

Chapter 5: User Management

Chapter 5 User Management



Chapter 5 Outline

- In this chapter we will learn
 - ✓ Where user and group account information is stored
 - ✓ How to manage user accounts and groups
 - ✓ How to establish disk quotas for users

User accounts

- User Accounts

— User accounts

— Where are user accounts stored?

— The `/etc/passwd` file

— The `/etc/shadow` file

— Changing passwords

— Password aging

— The `/etc/group` file

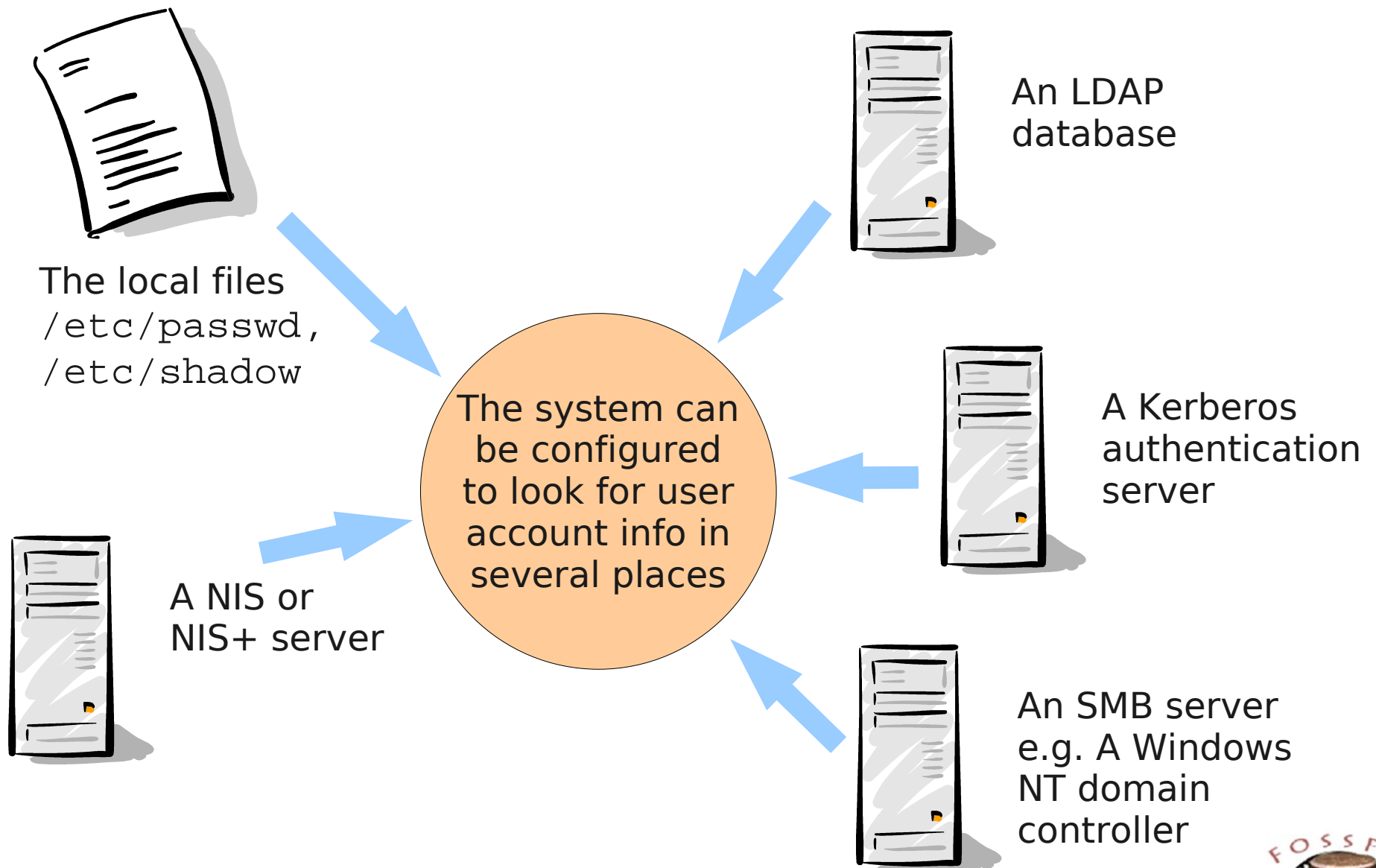
— Group membership

— System accounts and system groups

User accounts

- A user account is defined by:
 - A user name (e.g. tux)
