

**Q1. Basic salary of A is input through the keyboard. His dearness allowance is 40% of basic salary and house rent allowance is 20% of basic salary. Write a script to calculate his gross salary.**

**Sol.**

```
echo ' enter the basic salary '  
read bassal  
da=$((bassal*40/100))  
echo "da="$da  
hra=$((bassal*20/100))  
echo hra=$hra  
gs=$((bassal+$da+$hra))  
echo "Gross Salary = Rs." $gs
```

**Q2. If a five digit number is input through the keyboard, write a script to calculate the sum of its digits.**

**Sol. While [ cond ]  
    Do  
        Statements  
    done**

```
echo Enter any five digit number
read num
((d1=num%10))
((num=num/10))
((d2=num%10))
((num=num/10))
((d3=num%10))
((num=num/10))
((d4= num%10))
((num=num/10))
((d5=num%10))
((sum=d1+d2+d3+d4+d5))
echo Sum of digits = $sum
```

**Ques 3: Write a shell script which will receive either filename or the filename with its full path during execution. This script should obtain information about this file by `ls -l` and display it in proper format.**

**Sol :**

```
echo Enter a filename
read filename
```

```
var=`ls -l | grep $filename`
set -- $var
```

```
echo Name=$9
echo File access permissions=$1
echo Number of links=$2
echo Owner of file=$3
echo Group to which he belongs=$4
echo Size of file=$5
echo file modification date=$6 $7
echo file modification time=$8
```

**Q4. If cost price and selling price of an item is input through the keyboard, write a script to determine whether the seller has made profit or incurred loss. Also determine how much profit was made or loss incurred.**

**Sol.**

```
echo Enter cost price of the item
read cp
echo Enter selling price of the item
read sp
if [ $sp -gt $cp ]
then
echo Seller has made profit
profit=$(( $sp - $cp ))
echo Profit = $profit
else
if [ $cp -gt $sp ]
then
echo Seller has incurred loss
loss=$(( $cp - $sp ))
echo Loss = $loss
else
echo No profit, No loss
fi
fi
```

**Q5. Write a script which receives two file names as arguments. It should check whether the contents of the two files are same or not. If they are same then second file should be deleted.**

**Sol.**

```
cmp -s $1 $2
if [ $? -eq 0 ]
then
rm $2
else
echo The contents of two files are not same
fi
```

**Q6. While executing a script either the LOGNAME or the UID is supplied at the command prompt. Write a script to find out how many terminals has this user logged in.**

**Sol.**

```
num=`who | grep $1 | wc -l`  
if [ $num -gt 0 ]  
then  
echo The user $1 has logged in at $num terminals  
else  
a=`grep $1 | /etc/passwd`  
  
oldifs="$IFS"  
IFS=":"  
set -- $a  
logname=$1  
num=`who | grep $1 | wc -l`  
echo The user $logname has logged in at $num terminals  
fi
```

**Q7. Write a script to read the contents from the file.**

**Sol.**

```
echo Enter File Name
read fname
terminal='tty'
exec < $fname
while read line
do
echo $line
done
```

**Q8. Write a program to calculate overtime pay of 5 employees. Overtime is paid at the rate of Rs.12/hr for every hour worked above 40 hours. Assume that employees do not work for fractional part of an hour.**

**Sol.**

```
emp=1
while [ $emp -le 5 ]
do
echo enter working hours
read hours
if [ $hours -gt 40 ]
then
otpay=$(((hours-40)*12))
echo overtime pay = Rs. $otpay
else
echo no overtime pay
fi
emp=$((emp+1))
done
```



**Q9. Write a program to find the factorial value of any number entered through the keyboard.**

**Sol.**

```
echo Enter any number  
read num  
i=1  
fact=1
```

```
while [ $i -le $num ]  
do  
fact=`echo $fact * $i | bc`  
i=`expr $i + 1`  
done  
echo factorial value of $num = $fact
```

**Q10. The marks obtained by a student in 5 different subjects are input through the keyboard. The student gets a division as per the following rules:**

- percentage above or equal to 60 – First Division**
- percentage between 50 and 59 –Second Division**
- percentage between 40 and 49 – Third Division**
- percentage less than 40 – Fail**

