

# Report subjects

- Select **at least 5** questions from below.
- Answers should be submitted by E-mail to `tito@csa0` . Don't forget to put your name on the mail.
- Deadline is **13:00, January 12** .
- Sample files are in `~tito/Reports` , so at first you should copy them by  

```
% cp -pr ~tito/Reports ~
```

## C shell basics

1. How to get the current directory into the prompt? For example, when you are in `/home/csa0/bin`, let prompt be like  

```
fkokus07@csa0:/home/csa0/bin/%
```

and when you `cd` to `/usr/lib`, let prompt be like  

```
fkokus07@csa0:/usr/lib/%
```
2. What should we do to remove the files below? They can not be deleted in the ordinary ways. In addition, how to make the files which have such extraordinary names?  

<code>-abc</code>	(which starts with hyphen)
<code>?abc</code>	(which starts with wildcard ?)
<code>*abc</code>	(which starts with wildcard *)
<code>abc def</code>	(which contains blank within)
<code> abc</code>	(which starts with blank)
<code>;</code>	(whose name is "colon")
<code>-;</code>	(whose name is "hyphen + colon")

You can find these files in `~tito/Reports/files` . Copy them to your directory and try to delete them.
3. Someone would like to print out the contents of `fileA` with line numbers, and did the following command, but failed.  

```
% cat -n fileA > lpr
```

Why?, and what is the correct way?
4. To concatenate `fileA` and `fileB`, someone tried the command  

```
% cat fileA fileB >> fileA
```

In this case, what kind of output did he get? And why is that? Then, What is the best way to achieve his object?

5. On the machines in this classroom, someone who wanted to know the settings of his prompt did

```
% echo $prompt
```

However, the shell returned an error message

```
echo: No match.
```

and failed. Similar commands like `echo $history` worked. Why is that? What is the best way to show the setting of `prompt` successfully?

6. Alias to `pwd` is perhaps `echo $cwd` in your `.cshrc` file. However, alias format like

```
alias pwd echo $cwd
```

does not give the right output. Why? What is the right way?

7. In SunOS system (which you are using now), how long can words on C shell be? Can you set a variable which has 1,000,000,000 characters long?

8. Redirection method of `>&` mixes the standard output and standard error output. Suppose there is a program which creates both, how can we separate them by using `>` and `>&`? Sample code is as follows:

```
#include <stdio.h>
main(){
    fprintf(stdout, "This line is for standard output.\n");
    fprintf(stderr, "And this line is for standard err output.\n");
}
```

Using this code (write, save, compile and execute), try to separate the outputs into standard output (first line) and standard error output (second line).

## C shell scripts

9. Editor Emacs makes backup files which start from `#` or trailing `~` (like `#file` or `file~`). Someone wanted to delete all these files at a time, and made a C shell script like

```
#!/bin/csh
rm -f #* *~
```

However, it didn't work. Why is this?, and how you can solve the problem? Sample files are in `~tito/Reports/emacs`.

10. In MS-DOS system, it is possible to rename all the `*.foo` files to `*.bar` at a time by using the command `ren`. But UNIX equivalent `mv` can not work in this way. For example,
- ```
% mv *.foo *.bar
```

will fail. Why is that? Then, try to make a shell script which rename all the \*.foo files to \*.bar at a time such like

```
% somescript *.foo *.bar
```

or

```
% somescript foo bar
```

11. Below, C shell script to exchange the two file names (same as List 2 today) is shown. This script does not work if each of the specified files is a directory file. Rewrite the script and make it handle the directory files also.

```
#!/bin/csh
if ($#argv != 2) then
    set cmd = $0
    echo "Usage: $cmd:t [file1] [file2]"
    exit
else
    if      (! -e $argv[1]) then
        echo "$argv[1] does not exist."
        exit
    else if (! -e $argv[2]) then
        echo "$argv[2] does not exist."
        exit
    else
        set TMPFILE = /tmp/NAME1_____
        /bin/cp    $argv[1] $TMPFILE
        /bin/mv -f $argv[2] $argv[1]
        /bin/mv -f $TMPFILE $argv[2]
        /bin/rm -f $TMPFILE
    endif
endif
```

12. Below, C shell script to calculate the total bytes in subdirectories recursively (same as List 3 today) is shown. At the top of the script, the C shell variable **nonomatch** is set. What for? (hint: try to make an empty subdirectory, and run the script)

```
#!/bin/csh
set nonomatch # <--- HERE!
if ($#argv != 1) then
    echo "usage: $0 directory"
    exit
endif
set count = 0
foreach file ($1/*)
```

```

if (-f $file) then
    set size = '/bin/ls -l  $file'
else if (-d $file) then
    set size = '/bin/ls -ld $file'
    set recurse = '$0 $file'
    @ count = $count + $recurse
else
    echo "$file not included" >>! /tmp/nototal
    continue
endif
@ count = $count + $size[4]
end
echo $count

```

13. Make a shell script which rings the terminal bell, such like
  - % bell (rings once)
  - % bell 5 (rings five times)

## Other basic commands

14. What should we do to get the list of all files which satisfy the three conditions
  - (a) which are under /home, and
  - (b) which are modified within these two weeks, and
  - (c) which have the octal permissions of 644 (-rw-r--r--).
 by **one line command**? (hint: use the command **find**)
15. What way are there to know whether the user to whom you sent an E-mail has read your mail or not?
16. Make aliases as follows:
  - (a) Sort the results of **ps -aux** in order of process ID (PID).
  - (b) Sort the results of **ps -aux** in order of the CPU time used by each process.
  - (c) Show *your* processes only in the results of **ps -aux**.
17. Try to write one of the UNIX commands **cat** by yourself (simple version is enough) using C language or shell script. Send the code (or script) by E-mail.
18. Tell at least two ways to get a recursive directory listing.
19. Explain the reason that the command **chmod** is available only for the owner of the files and superuser.

## Else

20. Explain briefly the major differences between MS-DOS and UNIX, advantages and disadvantages of them.
21. Explain briefly the advantages and the disadvantages of E-mail compared with the other communication methods (such as telephone, facimilie, ...).
22. On UNIX editors like `vi` or `emacs`, function keys or arrow keys are not available. Why is that?
23. `cal` is a command to display a calendar for the specified year. According to the on-line manual of `cal`, they say “Try September 1752”. So, try to do that, and explain briefly what the result means. It is not joking (I think).