**CS2240, Final Examination**

**Fall 2015, December 15, 2015**

**git git@jazz.cs.wmich.edu:2240/FinalF15**

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1) What is the kernel of the operating system ?

2) Give an example of a

a) Compile time limit

b) Run time limit

c) Indeterminate limit

3) a) Write a complete program that has a race condition. Push it to the FinalF15 as Race.c. Please document clearly. Include a script of the condition.

b) Remove the race condition. Push it to the FinalF15 as NoRace.c. Please document clearly. Include a script of the cured condition.

You may put any information here that can inform me as I look at your work.

5) Write a segment of C code that creates a pipe between a parent and child process such that stdin of parent receives anything written to stdout in the child.

6) Assume that your shell has a command line of the form

% CMD1 | CMD2 | CMD3 | CMD4 | CMD5

Write pseudo-code that describes how CMD3 is executed in the chain of other commands. You may assume a data structure that makeargv() has created. Specify any other necessary assumptions to describe the algorithm for this ‘middle’ case.

7) Kernel Data Structures for files include a) a table of all processes indexed by process id, b) for each process, an array of file descriptors that include file descriptor flags and pointers to file tables for each open file. Each such file table contains the current file offset for that descriptor and a “v-node pointer” into a v-node table that includes the current file size and i-node references to actual sectors that make up the file.



1. Can two different file descriptors have the same file open, one for read only and write only ? Explain.
2. Can two different processes have the same file descriptor value (e.g. 4) reference two different files ? Explain
3. Can a file offset be greater than a file size ? Explain.
4. If Process 1 has file F open for reading, can Process 2 open file F and truncate it, or will this cause an error ?

9) What is the 32 bit IEEE floating point representation for -31.875 ?

10)Fill in the blanks with numbers from the list in the box.

Some numbers can be used more than once.

1. accept
2. address
3. bind
4. blocked
5. client
6. connection
7. ip
8. listen
9. port
10. receive
11. send
12. server
13. socket
14. terminal

Not all numbers must be used.

When a server uses a socket it must first use the \_\_\_\_\_ system call.

The socket is then \_\_\_\_ to a \_\_\_\_\_ using the \_\_\_\_\_\_\_ function.

Then a \_\_\_\_\_\_\_\_\_\_\_ process can transmit to the server whose control

Is \_\_\_\_\_\_ while waiting for a return from the system call \_\_\_\_\_\_ .

The server then creates a \_\_\_\_\_\_ when a successful execution of the

system call \_\_\_\_\_\_\_\_ is executed.

The resultant \_\_\_\_\_\_\_ can be used to \_\_\_\_\_ and \_\_\_\_\_ information with

the \_\_\_\_\_\_\_\_\_\_\_ .

11) A) A socket is an IPC mechanism which allows processes to communicate in a FIFO manner. Sockets differ from pipes because

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

B) How does a thread differs from a process ? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

C) If thread A contains the system call to join thread B, which thread is guaranteed to finish first ?

12) An IP address can be stored in memory as a long int and then displayed in the familiar dotted form (e.g.“255.254.254.252”)

a) What long int value does the above represent ? Represent it in hex . \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b) When the above is transmitted on the internet, in what order are the bytes in a) transmitted ?

\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_

13) Lucky 13. Write a program that will play musical chairs in the following way. The program is run with the number of players on the command line (call it N). The program will create N threads and implement (N – 1) ‘chairs’ by passing them from one to the next (N+1) and similarly receive a chair from the previous (N-1). Create a stopping condition after which the thread that has no chair reports.