**Sol.**

```
echo Enter marks in five subjects
read m1 m2 m3 m4 m5
let per=($m1+$m2+$m3+$m4+$m5)/5
if [ $per -ge 60 ]
then
echo First division
else
echo second division
fi
# IF THEN
read m1 m2 m3
tot=$((m1+m2+m3))
avg=$((tot/3))
echo tot=$tot
if [ $avg -gt 60 ]
then
echo first division
else
echo second class
fi
```

**Q11. Write a script to receive an adapter name (MA or CGA or VGA or EGA) from the keyboard and then decide and print as per the following rule:**

**if MA it is monochrome adapter  
if CGA it is color graphics adapter  
if EGA it is enhanced graphics adapter  
if VGA it is video graphics adapter  
if SGA it is super graphics adapter**

**Sol.**

```
echo Enter the Adapter name ( MA/CGA/EGA/VGA/SGA)
read adapter
if [ $adapter = "MA" -o $adapter = "ma" ]
then
    echo You have a monochrome adapter
elif [ $adapter = "CGA" ]
    then echo You have a color graphics adapter
elif [ $adapter = "EGA" ]
    then echo You have a enhanced graphics adapter
elif [ $adapter = "VGA" ]
    then echo You have a video graphics adapter
elif [ $adapter = "SGA" ]
    then echo you have a super graphics adapter
else
    echo enter correct adapter name
fi
```

**Q12. Write a script to input a file name and check if file is readable, writable and executable or does not exist.**

**Sol.**

```
echo Enter any file name
read fname
if [ -e $fname ]
then
    if [ -r $fname -a -w $fname -a -x $fname ]
    then
        echo You have a read, write and execute permissions to $fname
    else
        echo Read, write and execute permissions denied
    fi
else
    echo Improper file name
fi
```

**Q 13. A friend of yours has promised to log in at a particular time. However, he has not kept the promise. You want to contact him as soon as he logs in. Write a shell script which checks after every 1 minute whether your friend has logged in or not. The Logname should be supplied to the shell script at command prompt**

**Sol :**

```
echo "Waiting for $1"
while [ 1 ]
do
    whoami|grep "$1" > /dev/null
    if [ $? -eq 0 ]
    then
        echo "$1 logged in"
        break
    fi
    sleep 60
done
```

**Ques 14: Write a menu driven program which has following options :**

1. Content of /etc/passwd
2. List of users who have currently logged in
2. Present Working Directory
3. Exit

**Sol :**

```
ch=1
while [ ch -eq 1 ]
do
echo "AIM : Menu Driven Program"

clear
tput cup 2 10
echo "1. Content of /etc/passwd"

tput cup 3 10
echo "2. List of users who have currently logged in"

tput cup 4 10
echo "3. Present Working Directory"

tput cup 5 10
echo "4. Exit"

read choice

case $choice in
    1)  tput cup 8 10
        echo "Enter the name of user"
        read user
        grep $user /etc/passwd
        ;;
    2)  whoami
        ;;
    3)  echo "Present Working Dir is"
        pwd
        ;;
    *)  echo "Invalid Choice"
        ;;
esac
echo "do you want to continue : press 1 for yes or 0 for no"
read ch
done
```

**Q 15 Write a script to do the following: the output of who should be sorted and displayed on the screen and the same output with the number of users should also be stored on a file "file1"**

**Sol.**

```
touch file1
whoami|sort
whoami|sort>file1
whoami >>file1
cat file1
```

# User Account Management

## User Management Commands

### **useradd**

This command add users to a Linux system.

```
useradd user_name
```

```
passwd user_name
```

The above command creates a new account (This requires login through root).

After creating a new account,set up the password for the new user.

The user home directory (which is created) is */home/user\_name*.

### **adduser**

Another option is to use the command adduser. It has a more user friendly interface than useradd. The program prompts for the username, user ID, Group ID, Home directory, and shell.

```
adduser user_name
```

### **userdel**

This command is used to delete users from a system.( It requires login through root password)

```
userdel -r login
```

Deletes entries in the system account files that refer to the login.

```
userdel user_name
```



## **deluser - Debian specific**

It removes the home directory, or all files on the system owned by the user.

