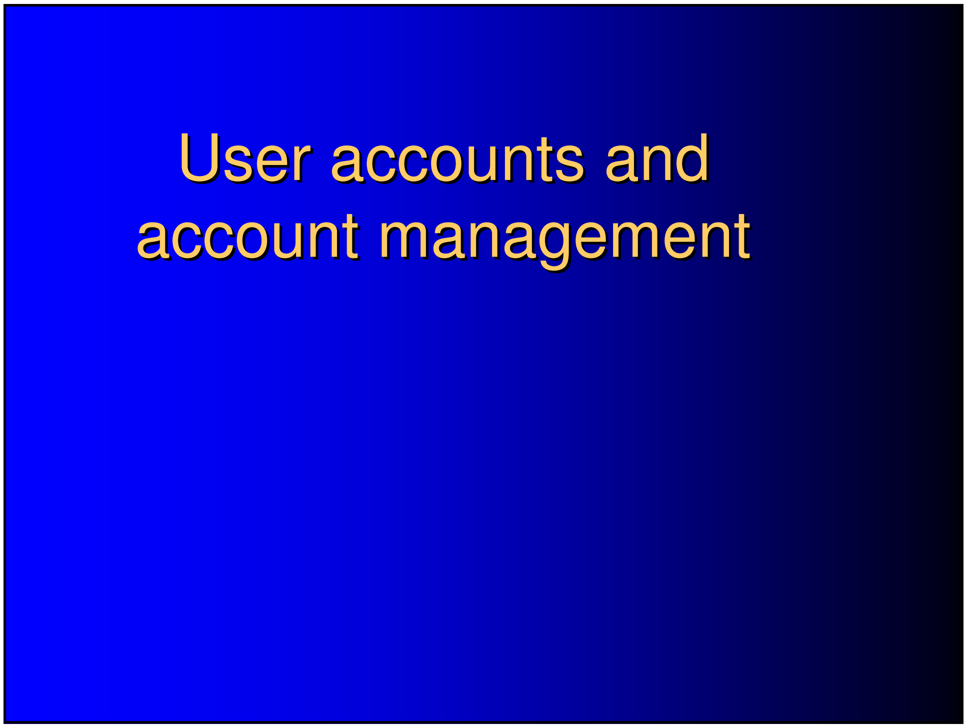
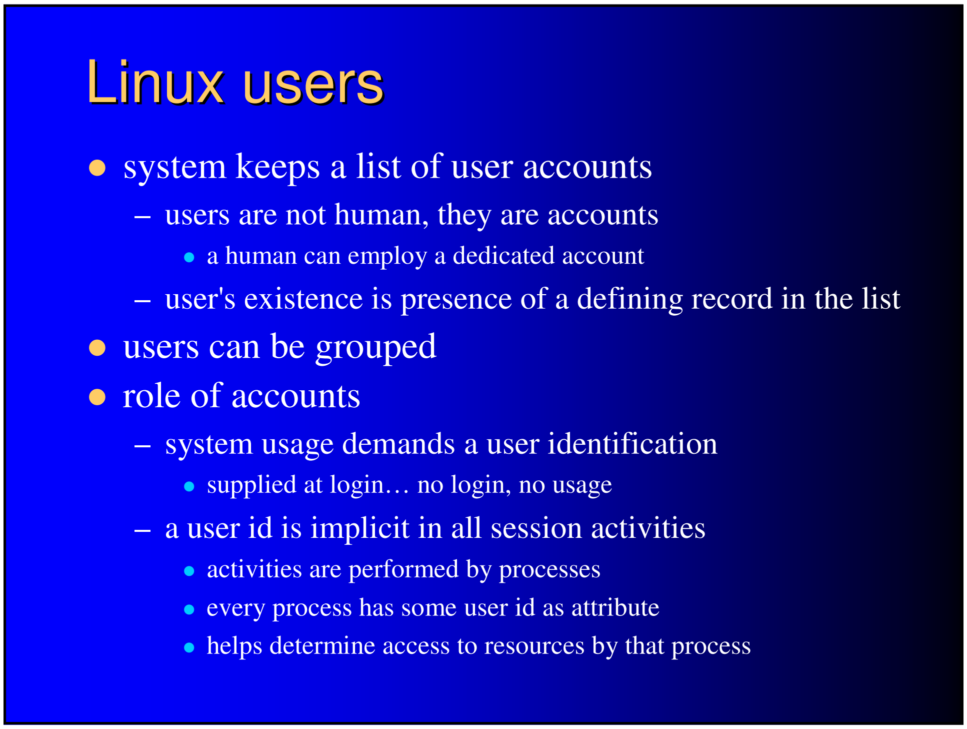
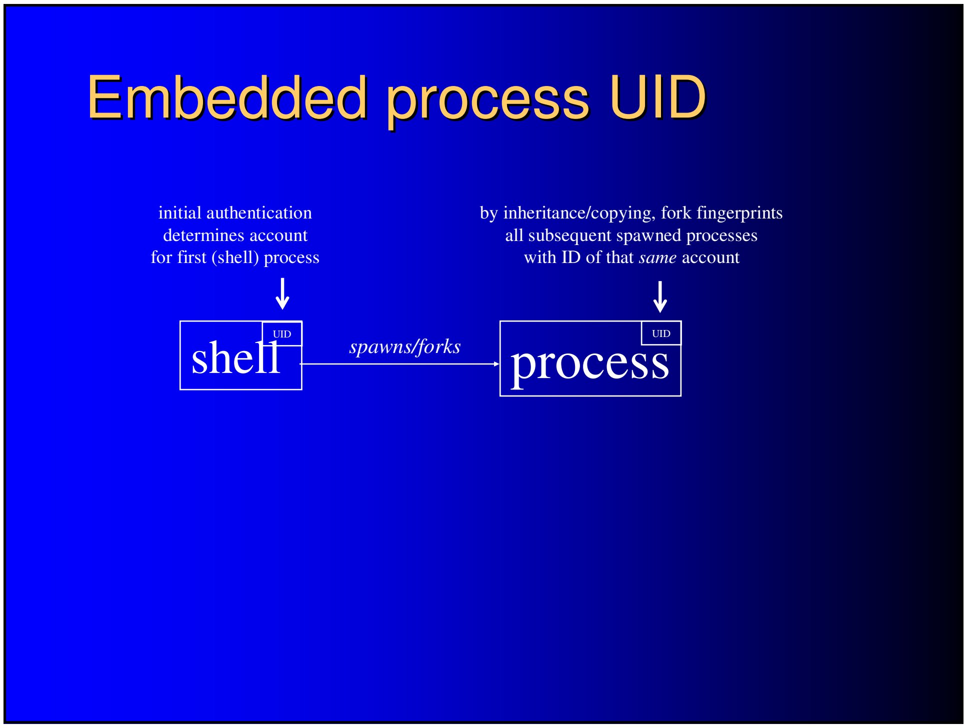


User accounts and account management



Linux users

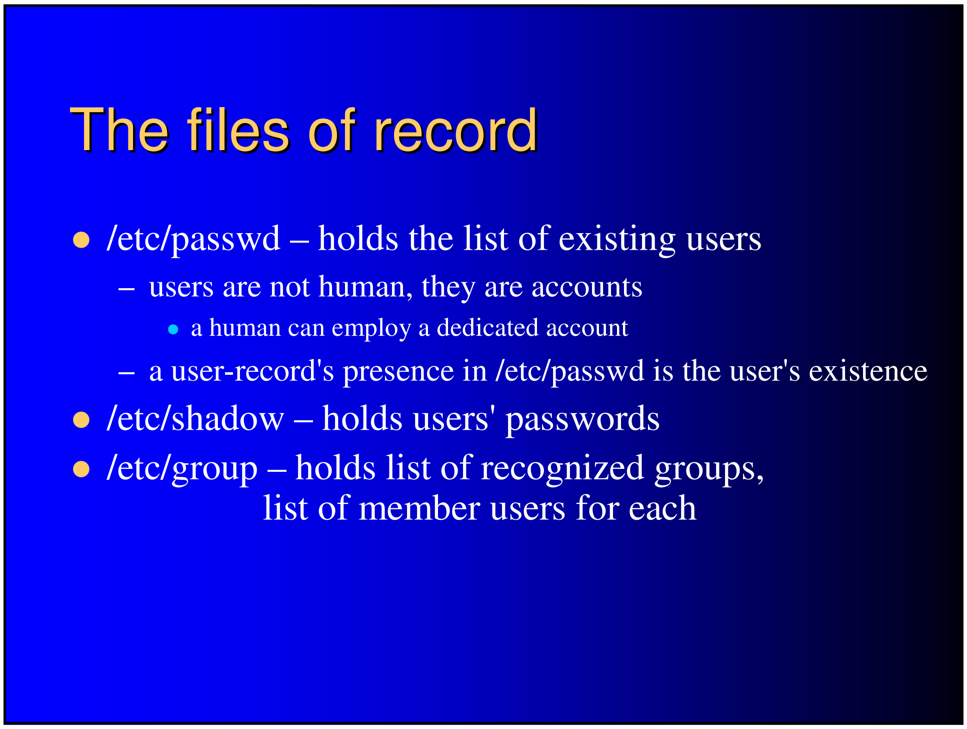
- system keeps a list of user accounts
  - users are not human, they are accounts
    - a human can employ a dedicated account
  - user's existence is presence of a defining record in the list
- users can be grouped
- role of accounts
  - system usage demands a user identification
    - supplied at login… no login, no usage
  - a user id is implicit in all session activities
    - activities are performed by processes
    - every process has some user id as attribute
    - helps determine access to resources by that process