 - A numeric user ID (e.g. 500)
 - A primary group ID (e.g. 100)
 - A full name (e.g. Tux Penguin) which the system does not use internally
 - A home directory (e.g. /home/tux)
 - A shell (e.g. /bin/bash) - this is the program that is started up after the user has authenticated
 - A password. This is stored as a hash (not in clear text)
 - Password aging information, discussed later
- All other information that customises a user's account is held in hidden files in the user's home directory
 - Shell configuration, desktop preferences, environment settings, etc.

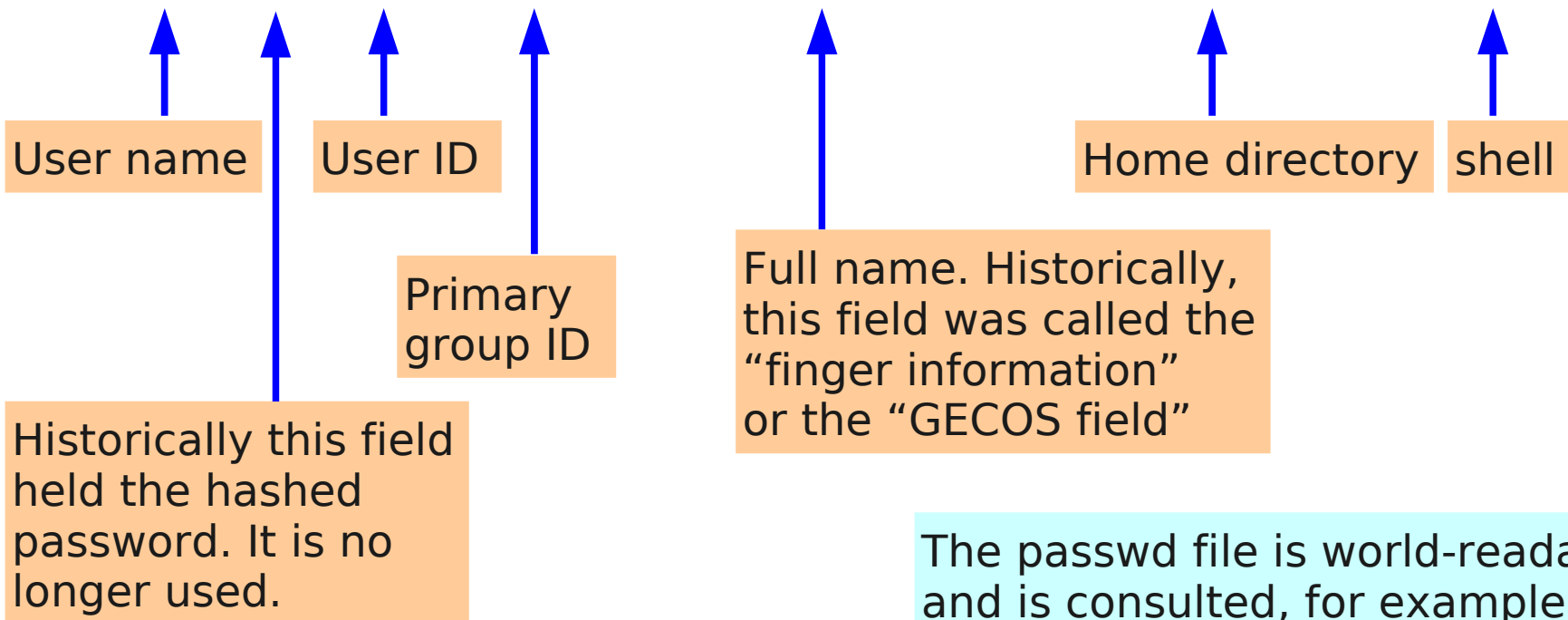
Where are user accounts stored?



