COURSE Final Exam 10

DATE

Your Name Here: ________________________

Instructions: You have TIME for this exam. Please write your answers on the pages in this exam booklet. No scrap paper or additional sheets will be accepted. Watch your time and be concise. Write clearly (illegible answers will be ‘silently ignored’), and *always* check the return value of a system call. Good luck.

prob	points	got	section
1	4		
2	4		
3	4		
4	4		
5	4		
6	4		
7	4		
8	4		
9	6		
10	6		
11	6		
12	6		
13	6		
14	3		
15	3		
16	3		
17	3		
18	9		
19			
a	3		
b	2		
c	3		
d	2		
e	3		
f	2		
g	2		

.

Problems 1-8: Short answer questions. Answer each question clearly, precisely, and refer to specific system calls when appropriate. Write your answer in the space provided.

1. Where can a programmer find information about the format of structs used in Unix system calls?

2. What is the meaning and value of *buffered* input and output?

3. What are *inodes* and where are they stored?

4. How does Unix store time and date values for things like file modification times and user login times?

5. What is the role of the *wait()* system call and what does its return value represent?

6. How does a program turn off keyboard echoing without calling an external program?

7. What does the function `perror()` do, and when would you use it?

8. Name two attributes of a process and the system calls a process uses to determine those attributes.

Part Two Five problems, each worth 6 points

Problems 9-13: Compare and contrast. Each of these problems mentions two related concepts, system calls, or operations. For each pair, explain briefly and clearly (a) what they have in common, (b) when you would use the first item, *and* (c) when you would use the second item.

9. `fopen()` vs `fdopen()`

10. `stat()` vs `tcgetattr()`

11. `datagram socket` vs `stream socket`

12. `kill()` vs `exit()`

13. `connect()` vs `pipe()`

.

‘*Parent-Child Communication*’ Unix provides many means for one process to send information to another process. Consider the situation where a parent process has some strings it wants to transmit to another process. For each of these four methods, explain how the strings are sent and how they are received.

14. transmitting data using a disk file

15. transmitting data using execvp()

16. transmitting data using a pipe

17. transmitting data using the environment

19. An enhancement to your small shell - $(cmd)

The shell you wrote for homework does variable substitution. That is, it scans each command line for strings of the form `$varname` and substitutes for the dollar sign and variable name the value of the variable.

The real shell does other sorts of substitution. My favorite kind of substitution is command output substitution. The Bourne and C shells use the backwards apostrophe notation **‘** while the korn shell uses the **$(cmd)** notation. For example if the user types:

`X=$(pwd)` or `echo the date and time now are $(date)`

the shell replaces the pattern `$(pwd)` with the *output* of the pwd command and the replaces the pattern `$(date)` with the output of the date commaand. In the bourne shell, the notation would be `` `date` ``. The korn shell notation is more powerful because it allows nested expressions.

Describe how you would implement $(cmd) command output substitution in your shell. Include pseudo-code that demonstrates the logic of the new feature. Please be sure to discuss: (a) What parts of your shell program you need to modify and (b) What system calls you would use.

c) The korn shell allows for embedded variables and embedded command substitution like:

```
F=/etc/passwd
echo the number of registered users is $(wc -l $F) at $(echo $(date))
```

How would your solution support this expression nesting?

.

20. *Multi-Sumac*

Your SUMAC program allows you to read your mail on any POP server on which you have an account. Some people receive mail in more than one place. Sometimes you are waiting for mail from a particular person or one with a particular subject. It is a nuisance to use SUMAC every few minutes to check several mail servers for a letter from user *X* or about subject *S* .

It sure might be handy to have a different version of SUMAC that would scan a list of sites at a regular interval and display, using the standard SUMAC screen display the letters you have been waiting for.

For convenience, you could prepare a file that contains the hostnames of the pop servers and your logname on each one. For security, you would not want to put the passwords in that file.

The program would start up and prompt you for the passwords for each of the machines in your list of mail servers. If you do not enter a password for a machine that machine is not included in the list of scanned sites. The program then prompts you for an interval in seconds and then for the name of the sender you want to hear from or a subject you want to hear about.

The program then polls the various mail servers at the specified interval and lists in the standard format any letters that match your criteria. You may view or save any of the matching letters.

Ok so far? For this problem, you will discuss the general design of this mail watching program.

Part I: Watching for Special Letters [5 points]

a) Describe how the program will watch for letters at multiple sites. What is the procedure for doing so? You do not need to write code, but you do need to describe what functions you need to contact the servers and get the desired information.

b) How is the scheduling done?

Part II: Finding Special Letters [5 points]

c) Once you find a letter that matches the search criteria, how do you keep track of it so you can find it when the user wants to view or save it?

d) How do you inform the user that such a letter has arrived?

Part III: User Interface [3 points]

e) One or more letters appear in the list. A user asks to view or save one of them. What does the program do to perform this operation? Describe the steps in some detail.

Part IV: Resource Management [2 points]

f) What if a new letter is detected when the user is viewing a letter? Can the screen be updated when the user is running ‘more’ or ‘less’? Should the searching activity be suspended during viewing?

Part V: Other Ideas[2 points] Add any other ideas or questions you have about this project.