

Deadline

Mon May 31 17:00:00 GMT-8 2004

Learning Keywords

who, while, sleep, awk, grep, echo, read

Part 1

In the first part of this assignment, you are required to implement a "buddy" utility using shell script which alert you when one of your buddies logged on or logged off sunfire.

You should name your script `buddy.sh` and your script should accept one command line argument, which is the userid of your buddy. For example,

```
buddy.sh tanahkow
```

When you first run your script, it should check if your buddy is currently log on. If so, it will alert you with the message:

```
tanahkow is currently logged in (Tue May 18 13:25:10 SGT 2004)
```

After which, your script should check periodically (every 30 seconds) if the login status of your buddy has changed using a combination of `who` and `grep`. If your buddy has just logged in, your script should output a message, together with the *current* time.

```
tanahkow has logged in (Tue May 18 15:50:40 SGT 2004)
```

When your script detects that your buddy has logged out, you should output a message together with the current time.

```
tanahkow has logged out (Tue May 18 17:31:10 SGT 2004)
```

Your script should notify you about your buddy only when his login status has changed, i.e., it should not keep telling you that your buddy is logged in every 30 seconds.

Part 2

In the second part of this assignment, you are required to implement an instant messaging utility using shell script which allows you to chat online with your buddies on sunfire¹.

You should name your script `chat.sh` and your script should accept one command line argument, which is the userid of the person you want to chat with. For example,

```
chat.sh tanahkow
```

¹Similar utility, called `chat` already exists on sunfire. You are not allowed to call `chat` from your shell script

How It Works

Writing to tty

When a user logon remotely to sunfire, a special file that represents the terminal is assigned to that user. You can find out your terminal name by using the command `tty`.

```
$ tty
/dev/pts/95
$ echo abc > /dev/pts/95
abc
```

If you login more than once concurrently to sunfire, each login session will be assigned a different terminal. From one login session, you can make text appears on the terminal of the other login session by writing to its terminal. You can try this yourself. Login into sunfire using two ssh sessions concurrently. If the tty for session 1 is `/dev/pts/95`, Running the command below should make the text "abc" appears on the screen of session 1.

```
$ echo abc > /dev/pts/95
```

By default, you can only write to the terminals that is assigned to you. This is to prevent arbitrary person from writing on your screen. But for this assignment, we will have to lower the permission and make your terminal world writable.

```
$ chmod a+w $(tty)
```

Now anyone who knows your tty can write on your screen! Remember to change the permission back when you are done chatting. (Preferably this should be done automatically from your script.)

WARNING: Never, never make your terminal world readable. By making it readable to the world, anyone can read what you type on your keyboard, including passwords.

Finding Out Someone Else's tty

To find out the tty of the person you like to chat with, you can use the `who` command. The second column indicates the name of the tty.

```
tanahkow pts/95 May 23 12:31 (somewhere.ddns.comp.nus.edu.sg)
```

For example, the second column above indicates that the tty for tanahkow is `/dev/pts/95`.

Start Chatting

Once you found out the other party's tty, you can repeatedly read your message from your keyboard (using `read`) and send it to the remote tty (using `echo` and redirection), until the user enters Control-D (EOF).

You may want to prepend your message with your userid so that the other party knows who the messages are coming from. You can retrieve the userid of the current user using the environmental variable `$USER`.

If the other party has more than one tty, pick one which is writable. If none of its tty is writable or the user is not logged in, output an appropriate error message and quit.

Screen Formatting

You have to be careful when you write on someone else's screen. It is rude to replace part of the screen with your message and move the cursor around. To avoid this, we will use some terminal control sequence to properly place your message on the screen. You should prepend your message to the remote user with the following terminal control sequence

```
\033\067\033[H\033[M\033\070\033[A\033[L\r
```

and append it with

```
\033\070
```

so that your message will appear above the current cursor and will not interfere with the user's current prompt. Remember to use `-e` option of `echo` when you want to send special characters.

Example Usage

```
$ chat.sh leeahhua
  tanak: hello Ah Hua
leeahhua: hi, Ah Kow. sorry not free.
  tanak: ok lah. ciao.
$
```

Abuse Warning

Such chat program can easily be abused. This includes faking userid and screen bomb (`chat.sh ooiwt < verylargefile`). School of Computing has strict rules against the abuse of chat utilities, and the rules apply to this assignment as well. Please apply your common sense when you play with this assignment!

Submission Requirement

You are required to submit the encrypted version of your shell scripts `buddy.sh` and `chat.sh`. Make sure you have read the submission instruction document posted on CS2281 website. For this assignment, create a subdirectory under `$HOME/CS2281_LABs/` called `a5` and put your encrypted files under the subdirectory. You must include your name as a comment in the *second* line of your files (Since first line is sha-bang). I will access your submission through the pathname `$HOME/CS2281_LABs/a5/{chat,buddy}.sh.pgp`. It is your responsibility to make sure that the filenames are correct and permissions are set properly according to the instructions given.