User Administration

Each user has a unique user identification number (UID)

low UIDs (0-99) are for special accounts

UID 0 is root, the superuser/system administrators account, this account always has access to everything.

UID 1–10, system services and system users

UID 11–99, System and special accounts.

Nobody: 99 (numbers may vary), no priviledge account

All operations are by UID.

Human: each user name is assigned a UID.
Linux: UIDs are assigned user names.
`/etc/passwd`: define the users
Name, password, UID, primary group, User Info, home, shell.

Linux: when you need to display a name
(you have the uid),
linearly search the password for the uid
use the first match.

On login: you have a name–
linearly search the password for the name,
use the first match.

Note:

```
joe:xxxxyyyyzzzzt:111:100:OK:/home/joe:/bin/csh
sam:xxxxyyyyzzzzs:111:100:OK:/home/sam:/bin/bash
```

If one user, logs in as sam, gets sam home and shell, but still owns joe home directory.

`joe:x:` means see shadow for password.
`mail:*:` means no login allowed.

Password File Format

```
joe:xxxyyyzzzt:111:100:Joseph,ECS202:/home/joe:/bin/bash
```

Fields separated by :
User name–login name
Encrypted password (or an `x`–see shadow)
UID
Default (primary) group.
GECOS (comment) – user's actual name, and other finger info
Home directory
login shell

Shadow File

Problem: encrypted passwords in a publically readable file.
Attack: guess and test.
Advantage: Only the system checks passwords.
Solution: encrypted passwords in a special (non-readable) file and have the system check it.

`/etc/shadow`

```
sam:OHCp6yRvIfpg2:10296:0:::::
```

Login name, password, password change and account expire information

Groups

Group names are specified in the `/etc/group` file.

Each user belongs to the one group specified in the password file.

The `group` file can specify additional groups for the user

`/etc/group` entry

```
database::123:john,joe
```

Human: the database group is number 123
Computer: group number 123 is called database

In addition to the group specified in the password file, john and joe also belong to the database group.

password field (empty), ignored by many systems.

(some systems): Those in Group 0 group can `su` to root (password will be asked for). Those not in this group, can't su to root, even with password.

sudo: A command used to run a single command as root. You must be in a group (usually `admin`) to use this command. You will be asked your (not root's) password.

Switching Users

You can change to another user (if you have the password).

`su joe` – Start a shell, in this shell you become the user `joe`

You may be prompted for a password.

The password prompt is skipped if:

1) the user has no password (::)

2) you are root when you do the su.

Warning: environment setup is not done by default when switching users.

`su -`

`su -l`

run the full login scripts

User logins

After a user logs in the user environment is (automatically) set up.

1) cd to the users home

2) run the shell initialization scripts. Assuming bash (bourne-again shell):

a) `/etc/profile`
b) `/etc/profile.d/*.sh`
c) `~/.bash_profile` or `~/.profile`

Note: You can run your initialization scripts any time by running:
`source .bash_profile`

3) On logout a file is run:
`~/.bash_logout`

The `.bash_profile` or `.bashrc` may contain aliases (shorthand) for commands.
`alias myls='ls -aCF'`
establishes the command `myls` as an alias for a version of `ls`.

Mail Aliases

`/etc/aliases`

Entries like

```
joe  joe37@netcom.com
```

reroute mail.
Mail sent to "joe" is rerouted.

Entries like

```
admins sam sue bob
```

are mailing lists.
Mail to "techs" is sent to the 3 techs.

Paths

Executables are scattered over several directories, How do you find them?
The search path is a list of directories to be searched, in order, for an executable (colon delimited).

Sample search path:
`export PATH='/bin:/usr/bin:/sbin:. '`

First check `/bin` for the executable,
if you find it, run it;
Next check `/usr/bin`, then check `/sbin`
finally, check the directory you are in (.)

Note: if there is an `ls` in both `/bin` and `/sbin`, you will use the one in `/bin` because that is first in your path.

Warning: putting . into the administrator's path can be dangerous.
Suppose the user `joe` creates an executable called `ls` that formats the hard drive. Joe does not have the priviledge to format the hard drive so he cannot use the program; but if the administrator has . at the front of his path, `cd`'s to joe's home directory and does an `ls` he will run the format program. Joe will have tricked him into destroying the system.

Adding a user

1) Create the password entry.

2) Create shadow password entry.
(Leave password undefined `:*:`)

3) Set a password (`passwd` command)

4) Create the users home directory (with contents)

4a)) `mkdir ~joe` creates a home directory for user `joe`
4b) `cp -r <skeleton-dir> ~joe` gives joe a copy of the skeleton directory

5) Set home directory permissions.
`chmod 755 ~joe`
Allows others to read unprotected stuff of joe, but only joe can write or create files.

6) Change ownership, groups of home and content to the user.
`chown -R joe.joe-group ~joe`
Recursively changes the ownership and group of the home directory and contents.
`joe-group` correspond to the group you listed for joe in the password file.

Deleting Users

1) remove password/shadow entries

2) remove home directory

3) remove any cron/at entries

4) remove/change/add mail aliases

5) remove unread mail

Notes:

3) `cron` and allows users to start jobs when they want, e.g., start my long build job at midnight.

4) aliases may be used as a means to forward the mail to a new system, but nothing should be left that points to the defunct account.

Administration

The password and group files are essential for proper running of Unix.

Password file:

Use vipw whenever it exists
(1) it makes sure only one administrator/program at a time is making changes
(2) it performs some minimal consistency checks.

Regularly back-up the `/etc/passwd`
Back up `/etc/group` occasionally.

Double check your changes

Before going network, make sure all real users have passwords especially root.

Permissions of `passwd` and `group`: `-rw-r--r--`
Permissions of `shadow`: `-rw-------`
All should be owned by `root`

Use user IDs between 500 and 60000.
Low ones = system; high ones = no priviledges

Network File System Consideration–file ownership is by UID, user must have same UID on all systems.

Linux Specifics

`/sbin/adduser`: prompts you for user information, creates the account

`/etc/skel`: default skeleton for new accounts. You *must* set up a default appropriate to your site.

`/home`: standard location of home directories
e.g. `/home/joe`, `/home/sue`

System vs. user setup
System: `/etc/profile` `/etc/profile.d/*.sh`
User: `~/.bash_profile` `~/.bashrc`

System: particular to the computer
e.g. defaults effected by screen size.

User: particular to the user
e.g. paths to programs (compilers, spread sheets...)

Disabling a user: add an asterisk at the start of the password field.
Changing shell: ftp is still available and can be used to modify the contents of the account.

User Account Modification

Several programs allow the user to configure their own account or root to configure any account. If you don't want the user to do a command, apply a:

`chmod o-x`

e.g. `chmod o-x /usr/bin/passwd`

`/usr/bin/passwd`–the user can change their password.
Note: in a group of tightly coupled network systems, special handling is necessary because the password must be changed on all machines at the same time. (see `yppasswd` in the networking section of this course)

`/usr/bin/chfn`–the user can change the GECOS (comment) field information, such as phone, name, address.

`/usr/bin/chsh`–the user can select a different login shell such as `bash`, `ksh`, `csh`
Note: selecting a bad shell such as `/bin/rm` can disable the account.
This command restricts the user's choice to those shells listed in `/etc/shells`.