
Name: _____

PSU User ID (e.g., abc123, not 9 digit numbers): _____

To avoid grading mistake and also make grading faster, please write more heavily and clearly (if possible using a pen instead of pencil), as we will scan your exam papers, which often turn out to be blurry and hard to recognize your handwriting.

CMPSC 311 Final

May 5, 2022

Closed book, closed neighbor, no electronic tools or additional papers.

You may not share or discuss exam questions with anyone.

1. Short Questions (36 pts total, 4pts each, be brief)

- a. What is command substitution in bash program? Give one example.

Command Substitution allows the output of a command to replace the command itself.

For example:

file \$(which ls)

or

file `which ls`

here "which ls" command gives an output and that output will replace "which ls" for "file" command

- b. Write a grep command that finds the lines in a text document test.txt which contain the word "foo" or "bar"

egrep "foo|bar" test.txt

- c. Write a regular expression using a range to identify any word that represents hexadecimal and which length is exactly 6

0x[0-9A-Fa-f]{6}

Or

[0-9A-Fa-f]{6}

- d. Here is a bash program. In this program, I have used a built-in command *test*. How can I write the same program without using *test* using *shorthand tests*

```
salary = 2000
test $salary -ge 1000 &&
echo I have good salary
```

Ans:

```
salary = 2000
```

```
[[ $salary -ge 1000 ]] &&  
echo I have good salary
```

- e. Write the names of function calls to create a child process and a new thread, respectively. You can ignore the function parameters.

```
fork()  
pthread_create()
```

- f. A parent process may call the blocking `wait()` function to receive a return value from the child process. Briefly mention in a sentence how to solve this problem so that the parent won't have to keep waiting.

Use signal. The child process may raise a signal (e.g., `SIGCHLD`), and the parent process can catch it with a signal handler, where the `wait()` is invoked.

- g. What is a critical section?

a piece of code that accesses a shared variable and must not be concurrently executed by more than one thread

- h. Name two methods to avoid race condition.

```
atomics  
mutex  
(or others such as semaphore)
```

- i. Name the five layers in the TCP/IP protocol stack

Application, Transportation, Network, Data Link and Physical layers

- j. Why a client program uses an ephemeral port number whereas the corresponding server program typically uses a well-known port number (e.g., 80 for a web server)?

The server program uses a well-known port number so that it can be located (or connected, or reached) by any client who initiates the connection. The client can use an ephemeral port number because the client's request packet contains the port number for the server to reach it back.

- k. Suppose you are sending a 32 bit integer x over the Internet. Write the line code to convert it into network byte order.

```
x=htonl(x);
```

- l. What is domain name service (DNS) used for?

find the ip address given a domain name (or the vice versa)

2. Medium Questions (48pts total, 6pts each)

- a. We have special variables PWD, USER, HOME, and PATH on Unix. What do they contain?

PWD: current directory

USER: name of the current user

HOME: the current user's home directory

PATH: executable files

- b. What is the difference between *Soft quotes* and *Hard quotes*? Give examples of each of them to explain.

Soft quotes: "...."

Allow variables (and some other things)

For example:

```
var="Hello"
```

```
echo "$var"
```

it will print *Hello*

Hard quotes: '....'

Quote everything

For example:

```
var="Hello"
```

```
echo "$var"
```

it will print *\$var*

- c. Write a bash program. In the program, you will run a while loop to print the following output

```
1
11
111
1111
11111
```

Ans:

```
while [[ "$x" -lt 99999 ]]; do
  echo "$x"
  x="1$x"
done
```

- d. What is a socket in socket programming and how is it uniquely identified?

Socket is an end point of a connection, and uniquely identified by the ip address and port number pair.

- e. The system call "int listen(int sockfd, int backlog)" tells the OS to receive connections for the process, where sockfd is the socket obtained previously. What is backlog here and what would mostly likely to happen when you set backlog to 0?

Backlog is the number of connections to queue. If it is 0, client requests may get dropped by the server when it is busy.

- f. Suppose you call the following blocking accept() system call to receive a connection from the client

```
int sock2 = accept(sock1, addr, addrlen);
```

Describe what is the difference between sock2 and sock1 and how is sock2 typically used in the code following it.

Socket1: is the passive socket used by the server to receive clients' initial requests.

Socket2: is the active socket which is used later to communicate with the client to exchange the following messages (through read/write).

- g. What is (program) profiling? How can it help performance optimization?

Profiling involves running a version of the program instrumented with code to measure how much time is spent in certain areas of the code (or how much time in each of the modules of the program)?

(Or it is a process to measure the performance behavior of a program, such as time cost and memory usage of instructions, modules (or function)).

Because profiling is able to identify the performance bottleneck (or say which parts of a program use the most time or run the most), thus providing the right location (or parts) for optimization

- h. Suppose you want to send 400 bytes from a buffer *buf* into a socket (number 100). Write the code which will guarantee all 400 bytes are sent successfully (with a loop). (hint: the write function signature is: `ssize_t write(int fd, const void *buf, size_t count)`)

```
ssize_t n=0;
while(n<400)
{
    n+=write(100, &buf[n], 400-n);
}
```

- i. Assume that a program has two modules A and B. A represents 80% of the run time of the program. By using L2 cache, the optimization reduces the runtime of module A from 400ms to 100ms. What is the program speedup?

The speedup for the optimized part is $s=400/100=4$;
 $1/(0.2+0.8/s) = 1/0.4=2.5$