Embedded process UID

initial authentication
determines account
for first (shell) process

by inheritance/copying, fork fingerprints
all subsequent spawned processes
with ID of that *same* account



The files of record

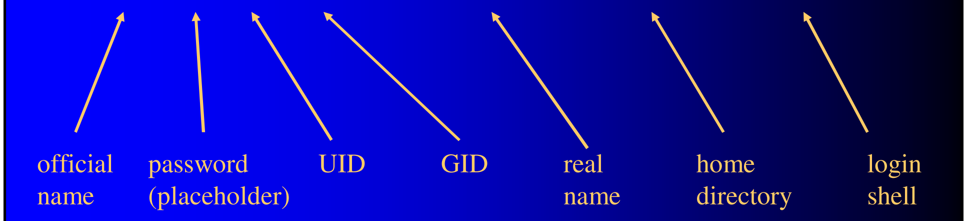
- /etc/passwd – holds the list of existing users
  - users are not human, they are accounts
    - a human can employ a dedicated account
  - a user-record's presence in /etc/passwd is the user's existence
- /etc/shadow – holds users' passwords
- /etc/group – holds list of recognized groups,
  list of member users for each

Editing the files of record safely

- plain editors invite introducing errors and conflicts
- /etc/passwd – use usermod or vipw
- /etc/shadow – use passwd, chage, usermod
- /etc/group – use groupmod and usermod, or vigr

/etc/passwd entries hold user information

```
craig:x:507:507:Craig Smith:/home/craig:/bin/bash
```



official name

password (placeholder)

UID

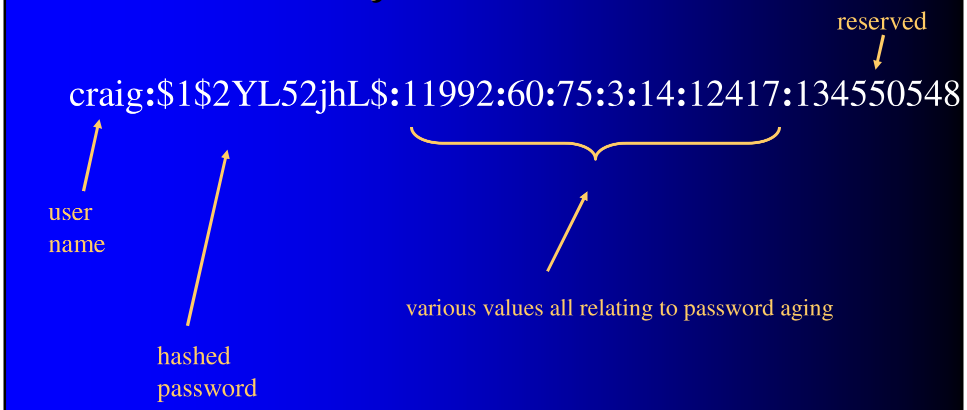
GID

real name

home directory

login shell

/etc/shadow entries hold ancillary user information



```
craig:$1$2YL52jhL$:11992:60:75:3:14:12417:134550548
```

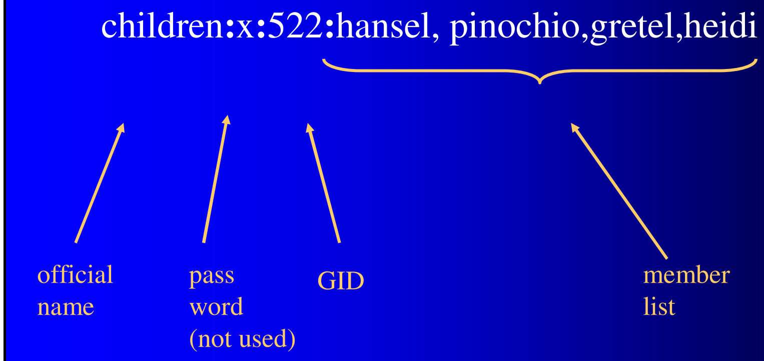
reserved

user name

hashed password

various values all relating to password aging

/etc/group entries hold group information



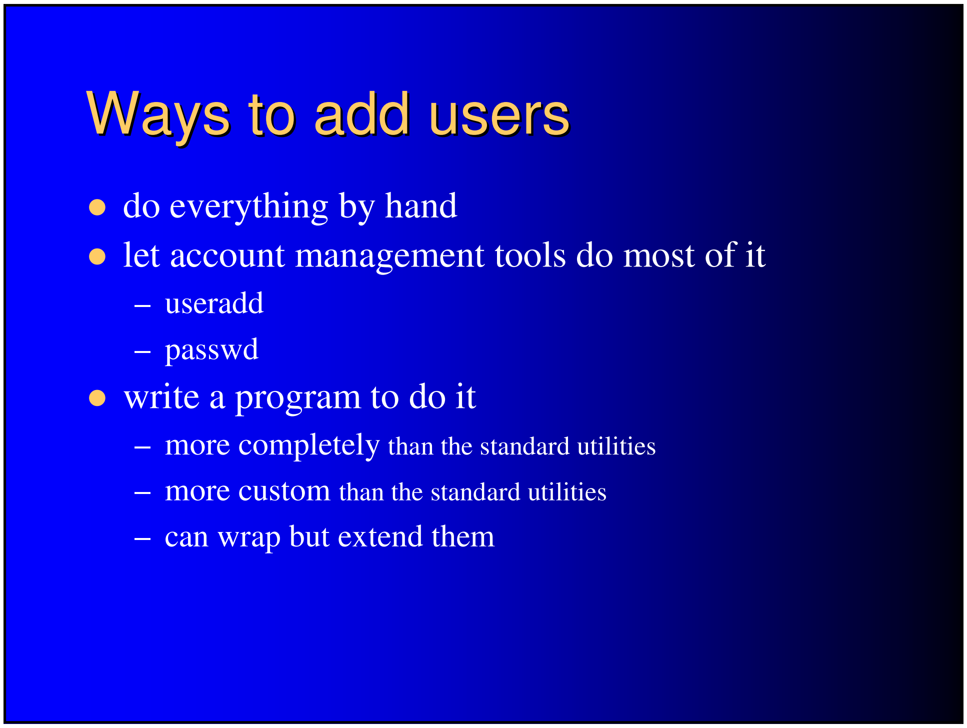
```
children:x:522:hansel, pinochio,gretel,heidi
```

official name

pass word (not used)

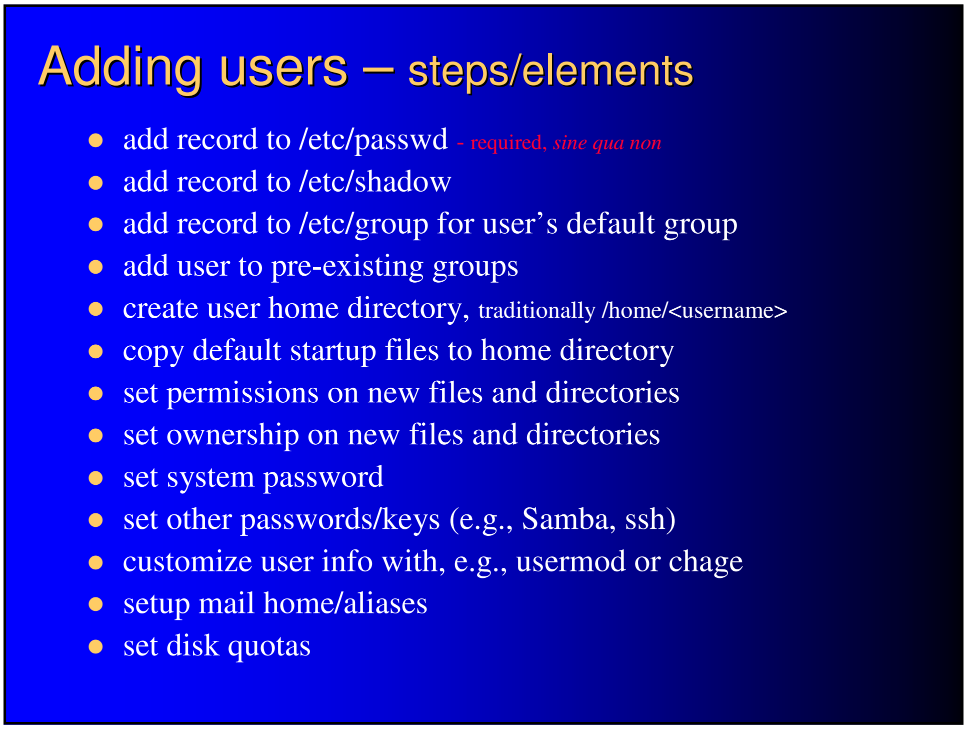
GID

member list



Ways to add users

- do everything by hand
- let account management tools do most of it
  - useradd
  - passwd
- write a program to do it
  - more completely than the standard utilities
  - more custom than the standard utilities
  - can wrap but extend them



Adding users – steps/elements

- add record to /etc/passwd - required, *sine qua non*
- add record to /etc/shadow
- add record to /etc/group for user's default group
- add user to pre-existing groups
- create user home directory, traditionally /home/<username>
- copy default startup files to home directory
- set permissions on new files and directories
- set ownership on new files and directories
- set system password
- set other passwords/keys (e.g., Samba, ssh)
- customize user info with, e.g., usermod or chage
- setup mail home/aliases
- set disk quotas

