. This allows only one process or thread to be able to enter the critical section at a time and prevents a possible race condition from occurring. One such mechanism for lock acquisition and release is through *disabling and enabling interrupts*. We also learnt that it is 'responsible' to enable interrupts *as soon as possible*. In the below pseudocode, there are three such opportunities for early enabling of interrupts. These are shown with markers A, B, and C. Markers point to the insertion of "enable interrupts". Comment on each of them (a) whether the idea will be feasible or not, and (b) 1-2 line reasoning for your answer.



Enabling interrupts at Marker A:

Enabling interrupts at Marker B:

Enabling interrupts at Marker C:

Question 7. [10 points: 1+3+3+2+1] In the following pseudo code for a bounded buffer problem in a producer-consumer form for a beverage dispenser, please answer the following ensuing questions:

```
Producer(item) {
    emptySlots.P();
    mutex.P();
    Enqueue(item);
    mutex.V();
    fullSlots.V();
}
Consumer() {
    fullSlots.P();
    mutex.P();
    item = Dequeue();
    mutex.V();
    emptySlots.V();
    return item;
}
```

Assume: Enqueue adds an item to the buffer, Dequeue does the opposite.

7a. What should the semaphores "emptySlots" and "fullSlots" be initialized to?

7b. Describe the function of the above code above by annotating each line above with comments.

For this question: **DO NOT WRITE HERE**. Annotate the code **ABOVE** with comments next to the code

7c. Is the order of P important? State YES, NO, or MAYBE along with REASON.

7d. Is the order of V important? State YES, NO, or MAYBE along with REASON.

7e. Will this code work for more than one producer and one consumer?

Question 8a [5 points] Consider the following code:

```
int main(int argc, char* argv[]) {
  char buf[] = "ab";
  int r = open("file.txt", O_RDONLY);
  int r1, r2, pid;
  r1 = dup(r);
  read(r, buf, 1);
  if((pid=fork())==0) {
    r1 = open("file.txt", O_RDONLY);
  }
  else{
    waitpid(pid, NULL, 0);
  }
  read(r1, buf, 1);
  printf("%s", buf);
  return 0;
}
```

Assume that the disk file file.txt contains the string of bytes 15213 . Also assume that all system calls succeed. What will be the output when this code is compiled and run? Explain your answer.

Reference: **(a) waitpid()** system call suspends execution of the calling process until a child specified by pid argument has changed state. **(b) dup(fd)** copies the given file descriptor fd into the lowest-numbered available file descriptor and then returns the new file descriptor.

Question 8b [5 points] Consider the following code

```
int main(int argc, char* argv[]) {
  char buf[] = "abc";
  int r = open("file.txt", O_RDWR);
  int r1 = 0;
  int r2 = open("file.txt", O_RDWR);
  dup2(r, r1);
  read(r, buf, 1);
  read(r2, buf, 2);
  write(r, buf, 3);
  read(r2, buf, 1);
  write(r1, buf, 1);
  return 0;
}
```

Assume that the disk file ***file.txt*** originally contains the string of bytes **12345** . Also assume that all system calls succeed. **What will file.txt contain when this code is compiled and run? Explain your answer.**

Question 9. [10 points] Circle the best answer for each of the 10 multiple choice questions below. *If in doubt about your answer, use the space below to explain in less than 2 sentences.*

1. Shared memory is faster for communication than pipes because
    a. the kernel stores shared memory segments in the CPU cache
    b. data does not have to be copied from one process to another
2. A program deletes a shared memory segment by using
    a. shmdt
    b. shmctl
3. If two different processes write to the same pipe,
    a. data from one writer may overwrite data from the other one
    b. both sets of messages will get to the reader
4. If two different processes read from the same file,
    a. both will read the same data
    b. one will get some data, the other will get the rest
5. If two different processes read from the same pipe,
    a. both will read the same data
    b. one will get some data, the other will get the rest
6. A named pipe is removed with the system call
    a. unlink
    b. close
7. A client program makes a connection to a server program by calling
    a. socket
    b. connect
8. The listen system call is used to
    a. receive data from a client
    b. enable incoming connections to a socket
9. A pipe can be used to transfer data between
    a. only two processes
    b. any number of processes
10. When setting up a pipe to a child, the order of system calls is
    a. the parent calls pipe then calls fork
    b. the parent calls fork, then both processes call pipe