<b>Command</b>	<b>Action</b>
<code>deluser</code>	Does not remove the home directory, mail spool, or any files owned by the user
<code>deluser -remove-home user</code>	Removes the home directory and mail spool
<code>deluser -remove-all-files user</code>	Removes all files on the system owned by the user

## **usermod**

This command is used to modify user accounts without manual editing of the files `/etc/passwd` `/etc/shadow` `/etc/group` and `/etc/gshadow`. (It requires login through root password )

## **passwd**

Change the password on your current account. If you are root, you can change the password for any user using:

```
passwd user_name
```

**chage -M 100 *login\_name***

(= "change age"). Set the password expiry to 100 days for the user named *login\_name* .

## User system files

- ***Format of /etc/passwd***

The /etc/passwd files gives a listing of the users that have accounts on the system. The file has the following fields

username:password:uid:gid:user\_info:home\_directory:shell\_type

<b>username</b>	- The name the person logs in with
<b>password</b>	- Typically x to indicate a password is set
<b>uid</b>	- The system number assigned to a user to indicate which files they own or have access to.
<b>gid</b>	- The group id number which is the principle group they belong to.
<b>user_info</b>	- Information about user separated by a comma. Possible information includes: Full name, Room number, Work number, Home phone, and Other
<b>home_directory</b>	- The directory that the user owns. Typically /home/username
<b>shell_type</b>	- The default shell setting is /bin/bash.

- ***Format of /etc/shadow***

The /etc/shadow file is the encrypted password file. The file has the following fields.

username:encrypted password:11843:0:99999:7:::

<b>username</b>	- The name the person logs in with
<b>encrypted password</b>	- Typically x to indicate a password is set
<b>##:##:##:::</b>	- Contains information about the number of days since the password was changed, when it expires or is disabled.

- ***Format of /etc/shadow - encrypted Samba password file***

login name:encrypted password:information about days since password changed and when expired or disabled

## **Linux User Management Commands**

<b>Ac</b>	: Print statistics about users' connect time.
<b>Adduser</b>	: Ex: adduser mark - Effect: Adds a user to the system named mark
<b>Chage</b>	: Used to change the time the user's password will expire.
<b>Chpasswd</b>	: Update password file in batch.
<b>Chroot</b>	: Run command or interactive shell with special root directory.
<b>Chsh</b>	: Change the login shell.
<b>Edquota</b>	Used to edit user or group quotas. This program uses the vi editor to edit the quota.user and quota.group files. If the environment variable EDITOR is set to emacs, the emacs editor will be used. Type "export EDITOR=emacs" to set that variable.
<b>Id</b>	: Print real and effective user id and group ids.
<b>Last</b>	: Display the last users logged on and how long.
<b>Lastcomm</b>	: Display information about previous commands in reverse order. Works only if process accounting is on.
<b>Newusers</b>	: Update and create newusers in batch.
<b>Passwd</b>	: Set a user's pass word.
<b>Pwck</b>	: Verify integrity of password files.
<b>Pwconv</b>	: Convert to and from shadow passwords and groups.
<b>Quota</b>	: Display users' limits and current disk usage.
<b>Quotaoff</b>	: Turns system quotas off.
<b>Quotaon</b>	: Turns system quotas on.
<b>Quotacheck</b>	: Used to check a filesystem for usage, and update the quota.user file.
<b>Repquota</b>	: Lists a summary of quota information on filesystems.
<b>Sa</b>	: Generates a summary of information about users' processes that are stored in the /var/log/pacct file.
<b>Useradd</b>	: Create a new user or update default new user information.
<b>Userdel</b>	: Delete a user account and related files.
<b>Usermod</b>	: Modify a user account.
<b>Users</b>	: Print the user names of users currently logged in.
<b>Utmpdump</b>	: Used for debugging.
<b>W</b>	: Display users logged in and what they are doing.
<b>Wall</b>	: Send a message to everybody's terminal.
<b>Who</b>	: Display the users logged in.
<b>Who am i</b>	: Print effective user id.

# **Group Management**

## ***Group Management Commands***

### **groupadd - Adding Groups**

groupadd - using default values from the system and command line specified values groupadd creates groups on the system.

```
groupadd group_name
```

the specified gid should be unique.

### **addgroup**

addgroup - friendlier front ends - the group is created with no users, two non-option arguments will add an existing user addgroup --system group common options

### **delgroup**

This command will delete a system group that is not the primary group of a user. A system group would be one created with the addgroup command.