Standard account management tools - adding user in 2 steps

- use useradd
- then set password with passwd

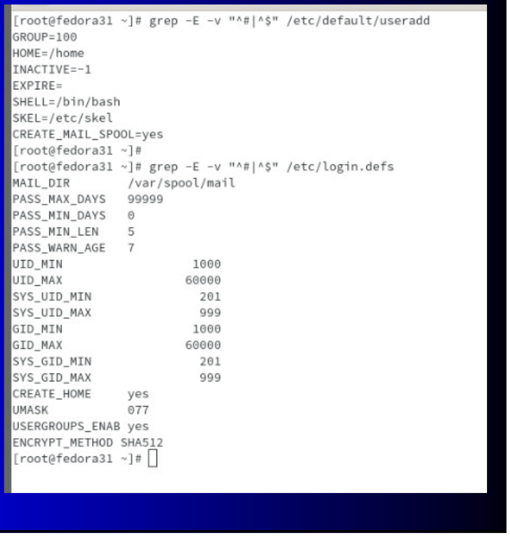
- does *some* of the steps
  - most
  - but not all

Steps performed by useradd or passwd commands

useradd	passwd	
✓		add record to /etc/passwd - required, *sine qua non*
✓		add record to /etc/shadow
✓		add record to /etc/group for user's default group
		add user to pre-existing groups
✓		create user home directory, traditionally /home/<username>
✓		copy default startup files to home directory
✓		set permissions on new files and directories
✓		set ownership on new files and directories
	✓	set system password
		set other passwords/keys (e.g., Samba, ssh)
		customize user info with, e.g., usermod or chage
		setup mail home/aliases
		set disk quotas

Standard tools' options

- by command line
- by tools' defaults
  - /etc/login.defs
  - /etc/defaults/useradd



```
[root@fedora31 ~]# grep -E -v "^#|^$" /etc/default/useradd
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
[root@fedora31 ~]#
[root@fedora31 ~]# grep -E -v "^#|^$" /etc/login.defs
MAIL_DIR        /var/spool/mail
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7
UID_MIN                  1000
UID_MAX                 60000
SYS_UID_MIN               201
SYS_UID_MAX               999
GID_MIN                  1000
GID_MAX                 60000
SYS_GID_MIN               201
SYS_GID_MAX               999
CREATE_HOME     yes
UMASK           077
USERGROUPS_ENAB yes
ENCRYPT_METHOD SHA512
[root@fedora31 ~]# 
```

Adding users in batch mode

Make a file listing users in the form username:password

e.g., file “userinfo”

```
able:apple
baker:banana
charlie:cantelope
```

Assigning passwords in batch mode with chpasswd command

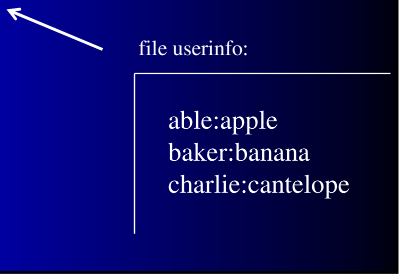
man chpasswd:

"chpasswd reads a file of user name and password pairs from standard input and uses this information to update a group of existing users. …
[but] The named user must exist."

Solution: make the named users exist first, with a script that "useradd"s them by looping through the list, then feed the list to chpasswd

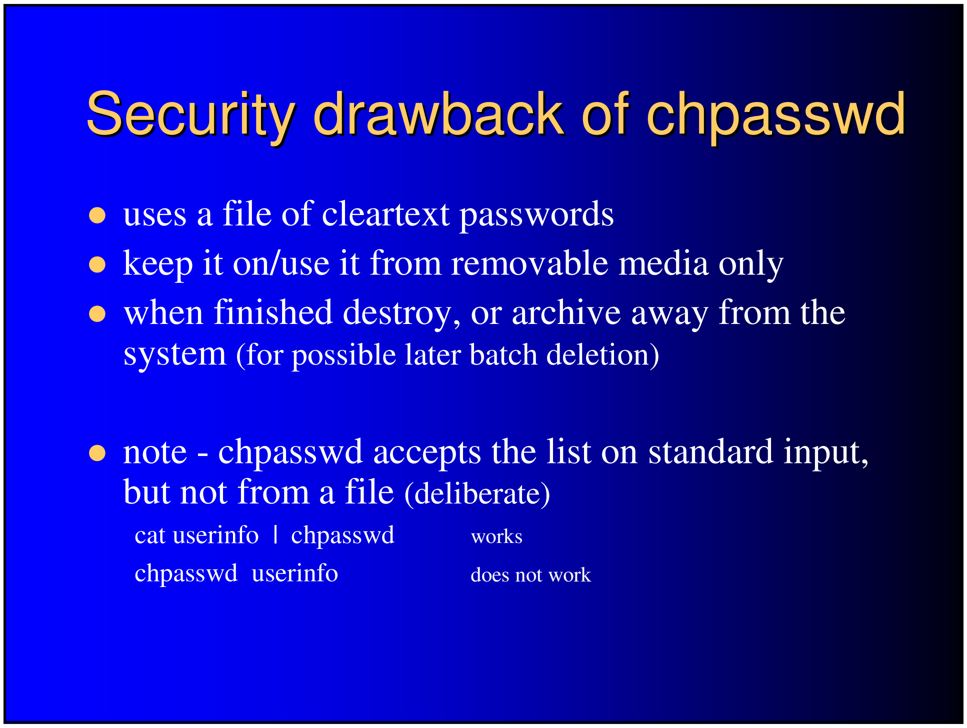
Minimal custom script

```
#!/bin/bash
while read LINE
do
        user=`echo $LINE | cut -f 1 -d :`
        useradd $user
done < userinfo

cat userinfo | chpasswd
```



file userinfo:

```
able:apple
baker:banana
charlie:cantelope
```

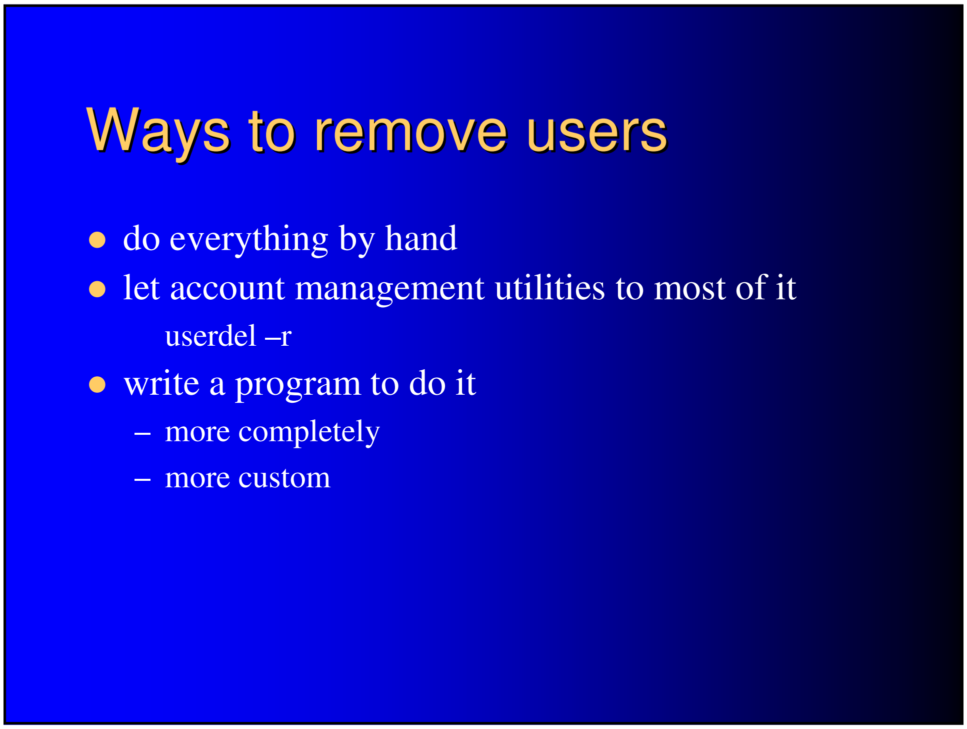


Security drawback of chpasswd

- uses a file of cleartext passwords
- keep it on/use it from removable media only
- when finished destroy, or archive away from the system (for possible later batch deletion)

- note - chpasswd accepts the list on standard input, but not from a file (deliberate)

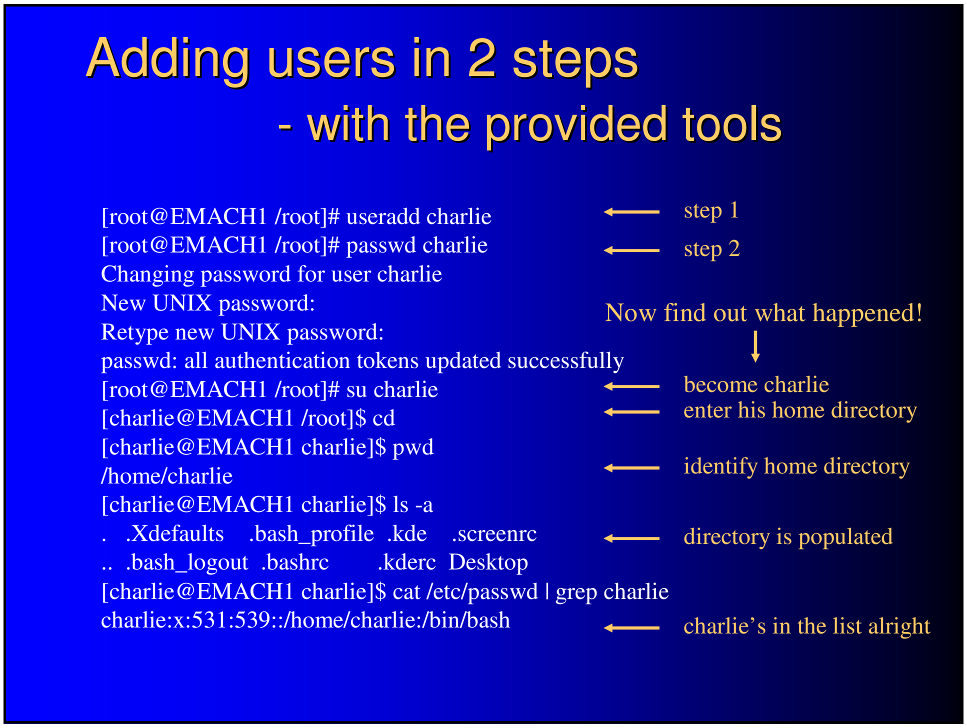
```
cat userinfo | chpasswd          works
chpasswd  userinfo               does not work
```