The `/etc/passwd` file

- Locally defined accounts are stored in the `/etc/passwd` file
 - The name is historical, passwords are *not* stored in this file!
 - Each line defines one user, fields are separated by colons:

```
tux:x:504:100:Tux Penguin:/home/tux:/bin/bash
```




The `passwd` file is world-readable and is consulted, for example, to allow `ls -l` to show a file's owner as a user name rather than a UID





The /etc/shadow file

- The /etc/shadow file stores hashed passwords and password aging information for each user
 - This file is readable only by root, to prevent users cracking passwords
 - Each line defines one user, fields are separated by colons:

```
tux:RZ1ZW2SM6JdGg:12538:0:99999:7:::
```

• 
User name


The hashed password
(Hashing is effectively
a form of “one way”
encryption)


These fields hold password
aging parameters

When a user logs in, the password he enters is hashed and compared against the hash stored in the shadow file

Changing passwords

- Users are allowed to change their own password
 - A simple password strength check is enforced

```
$ passwd
Changing password for tux.
Old Password: penguin
New password: secret
Bad password: too simple
New password: SuSE2004
Re-enter new password: SuSE2004
Password changed
```

← In reality the passwords are not echoed to the screen

- The superuser can change anyone's password

```
# passwd dilbert
Changing password for dilbert.
...
```

Password aging

- Password aging forces users to choose new passwords periodically
 - Opinions are divided over whether this is a good thing
 - Rarely enabled
- Password aging is controlled by several parameters which are stored in `/etc/shadow`. All these parameters are in days
 - A minimum period between password changes
 - A maximum period between password changes (password lifetime)
 - A warning period prior to password expiration
 - A maximum period of inactivity after password expiration before the account is locked (disabled)
 - An absolute expiration date for the account (days since 1 Jan 1970)
- Root can use the `chage` command to change the password aging parameters for an existing user
 - See `man chage` for details

The /etc/group file

- Groups are defined in the file /etc/group
 - Each line defines one group:

```
video:x:33:chris,dilbert,tux
```

The group name

The numeric group ID

Comma-separated list of users who are secondary members of this group

Some early versions of UNIX associated a password with each group. The hashed password was stored in this field. You could assume the group identity of any group you knew the password for. Linux does not implement this feature.

Group membership

- Each user has one primary group
 - Defined by his entry in `/etc/passwd`
- Each user has zero or more secondary groups
 - Defined by the appearance of his user name in `/etc/group`
- A user is always considered to be a member of his primary group and all his secondary groups

System accounts and system groups

- UIDs below 500 are regarded as “system accounts”
 - Do not usually correspond to an actual person
 - Serve to give an ownership category to system files
 - Example: the print server runs as user 'lp' and the printer config and spool files are owned by 'lp'. lp has UID=4
- Regular users receive UIDs from 500 up
- GIDs below 500 are regarded as “system groups”
 - Regular groups receive UIDs from 1000 up
 - Strangely, the default GID for a regular user account is 100 (“users”)

User account management

- User account management
 - Creating user accounts from the command line
 - Modifying user account defaults
 - Modifying and deleting existing accounts
 - Managing groups from the command line
 - Managing users with YaST
 - YaST user management main screen
 - Adding a user with YaST
 - Setting password aging parameters in YaST
 - Specifying detailed account settings in YaST
 - Specifying account creation defaults in YaST
 - Exercise: User account management
 - Command-line vs graphical tools

Creating user accounts from the command line

- Root can create user accounts with the command `useradd`

```
# useradd kurt
```

- The UID will be one bigger than the largest currently in use
 - Other defaults (location for home directory, shell, etc) are taken from the file `/etc/default/useradd`
- Root should set an initial password for the account (and tell the user!)
- Command options can be used to specify non-default settings, e.g.