### **groupmod - Tool to modify groups**

Once groups are created on the computer they will need to be modified and edited at some point. It modify groups without manual editing of the files `/etc/passwd` `/etc/shadow` `/etc/group` and `/etc/gshadow`. The groupmod command will change the group id associated with a group or the name of the group.

```
groupmod [-g gid ] group
```

- -g gid -group's numerical identification number. [any files which the old group ID is the file group ID must have the file group ID changed manually.
- -n groupname - this option changes the name of the group from the initial group name to the groupname specified in the command

Example command	Resulting action
<code>groupmod -g 120 example</code>	Changes the group id number to 120 for group "example"
<code>groupmod -n group2 group</code>	Changes the group name of group to group2

Groups can be added and modified by directly access in the /etc/groups folder and modifying it. If a group is modified in this manner, chmod should be run to change the group attributes of files that were associated with the group ID.

*To change the permissions on a selected file so that the file belongs to me AND the group "friends".*

`chgrp friends my_file`(it changes the permission on a file (my\_file) so that now this file can be accessed by 'owner ' of the file and group 'friends') the listed members of the group(friends) have special access to these files that the rest of the world might not have, for example read and write permission:

`chmod g=rw,o= my_file`

## groups

List the groups to which the current user belongs.

Or

use `'groups john'` to find to which groups the user john belongs.

## **Linux Group Management Commands**

**chgrp** : Changes the group ownership of files.  
**gpasswd** : Administer the /etc/group file.  
**groupadd** : Create a new group.  
**grpck** : Verify the integrity of group files.  
**grpconv** : Creates /etc/gshadow from the file /etc/group which converts to shadow passwords.  
**groupdel** : Delete a group.  
**groupmod** : Modify a group.  
**groups** : Print the groups a user is in  
**id** : Print real and effective user id and group ids.  
**last** : Display the last users logged on and how long.  
**newgrp** : Lets a user log in to a new group.  
**newusers** : Update and create newusers in batch.  
: a user's pass word.  
**repquota** : Lists a summary of quota information on filesystems.  
**vigr** : Edit the password or group files.  
**vipw** : Edit the password or group files.

## **Group system files**

### **Format of /etc/group**

The /etc/group files are configured with the same general format across platforms. It lists the groups that have been created on the system and who is in each group. Groups can be managed by creating new lines in the file following the convention, users can be added to an already existing group by adding the new username to the end of the other usernames or after the semicolon of the gid if there are no other members.

The fields in the /etc/group are:

username:password:gid:username1,username2,username3

**username** - The name the person logs in with  
**password** - An x indicates a password is set and if left blank no password has been set.  
**gid** - The group id number which is the principle group they belong to.  
**usernames** - Members of the group separated by a comma

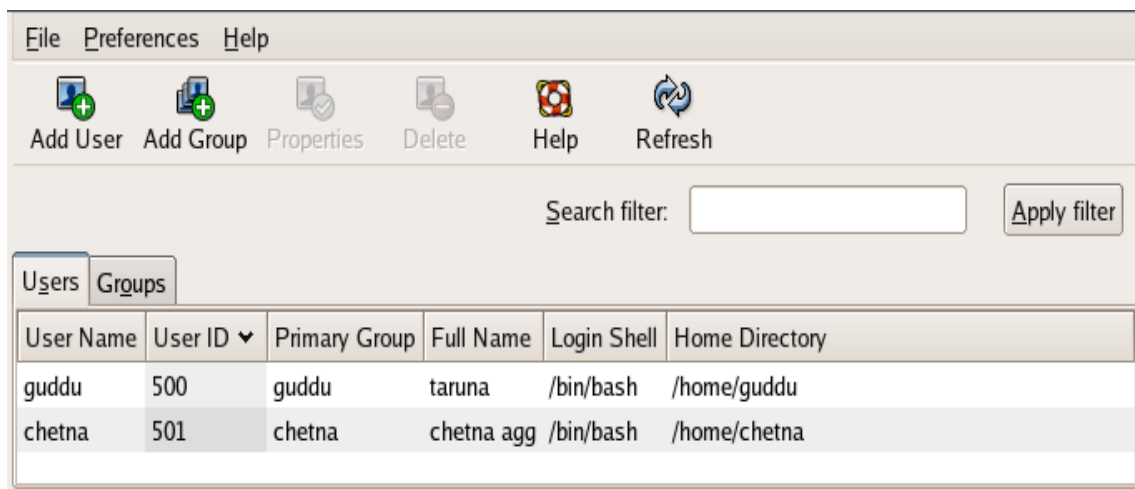
## **GUI User and Group Management Tool**

# Red Hat User Manager

The User Manager allows you to view, modify, add, and delete local users and groups.