Ways to remove users

- do everything by hand
- let account management utilities to most of it

  userdel –r
- write a program to do it
  - more completely
  - more custom



Adding users in 2 steps
- with the provided tools

```
[root@EMACH1 /root]# useradd charlie
[root@EMACH1 /root]# passwd charlie
Changing password for user charlie
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
[root@EMACH1 /root]# su charlie
[charlie@EMACH1 /root]$ cd
[charlie@EMACH1 charlie]$ pwd
/home/charlie
[charlie@EMACH1 charlie]$ ls -a
.   .Xdefaults    .bash_profile  .kde    .screenrc
..  .bash_logout  .bashrc        .kderc  Desktop
[charlie@EMACH1 charlie]$ cat /etc/passwd | grep charlie
charlie:x:531:539::/home/charlie:/bin/bash
```

step 1

step 2

Now find out what happened!

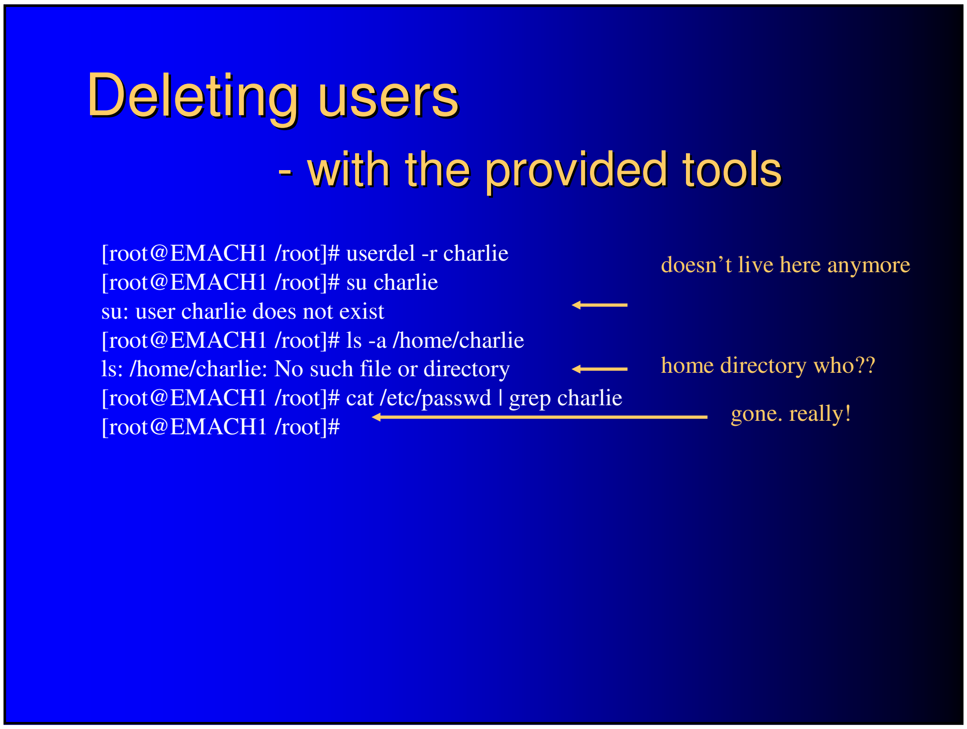
become charlie

enter his home directory

identify home directory

directory is populated

charlie’s in the list alright



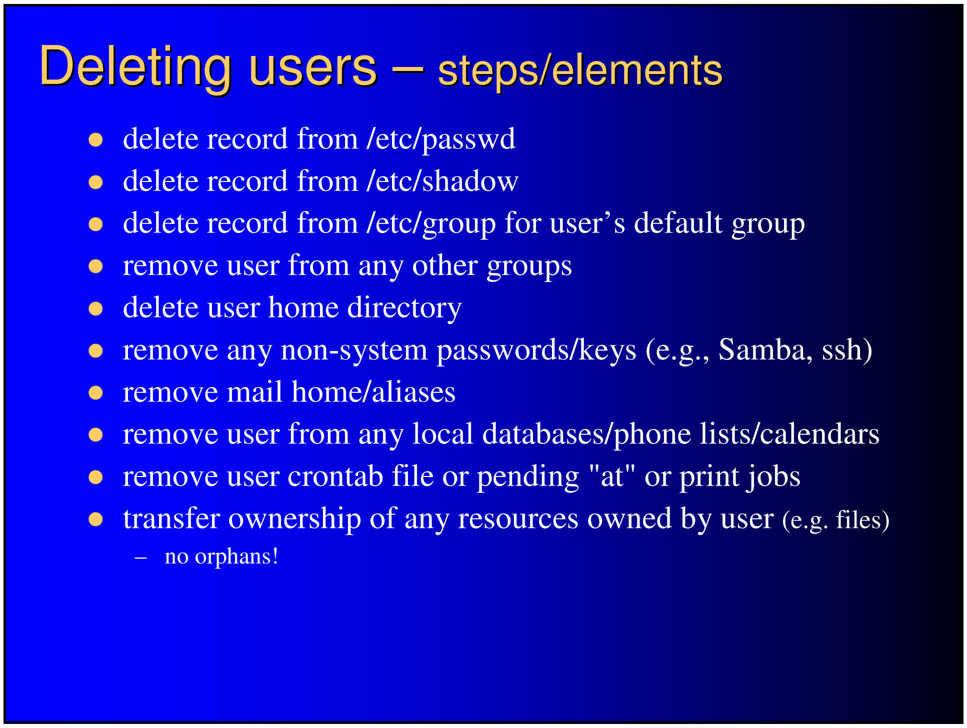
Deleting users
- with the provided tools

```
[root@EMACH1 /root]# userdel -r charlie
[root@EMACH1 /root]# su charlie
su: user charlie does not exist
[root@EMACH1 /root]# ls -a /home/charlie
ls: /home/charlie: No such file or directory
[root@EMACH1 /root]# cat /etc/passwd | grep charlie
[root@EMACH1 /root]#
```

doesn’t live here anymore

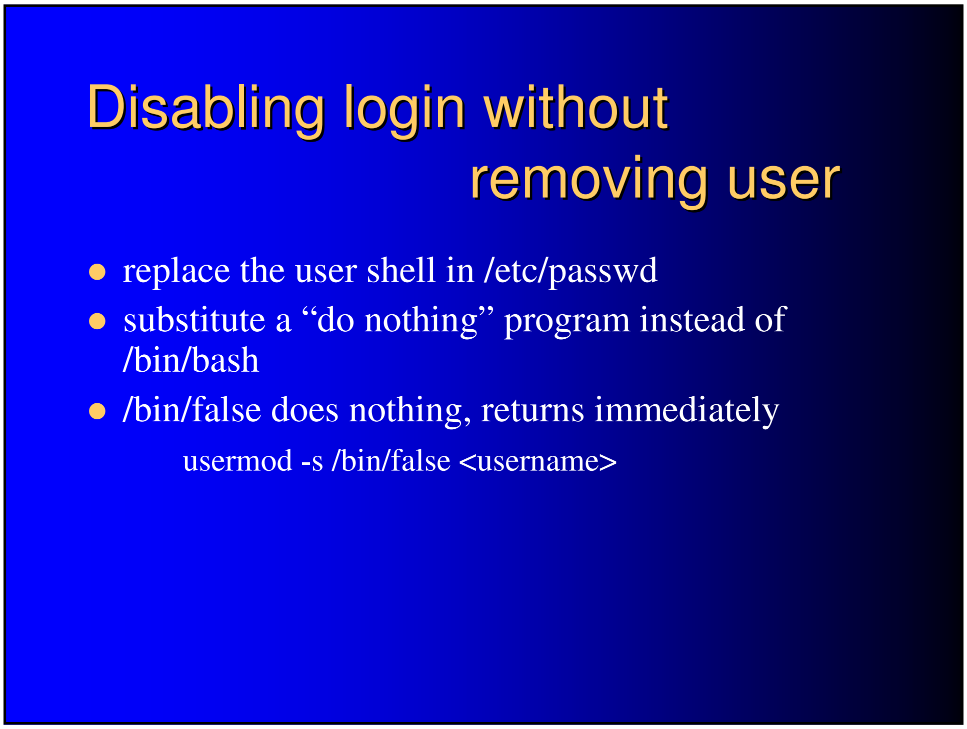
home directory who??

gone. really!