Option	Meaning
<code>-d homedir</code>	Set the home directory to homedir
<code>-u 600</code>	Specifies a user ID of 600
<code>-g 120</code>	Specifies a primary group ID of 120
<code>-m</code>	Create the home directory for the account. The home directory is initialised with a copy of the files in <code>/etc/skel</code>
<code>-c Full Name</code>	Specifies the user's full name
<code>-s shell</code>	Set the path name to the user's login shell

Modifying user account defaults

- Root can modify the defaults for new user accounts, for example:

```
# useradd --save-defaults -s /usr/bin/csh
# useradd --show-defaults
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/usr/bin/csh
SKEL=/etc/skel
GROUPS=dialout,uucp,video,audio
```

This example changes the default shell. Also use:

-d homedir
-g GID
-G group1,group2

See the man page for full details

- Subsequently, new users will receive a `csh` by default

Modifying and deleting existing accounts

- Root can modify existing accounts using `usermod`, for example:

```
# grep dilbert /etc/passwd
dilbert:x:501:100:Dilbert:/home/dilbert:/bin/bash
# usermod -c "My Hero" -s /usr/bin/csh dilbert
# grep dilbert /etc/passwd
dilbert:x:501:100:My Hero:/home/dilbert:/usr/bin/csh
```

- The options for `usermod` are similar to those for `useradd`

- Root can delete existing accounts using `userdel`, for example:

```
# userdel -r kurt
```