To use the **User Manager**, you must be running the X Window System, have root privileges, and have the **system-config-users** RPM package installed.

To start the **User Manager** from the desktop, go to the **Main Menu Button** (on the Panel) => **System Settings** => **Users & Groups**. Or, type the command **system-config-users** at a shell prompt (for example, in an XTerm or a GNOME terminal).



**Figure 1. User Manager**

## Adding a New User

To add a new user, click the Add User button. A window as shown in [Figure 2](#) will appear. Type the username and full name for the new user in the appropriate fields.

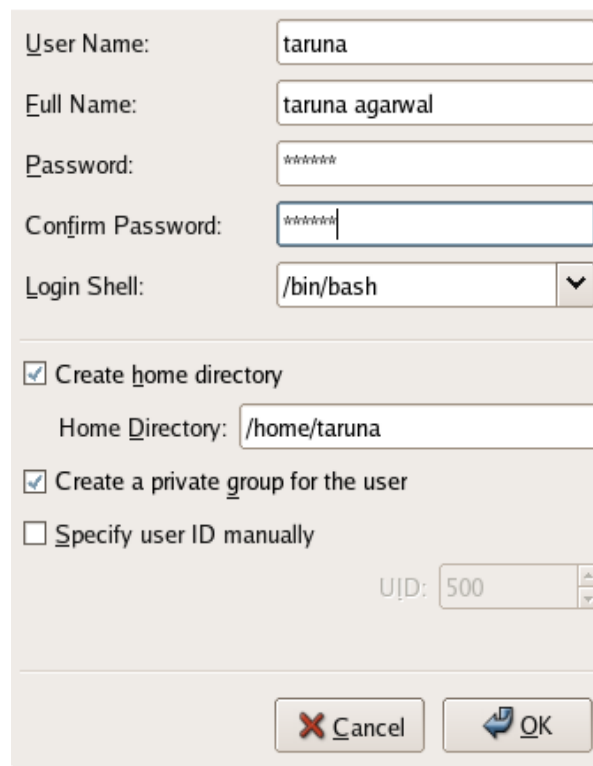
Select a login shell. If you are not sure which shell to select, accept the default value of `/bin/bash`. The default home directory is **`/home/username`**.

If you select to **create the home directory**, default configuration files are copied from the `/etc/skel` directory into the new home directory.

Whenever you create a new user, by default, a unique group with the same name as the user is created. If you do not want to create this group, unselect **Create a private group for the user** for the user.

To specify a user ID for the user, select **Specify user ID manually**. If the option is not selected, the next available user ID starting with number 500 will be assigned to the new user. Fedora Core reserves user IDs below 500 for system users.

Click **OK** to create the user.



User Name: taruna

Full Name: taruna agarwal

Password: \*\*\*\*\*

Confirm Password: \*\*\*\*\*

Login Shell: /bin/bash

☒ Create home directory

Home Directory: /home/taruna

☒ Create a private group for the user

☐ Specify user ID manually

UID: 500

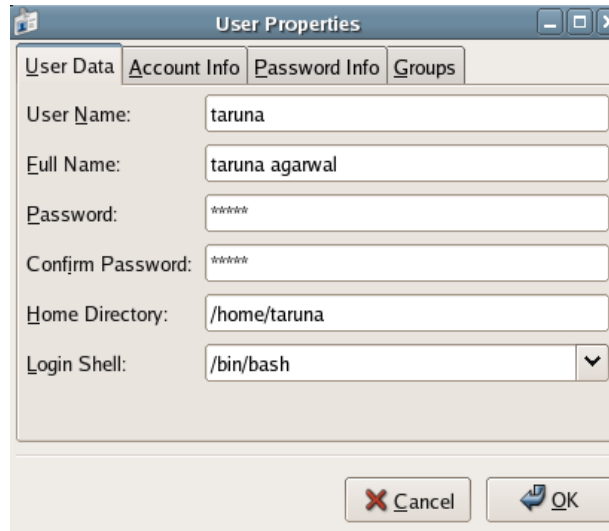
Cancel OK

Figure 2: New User

## Modifying User Properties



To view the properties of an existing user, click on the **Users** tab, select the user from the user list, and click **Properties** from the button menu (or choose **File => Properties** from the pulldown menu). A window similar to [Figure 3](#) will appear.