Deleting users – steps/elements

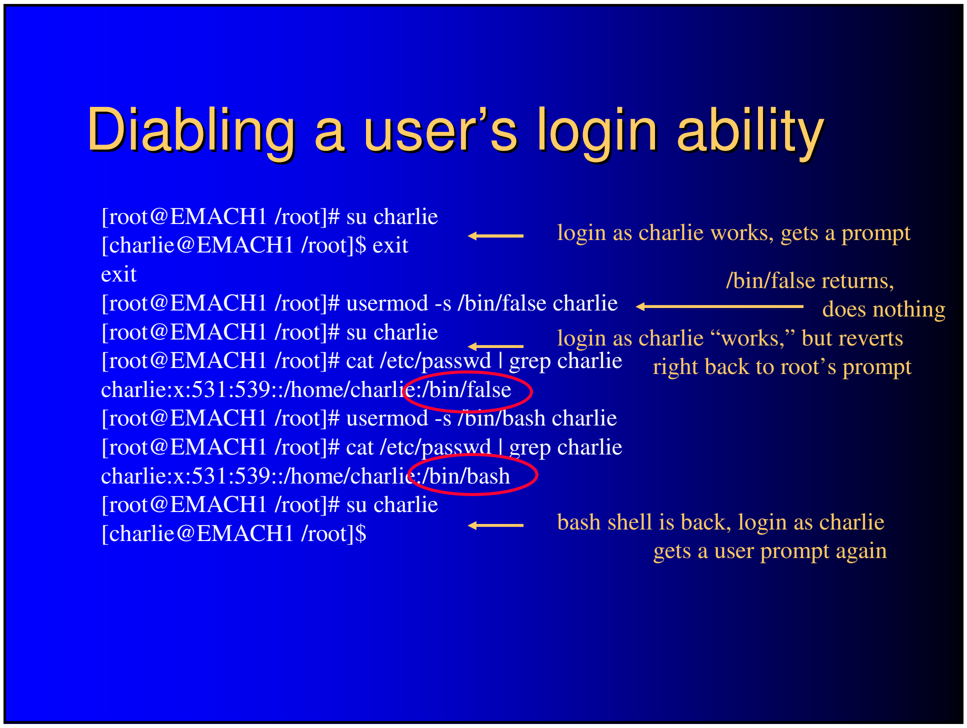
- delete record from /etc/passwd
- delete record from /etc/shadow
- delete record from /etc/group for user's default group
- remove user from any other groups
- delete user home directory
- remove any non-system passwords/keys (e.g., Samba, ssh)
- remove mail home/aliases
- remove user from any local databases/phone lists/calendars
- remove user crontab file or pending "at" or print jobs
- transfer ownership of any resources owned by user (e.g. files)
  - no orphans!



Disabling login without removing user

- replace the user shell in /etc/passwd
- substitute a "do nothing" program instead of /bin/bash
- /bin/false does nothing, returns immediately

```
usermod -s /bin/false <username>
```



Diabling a user’s login ability

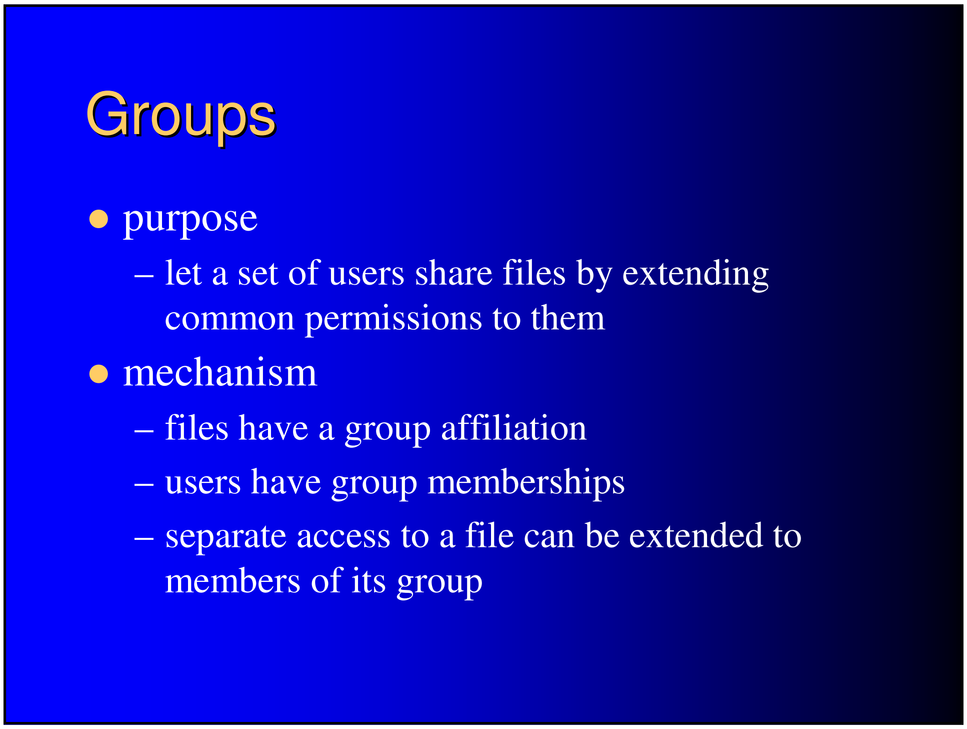
```
[root@EMACH1 /root]# su charlie
[charlie@EMACH1 /root]$ exit
exit
[root@EMACH1 /root]# usermod -s /bin/false charlie
[root@EMACH1 /root]# su charlie
[root@EMACH1 /root]# cat /etc/passwd | grep charlie
charlie:x:531:539::/home/charlie:/bin/false
[root@EMACH1 /root]# usermod -s /bin/bash charlie
[root@EMACH1 /root]# cat /etc/passwd | grep charlie
charlie:x:531:539::/home/charlie:/bin/bash
[root@EMACH1 /root]# su charlie
[charlie@EMACH1 /root]$
```

login as charlie works, gets a prompt

/bin/false returns,
does nothing

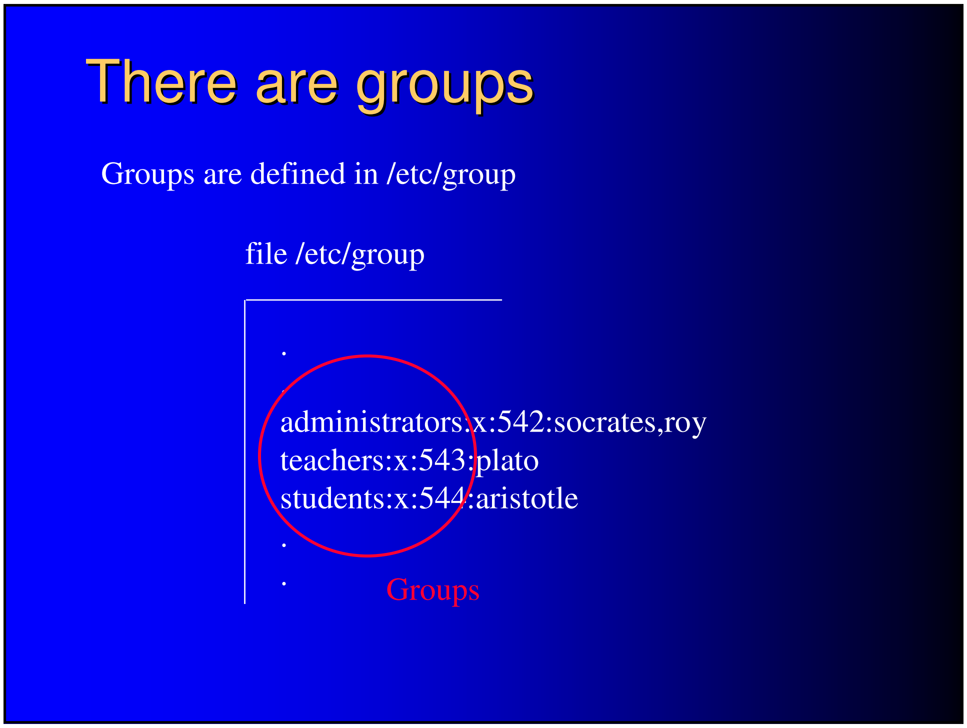
login as charlie “works,” but reverts
right back to root’s prompt

bash shell is back, login as charlie
gets a user prompt again



Groups

- purpose
  - let a set of users share files by extending common permissions to them
- mechanism
  - files have a group affiliation
  - users have group memberships
  - separate access to a file can be extended to members of its group



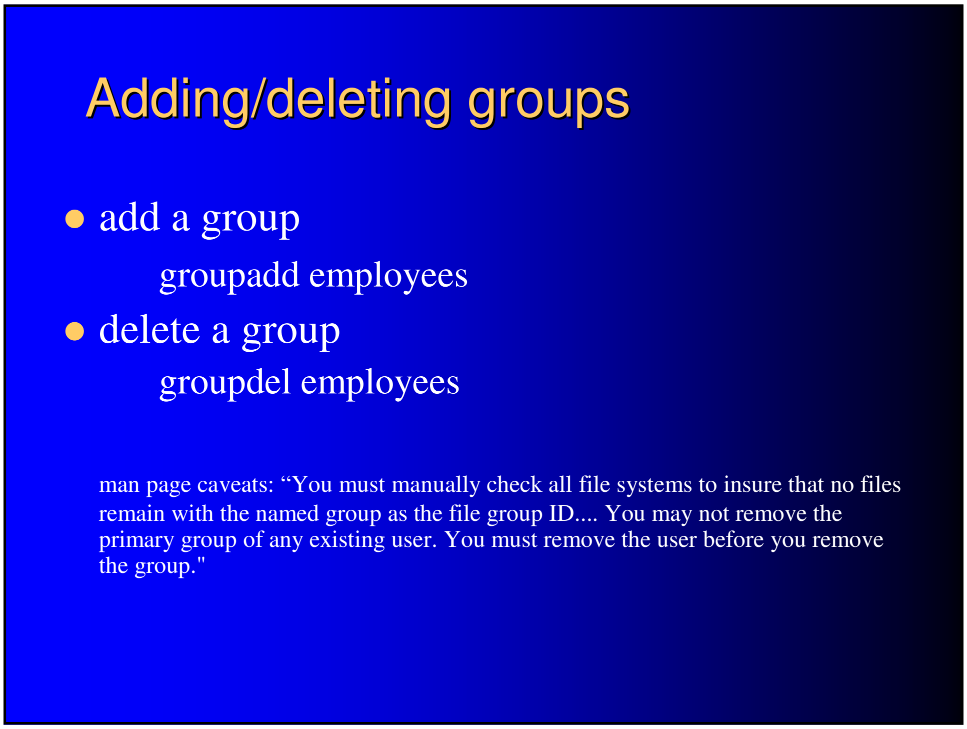
There are groups

Groups are defined in /etc/group

file /etc/group

```
.
.
administrators:x:542:socrates,roy
teachers:x:543:plato
students:x:544:aristotle
.
.
```