Forces removal of
kurt's home directory

Managing groups from the command line

- Root can create groups with `groupadd`

```
# groupadd -g 1445 hackers
```



Specifies the GID. By default the next available ID is allocated

- Root can delete groups with `groupdel`

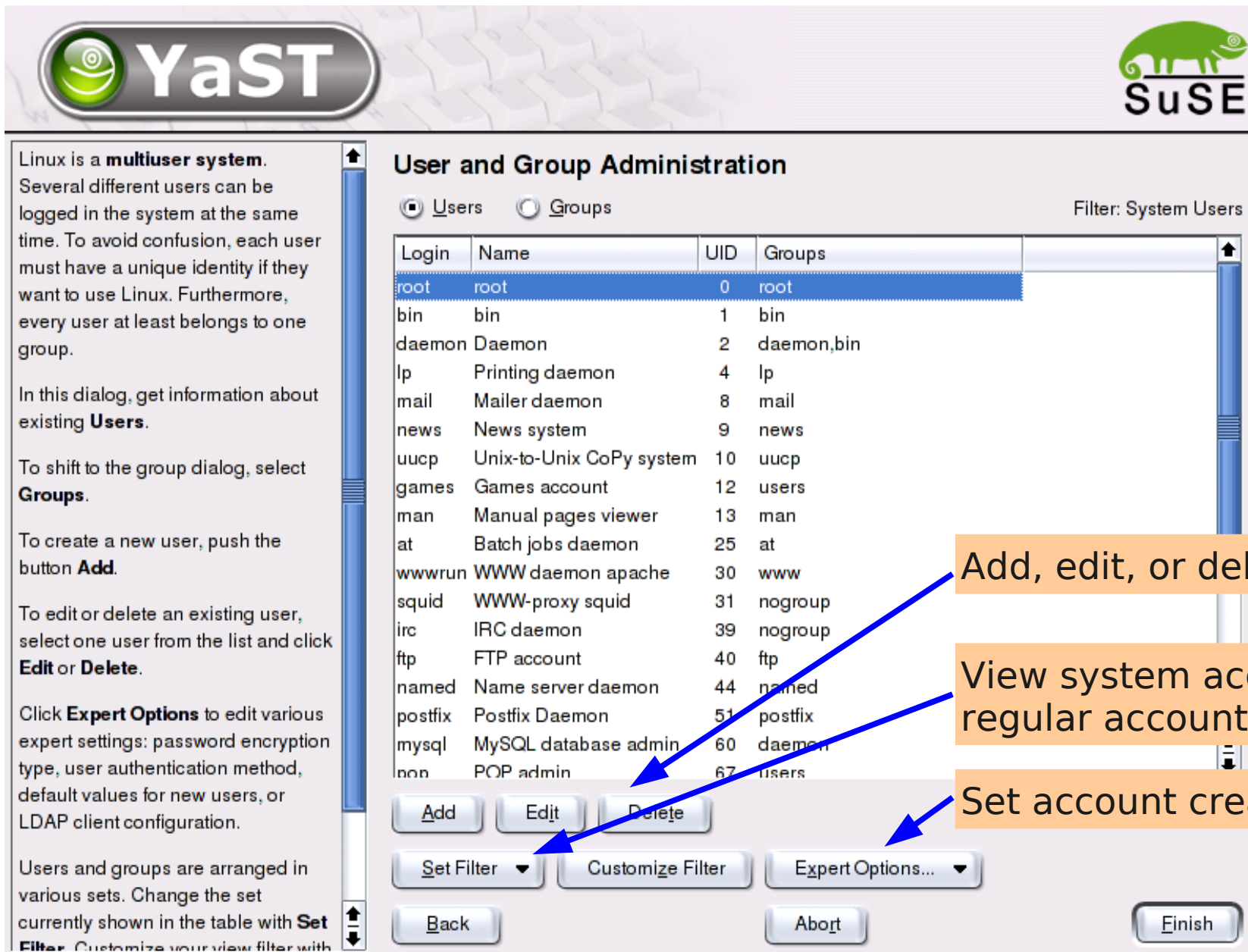
```
# groupdel hackers
```

- You are not allowed to delete a group which is someone's primary group

Managing users with YaST

- YaST provides a fill-in-the-form approach to user management
- On the next slides we'll see:
 - The main user management screen
 - The screen for adding a user
 - The screen for setting password aging parameters
 - The screen for editing user properties (UID, shell, secondary groups, ...)
 - The screen for editing account creation defaults

YaST user management main screen



The image shows the YaST User and Group Administration interface. On the left, there is a sidebar with instructions for managing users and groups. The main area displays a table of system users with columns for Login, Name, UID, and Groups. Below the table are buttons for Add, Edit, Delete, Set Filter, Customize Filter, Expert Options, Back, Abort, and Finish. Three orange callout boxes with blue arrows point to the Add, Edit, and Delete buttons, the Set Filter button, and the Expert Options button.

Linux is a multiuser system. Several different users can be logged in the system at the same time. To avoid confusion, each user must have a unique identity if they want to use Linux. Furthermore, every user at least belongs to one group.

In this dialog, get information about existing **Users**.

To shift to the group dialog, select **Groups**.

To create a new user, push the button **Add**.

To edit or delete an existing user, select one user from the list and click **Edit** or **Delete**.

Click **Expert Options** to edit various expert settings: password encryption type, user authentication method, default values for new users, or LDAP client configuration.

Users and groups are arranged in various sets. Change the set currently shown in the table with **Set Filter**. Customize your view filter with

User and Group Administration

Users Groups Filter: System Users

Login	Name	UID	Groups
root	root	0	root
bin	bin	1	bin
daemon	Daemon	2	daemon,bin
lp	Printing daemon	4	lp
mail	Mailer daemon	8	mail
news	News system	9	news
uucp	Unix-to-Unix CoPy system	10	uucp
games	Games account	12	users
man	Manual pages viewer	13	man
at	Batch jobs daemon	25	at
wwwrun	WWW daemon apache	30	www
squid	WWW-proxy squid	31	nogroup
irc	IRC daemon	39	nogroup
ftp	FTP account	40	ftp
named	Name server daemon	44	named
postfix	Postfix Daemon	51	postfix
mysql	MySQL database admin	60	daemon
pop	POP admin	67	users

Add, edit, or delete accounts

View system accounts or regular accounts

Set account creation defaults

Adding a user with YaST

Add a New System User

User Data

Full User Name

User Login

Password

Verify Password:

Set password aging parameters

Specify detailed account settings

Setting password aging parameters in YaST

Password settings for user isaac:

Date of the last password change: 30/04/04

Days before Password Expiration to Issue Warning

7

Days after Password Expires with Usable Login

3

Maximum number of days for the same password

100

Minimum number of days for the same password

1

Expiration date

2005-12-31

Specifying detailed account settings in YaST

Add/Edit User Properties - Details

Detailed Profile for User "isaac"