**Figure 3. User Properties**

The **User Properties** window is divided into multiple tabbed pages:

4. **User Data** — Shows the basic user information configured when you added the user. Use this tab to change the user's full name, password, home directory, or login shell.
5. **Account Info** — Select **Enable account expiration** if you want the account to expire on a certain date. Enter the date in the provided fields. Select **User account is locked** to lock the user account so that the user cannot log in to the system.
6. **Password Info** — This tab shows the date that the user's password last changed. To force the user to change passwords after a certain number of days, select **Enable password expiration**. The number of days before the user's password expires, the number of days before the user is warned to change passwords, and days before the account becomes inactive can also be changed.
7. **Groups** — Select the groups that you want the user to be a member of and the user's primary

## **Adding a New Group**

To add a new user group, click the **Add Group** button. A window similar to [Figure 4](#) appears. Type the name of the new group to create. To specify a group ID for the new group, select **Specify group ID manually** and select the GID. Fedora Core reserves group IDs lower than 500 for system groups.

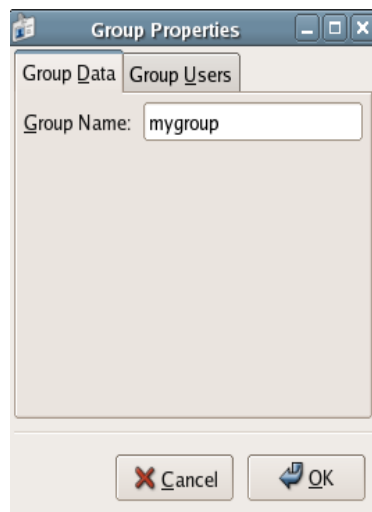
Click **OK** to create the group. The new group will appear in the group list.



**Figure 4. New Group**

## **Modifying Group Properties**

To view the properties of an existing group, select the group from the group list and click **Properties** from the button menu (or choose **File** => **Properties** from the pulldown menu). A window similar to [Figure 5](#) will appear



**Figure 5. Group Properties**

The **Group Users** tab displays which users are members of the group. Select additional users to be added to the group, or unselect users to be removed from the group. Click **OK** or **Apply** to modify the users in the group.

Serial no.	Topic	Teacher's Signature	Page no.
1	Installation		
2	User management		

3	Group management		
4	Disk management		
5	Basic salary of A is input through the keyboard. His dearness allowance is 40% of basic salary and house rent allowance is 20% of basic salary. Write a script to calculate his gross salary.		
6	If a five digit number is input through the keyboard, write a script to calculate the sum of its digits.		
7	Write a script which will receive either the filename or the filename with its full path during execution. This script should obtain information about this file as given by ls -l and display it in proper format.		
8	If cost price and selling price of an item is input through the keyboard, write a script to determine whether the seller has made profit or incurred loss. Also determine how much profit was made or loss incurred		
9	Write a script which receives two file names as arguments. It should check whether the contents of the two files are same or not. If they are same then second file should be deleted.		
10	While executing a script either the LOGNAME or the UID is supplied at the command prompt. Write a script to find out how many terminals has this user logged in		
11	Write a script to read the contents from the file		
12	Write a program to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs.12/hr for every hour worked above 40 hours. Assume that employees do not work for fractional part of an hour		
13	Write a program to find the factorial value of any number entered through the keyboard.		
14	The marks obtained by a student in 5 different subjects are input through the keyboard & display students division.		

## **INDEX**

15	Write a script to receive an adapter name (MA or CGA or VGA or EGA) from the keyboard and then decide and print as per the following rule:		
----	--	--	--

	if MA it is monochrome adapter if CGA it is color graphics adapter if EGA it is enhanced graphics adapter if VGA it is video graphics adapter		
16	Write a script to input a file name and check if file is readable, writable, or both or executable or does not exist .		
17	A friend of yours has promised to log in at a particular time. However, he has not kept the promise. You want to contact him as soon as he logs in. Write a shell script which checks after every 1 minute whether your friend has logged in or not. The Logname should be supplied to the shell script at command prompt		