Groups



Adding/deleting groups

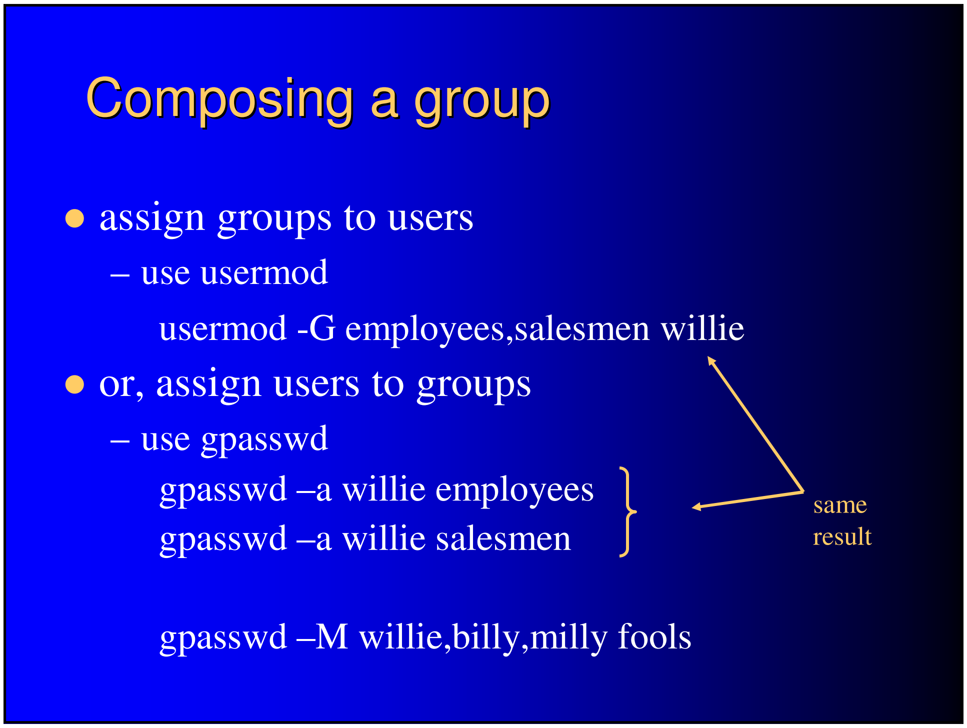
- add a group

  groupadd employees

- delete a group

  groupdel employees

man page caveats: “You must manually check all file systems to insure that no files remain with the named group as the file group ID.... You may not remove the primary group of any existing user. You must remove the user before you remove the group."



Composing a group

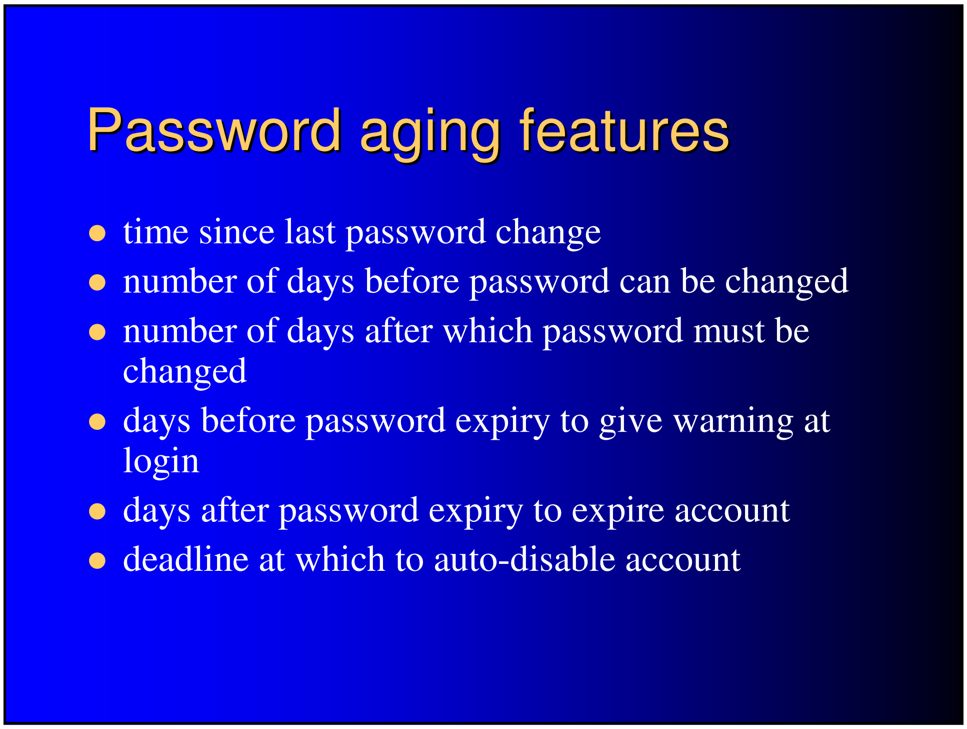
- assign groups to users
  - use usermod

```
usermod -G employees,salesmen willie
```

- or, assign users to groups
  - use gpasswd

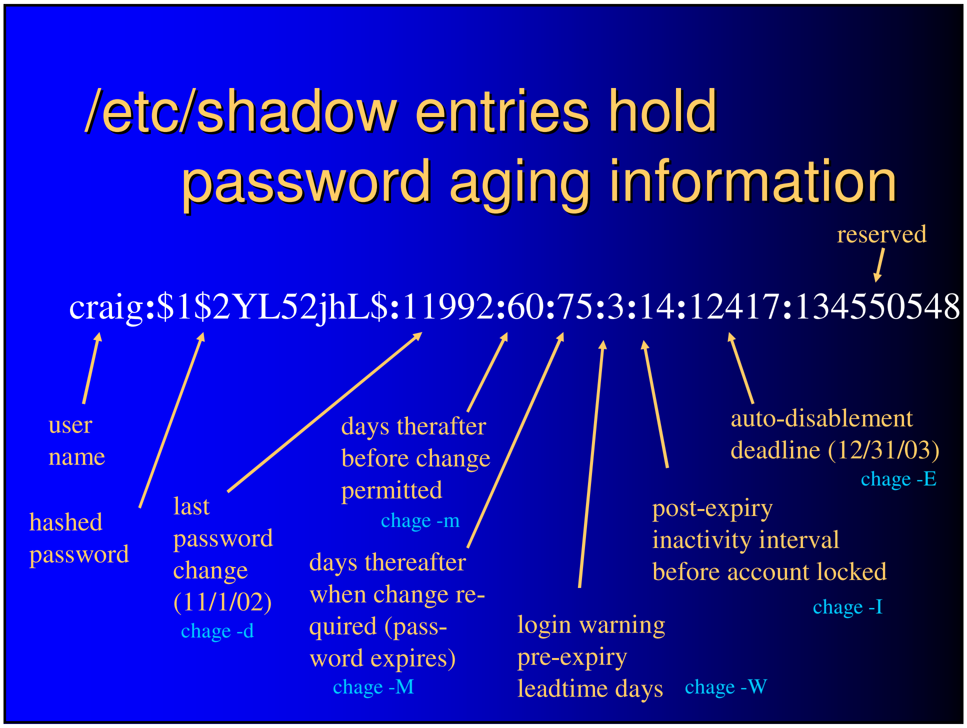
```
gpasswd –a willie employees
gpasswd –a willie salesmen

gpasswd –M willie,billy,milly fools
```

same
result



Password aging features

- time since last password change
- number of days before password can be changed
- number of days after which password must be changed
- days before password expiry to give warning at login
- days after password expiry to expire account
- deadline at which to auto-disable account