User ID (uid)
102

Home Directory
/home/isaac

Additional User Information:

Login shell
/bin/bash

Default group
users

Additional Group Membership

- users
- bin
- privoxy
- sys
- ntadmin
- postfix
- trusted
- shadow
- utmp
- floppy
- uucp
- xok
- named
- wheel
- nogroup
- cdrom
- public
- audio
- daemon
- man

Specifying account creation defaults in YaST

New User Defaults

Set the Default Values for Adding New User

Default Group

users

Secondary Groups

dialout,uucp,video,audio

Default Login Shell

/usr/bin/csh

Path Prefix for Home Directory

/home/

Browse...

Skeleton for Home Directory

/etc/skel

Browse...

Default Expiration Date

Days after Password Expiration Login Is Usable

-1

Exercise: User account management

For this exercise you can use either the command line tools or YaST, as you prefer.

1. Create a new group called `hacker` with group ID 1200
2. Create two new users with user names `catbert` and `dogbert` and UIDs 610 and 611 respectively. Their home directories should be created as `/new1/catbert` and `/new1/dogbert` and should be populated with initial configuration files from `/etc/skel`. Their primary group should be `hacker` and they should both have secondary membership of the groups `audio` and `video`
3. Set passwords for these two accounts. Make sure you remember them!
4. Examine the `/etc/group` and `/etc/passwd` files and verify that the entries for the `hacker` group and the `catbert` and `dogbert` accounts are as you expect

Exercise: User account management (continued)

5. Set password aging on catbert's account so that:
 - The minimum time between password changes is 1 week
 - The maximum time between password changes is 1 month
 - He receives 2 days warning of password expiry
 - The account is usable for 3 days after the password expires
 - The account expires on 31 December this year
 - Hint - look up the man page for `chage`, or use the YaST screens
6. Look up catbert's entry in `/etc/shadow` and examine the password aging parameters. Are they what you expect?
 - Hint - `man 5 shadow` will help you figure out which field is which
7. List the `/new1` directory. Verify that catbert and dogbert's home directories exist.
 - Who are they owned by? What group do they belong to?

Exercise: User account management (continued)

8. Verify that you can log in as catbert

- Run the command `id` to verify catbert's UID and group memberships. Are they what you expect?
- Run the command `pwd` and verify the path name of the home directory
- Create a file in the home directory and verify its ownership and group
- List the hidden files in the directory. Where have these come from?

9. Still as catbert, try to change your password. What happens?

End of Exercise

Command-line vs graphical tools

- The relative merits of performing system administration at the command line and using graphical tools provokes a lot of discussion
- Advantages of graphical tools
 - No need to remember command line syntax or file formats
 - Less likely to create an invalid configuration
 - Closer match to expectations of administrators migrating from Windows
- The use of graphical tools does *not* mean that you don't need to understand the underlying model
- Advantages of command line tools
 - Generally faster, once you know them
 - Do not require a graphical desktop
 - System admin can be automated, using scripts
 - May offer greater flexibility than the graphical tools

Disk quotas

- Disk quotas
 - About disk quotas
 - Initialising disk quotas
 - Enabling disk quotas
 - Setting and modifying quota limits
 - Quota grace period
 - Reviewing quota utilisation
 - Exercise: Establishing disk quotas

About disk quotas

- Linux supports a quota mechanism to limit disk usage
 - Soft limits (can be exceeded for a specified grace period)
 - Hard limits (cannot be exceeded)
 - Limit the number of disk blocks
 - Limit the number of inodes
 - Limits can be set per user and per group
- Quotas are set and checked on a per-partition basis
 - Typically, quotas are used on the `/home` partition

Initialising disk quotas

- To use disk quotas on a partition, it must be mounted with the `usrquota` and/or `grpquota` option

- Usually specified in the option field of `/etc/fstab`, for example:

```
/dev/hda4 /home /ext2 defaults,usrquota 1 2
```

- To change the mount options without rebooting, just remount:

```
# mount /home -o remount
```

- Quota information is held in the files `aquota.user` and/or `aquota.group` in the top level directory of the partition.

Use `quotacheck` to create these files:

```
# quotacheck -uv /home
```



```
-u Create user quota file  
-g Create group quota file  
-v Verbose mode
```

This only needs to be done once



Enabling disk quotas

- The `quotaon` command enables the quota mechanism

```
# quotaon -uv /home
```



-u Enable user quotas
-g Enable group quotas
-v Verbose mode

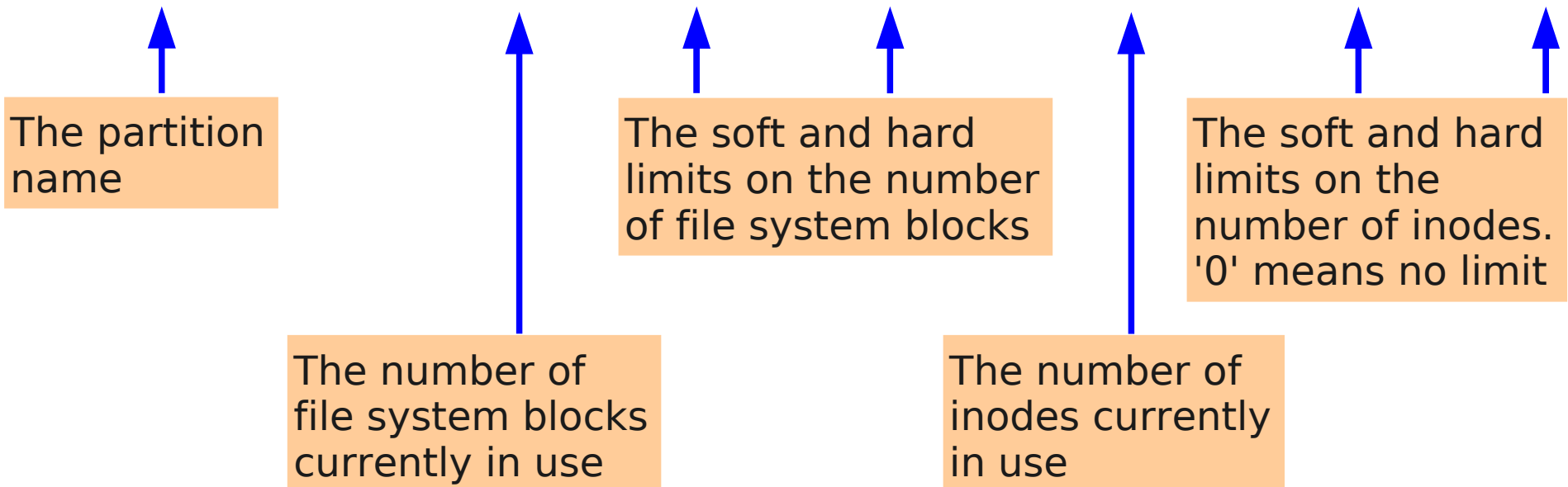
- This needs to be done each time the system is booted
 - Typically the command `quotaon -a` is executed from a boot-time script
 - The `-a` option enables quotas on all file systems listed in `fstab` with quotas
- Quotas can be disabled with `quotaoff`

```
# quotaoff -uv /home
```

Setting and modifying quota limits

- The command `edquota` is used to set quotas for each account
 - Invokes a text editor (default: `vi`) to edit the settings:

```
# edquota tux
Disk quotas for user tux (uid 500):
Filesystem  blocks    soft    hard    inodes    soft    hard
/dev/hda4   414      4000   5000    125       0       0
```



Quota grace period

- A user is allowed to exceed his soft limit for a specified grace period
 - If the soft limit is still exceeded at the end of the grace period, the user will be prevented from creating any more files
- The grace period can be edited with the command `edquota -u -t`
 - Invokes a text editor (default: `vi`) to edit the settings
 - Separate grace periods can be specified for the blocks limit and the inode limit, but you cannot specify a different grace period for each user
 - Can be specified in seconds, minutes, hours, or days
 - The default value is 7 days

```
# edquota -u -t
```

```
Grace period before enforcing soft limits for users:
```

```
Time units may be: days, hours, minutes, or seconds
```