/etc/shadow entries hold password aging information

```
craig:$1$2YL52jhL$:11992:60:75:3:14:12417:134550548
```

reserved

user name

hashed password

last password change (11/1/02)
chage -d

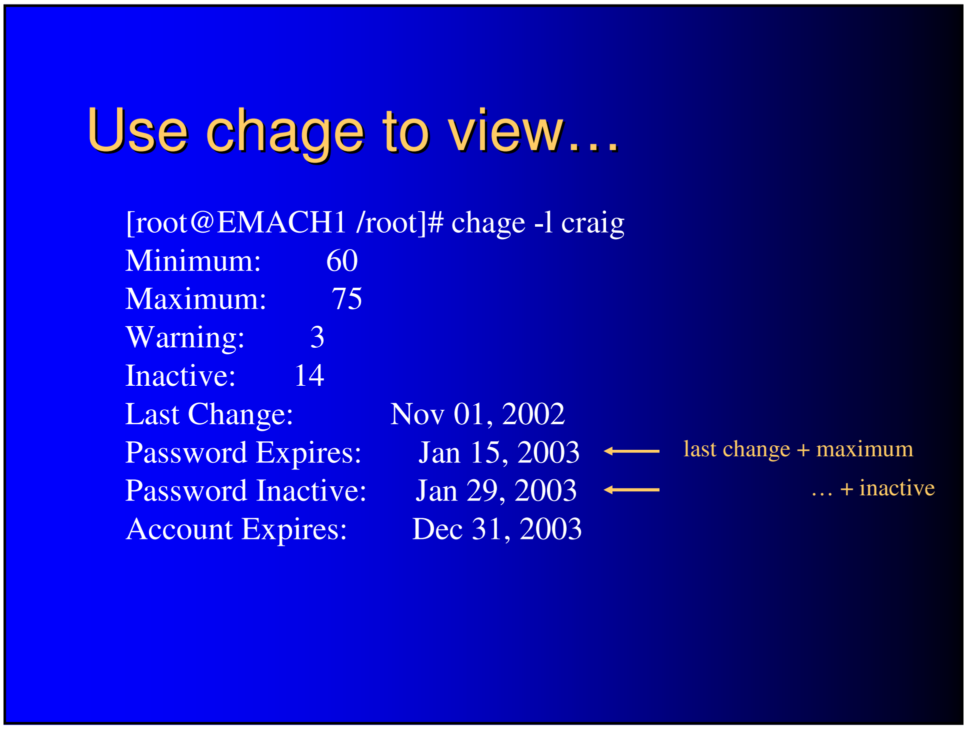
days therafter before change permitted
chage -m

days thereafter when change required (password expires)
chage -M

login warning pre-expiry leadtime days
chage -W

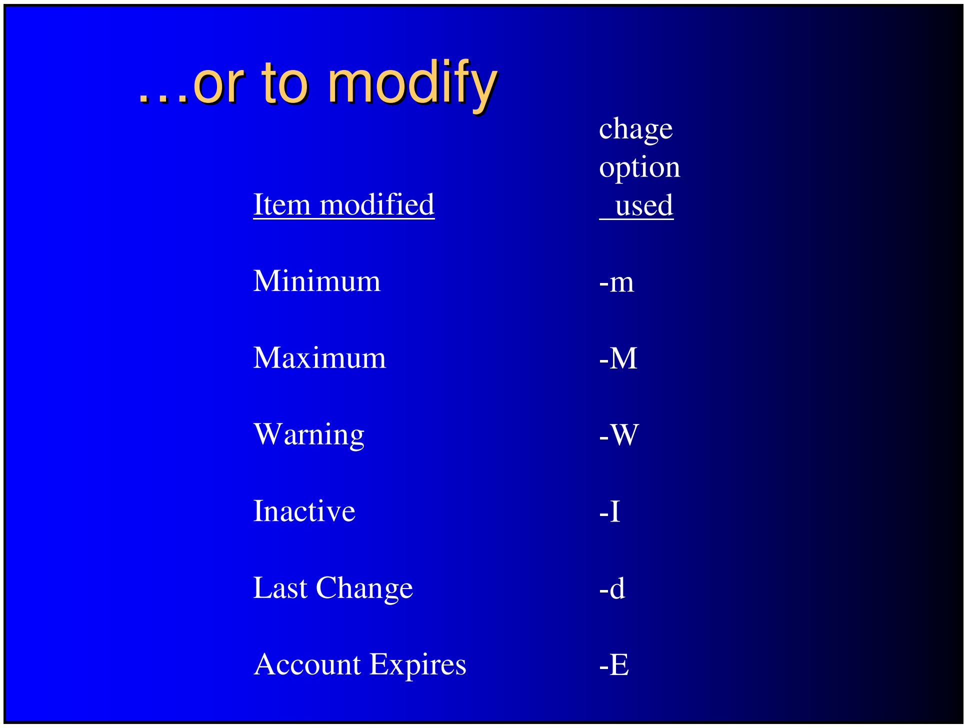
post-expiry inactivity interval before account locked
chage -I

auto-disablement deadline (12/31/03)
chage -E



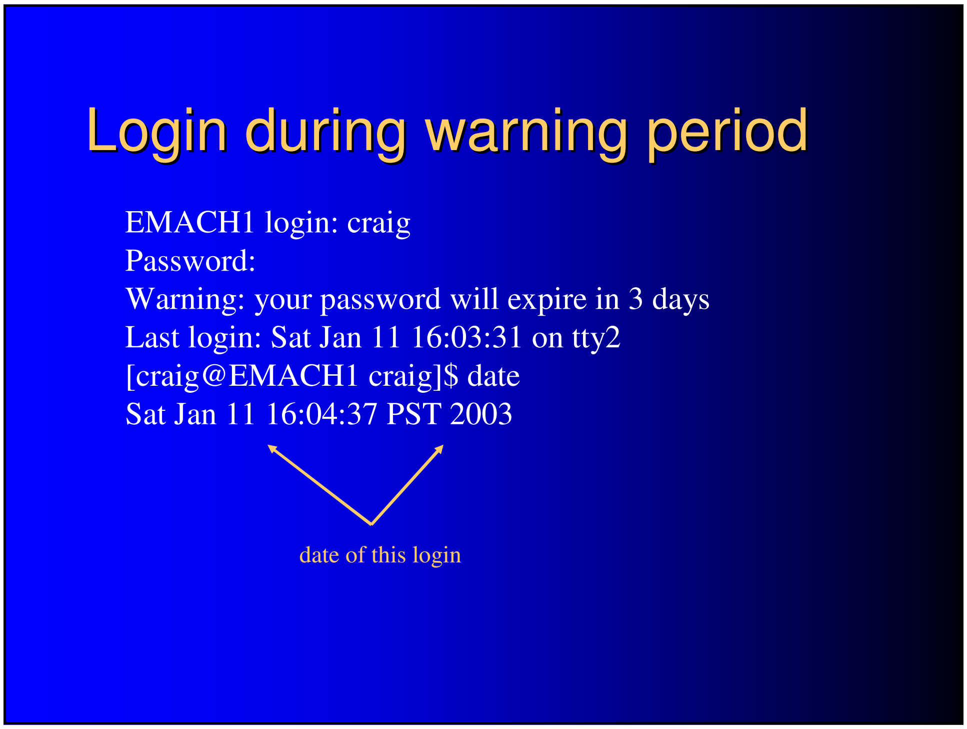
Use chage to view…

```
[root@EMACH1 /root]# chage -l craig
Minimum:        60
Maximum:        75
Warning:        3
Inactive:       14
Last Change:            Nov 01, 2002
Password Expires:       Jan 15, 2003  ←  last change + maximum
Password Inactive:      Jan 29, 2003  ←            … + inactive
Account Expires:        Dec 31, 2003
```



…or to modify

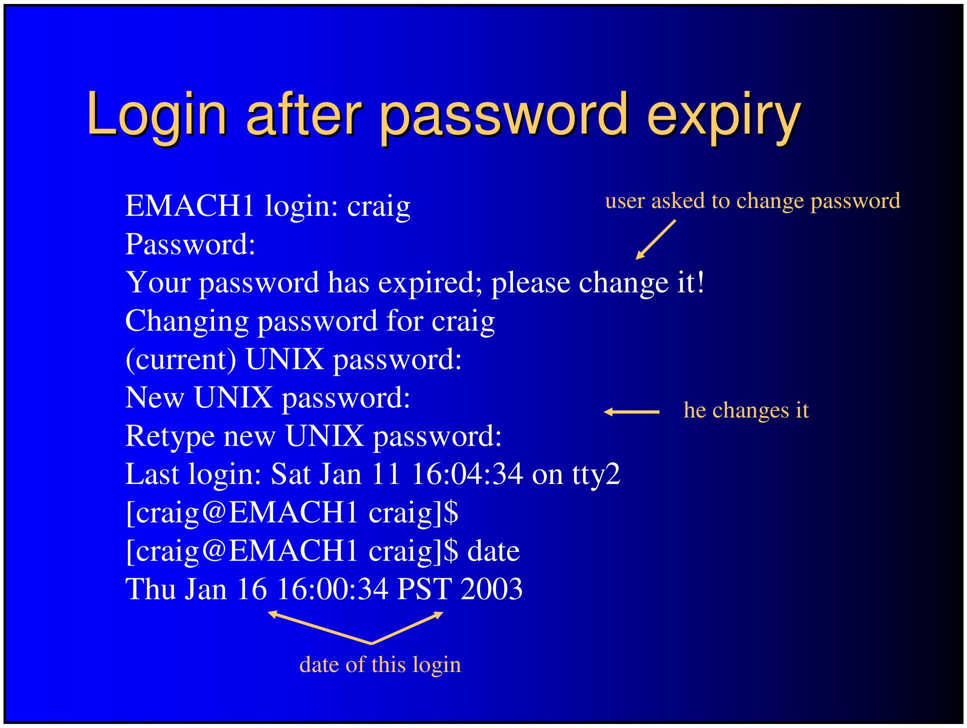
Item modified	chage option used
Minimum	-m
Maximum	-M
Warning	-W
Inactive	-I
Last Change	-d
Account Expires	-E