Filesystem	Block grace period	Inode grace period
/dev/hda5	3days	3days

Reviewing quota utilisation

- The `quota` command summarises quota usage by user
 - Normal users can display only their own quotas
 - Root can display quotas for any specified account

```
# quota tux
Disk quotas for user tux (uid 500):
Filesystem blocks quota limit grace files quota limit grace
/dev/hda4 4624* 4000 5000 3days 95 0 0
```

- The `repquota` command summarises quota usage by partition

```
# repquota /new1
*** Report for user quotas on device /dev/hda5
Block grace time: 3days; Inode grace time: 3days
```

User		used	Block limits			File limits			
			soft	hard	grace	used	soft	hard	grace
root	--	8239	0	0		5	0	0	
catbert	--	124	4000	5000		91	0	0	
dogbert	--	51	0	0		24	0	0	

Exercise: Establishing disk quotas

In this exercise we will establish a disk space quota for catbert so that he may not use more than 5000 blocks of space in his home directory

- Use the examples in the notes as a guide
 - Remember: catbert's home directory is on the `/new1` partition
1. Modify the line in `/etc/fstab` for the `/new1` mount point, adding "usrquota" in the options field.
 2. Use the `mount` command with appropriate arguments to force the `/new1` partition to be remounted
 3. Examine the mount table and verify that the partition is now mounted with the `usrquota` option set:

```
# mount | grep new1
```

Exercise: Establishing disk quotas (continued)

4. Use the `quotacheck` command with appropriate arguments to create the user quota file for the `/new1` partition
5. List the `/new1` directory and verify that the file `aquota.user` exists
6. Use the `quotaon` command with appropriate arguments to enable the quota system
7. Use the `edquota` command with appropriate arguments to establish quota limits for `catbert` as follows:

For the block count, set a soft limit of 4000 blocks and a hard limit of 5000 blocks. Do not set a limit for the inodes (i.e. leave the soft and hard limits at 0).

Save the file and quit the editor.

Exercise: Establishing disk quotas (continued)

Next, we will check that quotas are working

8. Log in as catbert
9. Run the `quota` command with no arguments. This will show you catbert's current quotas and usage. How many blocks is catbert currently using?
10. Copy the file `/boot/vmlinuz` into catbert's home directory, calling the copy `data1` (this is just a convenient, reasonably large file -- it's about a megabyte).
11. Re-run the `quota` command. How many disk blocks are now in use?
12. Calculate how many copies of the `vmlinuz` file catbert can store in his home directory before his soft limit is exceeded. (On our test system the answer was 3).
13. Make more copies of `vmlinuz` calling them `data2`, `data3`, etc. When the fourth copy is made, you should receive a warning that your user block quota has been exceeded.

Exercise: Establishing disk quotas (continued)

14. Re-run the `quota` command. Note that the block count now has a '*' after it, (indicating that the quota has been exceeded) and that the grace period is 7 days.

(Unless catbert deletes some files, the grace period will reduce each day. On reaching zero, catbert will not be allowed to create any more files)
15. Try to make one more copy of the `vmlinux` file. This time the operation should fail as the hard limit is exceeded.
16. Perform a long listing of the directory. You should see that the last copy of the file is truncated
17. Run the `quota` command one more time. How many disk blocks is catbert using now?
18. Finally, remove the file copies you made.

Congratulations! You have enabled and verified disk quotas.

Quiz

- Which files would you need to consult to discover:
 - The name of a user's primary group?
 - The names of a user's secondary groups?
- Where are a user's preferences (desktop settings, etc.) stored?
- What command would you use to delete a user account?
- Explain the difference between a soft limit and a hard limit for quotas
- Why is `/etc/passwd` world-readable?
- True or false?
 - The grace period for disk quotas can be set individually for each user
 - The inode limit for disk quotas can be set individually for each user
 - Password aging parameters are stored in `/etc/shadow`
 - Only the superuser can change passwords