Login during warning period

```
EMACH1 login: craig
Password:
Warning: your password will expire in 3 days
Last login: Sat Jan 11 16:03:31 on tty2
[craig@EMACH1 craig]$ date
Sat Jan 11 16:04:37 PST 2003
```

date of this login



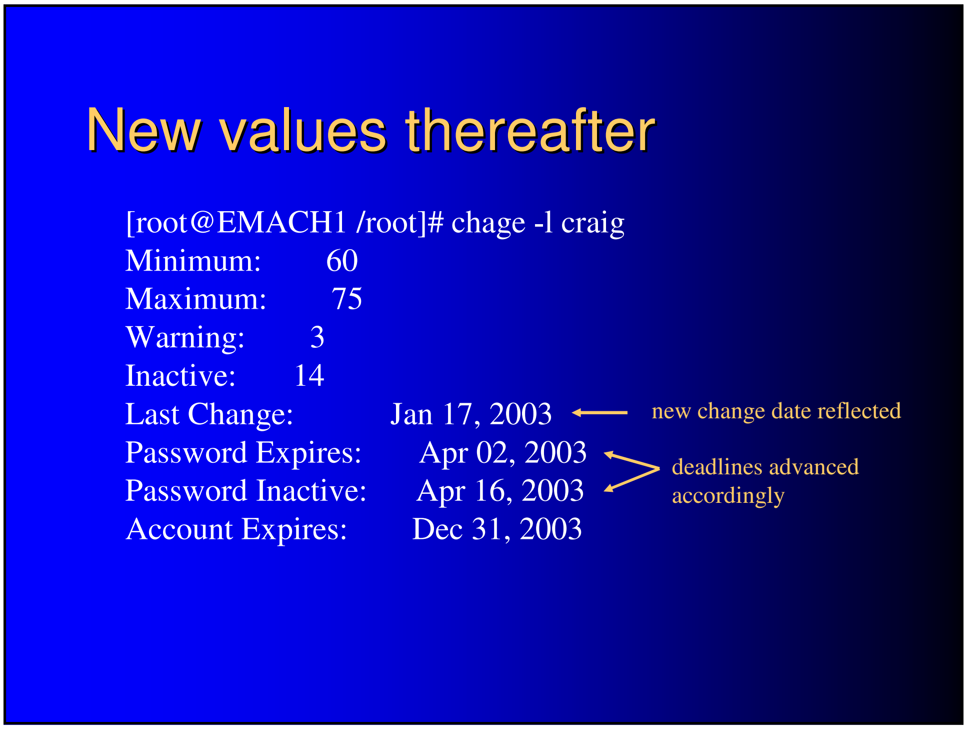
Login after password expiry

```
EMACH1 login: craig
Password:
Your password has expired; please change it!
Changing password for craig
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
Last login: Sat Jan 11 16:04:34 on tty2
[craig@EMACH1 craig]$
[craig@EMACH1 craig]$ date
Thu Jan 16 16:00:34 PST 2003
```

user asked to change password

he changes it

date of this login



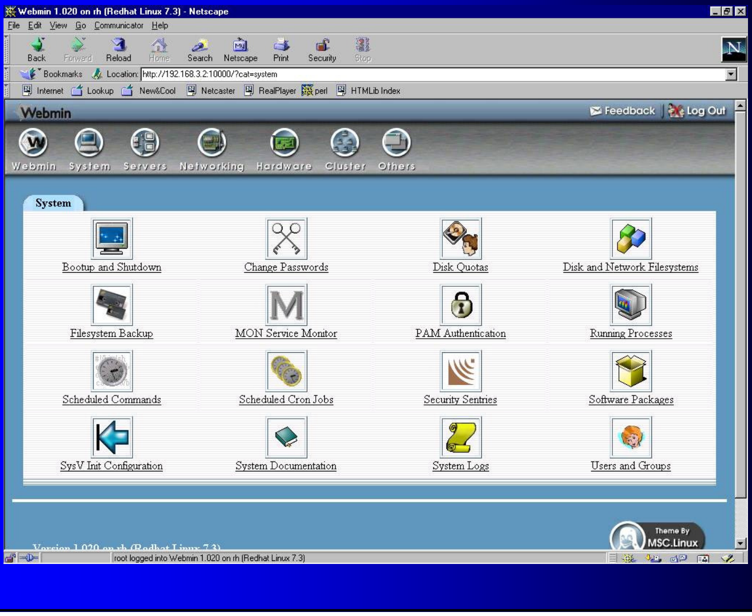
New values thereafter

```
[root@EMACH1 /root]# chage -l craig
Minimum:        60
Maximum:        75
Warning:        3
Inactive:       14
Last Change:            Jan 17, 2003
Password Expires:       Apr 02, 2003
Password Inactive:      Apr 16, 2003
Account Expires:        Dec 31, 2003
```

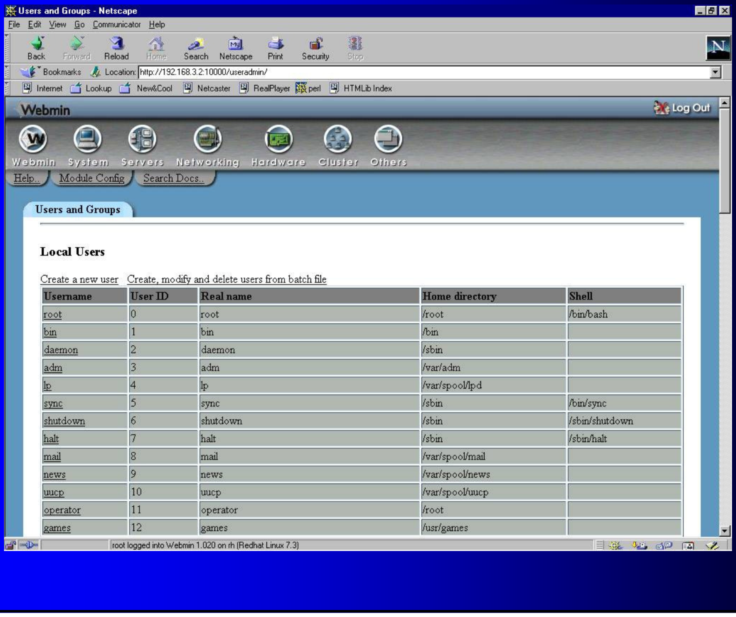
new change date reflected

deadlines advanced
accordingly

Webmin


Webmin 1.020 on rh (Redhat Linux 7.3) - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://192.168.3.2:10000/?cat=system
Internet Lookup New&Cool Netcaster RealPlayer perl HTMLib Index
Webmin
Feedback
Log Out
Webmin System Servers Networking Hardware Cluster Others
System
Bootup and Shutdown
Change Passwords
Disk Quotas
Disk and Network Filesystems
Filesystem Backup
MON Service Monitor
PAM Authentication
Running Processes
Scheduled Commands
Scheduled Cron Jobs
Security Sentries
Software Packages
SysV Init Configuration
System Documentation
System Logs
Users and Groups
Theme By MSC.Linux
Version 1.020 on rh (Redhat Linux 7.3)
root logged into Webmin 1.020 on rh (Redhat Linux 7.3)

Webmin

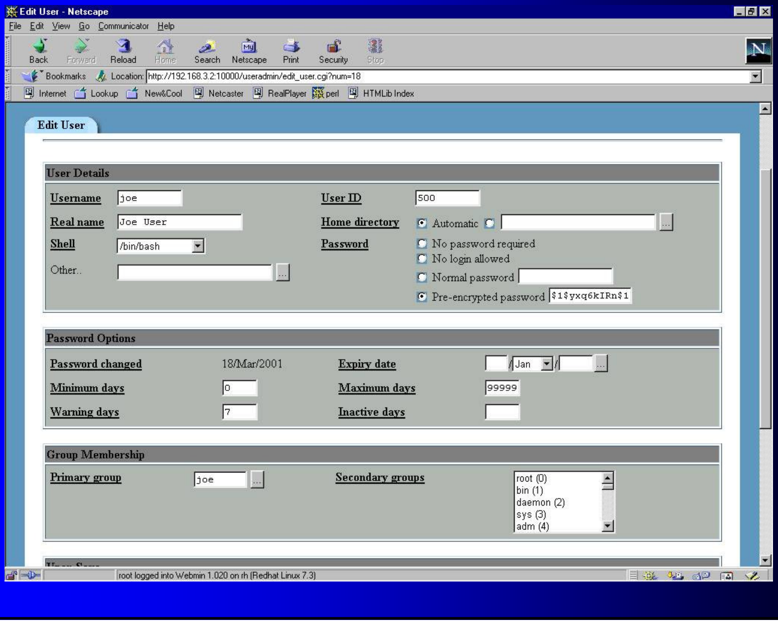

Users and Groups - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://192.168.3.2:10000/useradmin/
Internet Lookup New&Cool Netcaster RealPlayer perl HTMLib Index
Webmin
Log Out
Webmin System Servers Networking Hardware Cluster Others
Help.. Module Config Search Docs..
Users and Groups
root logged into Webmin 1.020 on rh (Redhat Linux 7.3)

Local Users

Create a new user   Create, modify and delete users from batch file

Username	User ID	Real name	Home directory	Shell
root	0	root	/root	/bin/bash
bin	1	bin	/bin	
daemon	2	daemon	/sbin	
adm	3	adm	/var/adm	
lp	4	lp	/var/spool/lpd	
sync	5	sync	/sbin	/bin/sync
shutdown	6	shutdown	/sbin	/sbin/shutdown
halt	7	halt	/sbin	/sbin/halt
mail	8	mail	/var/spool/mail	
news	9	news	/var/spool/news	
uucp	10	uucp	/var/spool/uucp	
operator	11	operator	/root	
games	12	games	/usr/games	

Webmin



Edit User - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks
Internet Lookup New&Cool Netcaster RealPlayer perl HTMLib Index

Location: http://192.168.3.2:10000/useradmin/edit_user.cgi?num=18

Edit User

User Details

Username	joe	User ID	500
Real name	Joe User	Home directory	● Automatic ○
Shell	/bin/bash	Password	○ No password required ○ No login allowed ○ Normal password ● Pre-encrypted password $1$yxq6kIRn$1
Other..			

Password Options

Password changed	18/Mar/2001	Expiry date	/ Jan /
Minimum days	0	Maximum days	99999
Warning days	7	Inactive days	

Group Membership

Primary group	joe	Secondary groups	root (0) bin (1) daemon (2) sys (3) adm (4)

root logged into Webmin 1.020 on rh (Redhat Linux 7.3)