Chapter 11
Software installation and backup

ict@innovation

FOSSFA

Last revised: 20/6/2004

Chapter 11 Outline

- In this chapter we will learn about:

  - ✓ Installing software from binary RPM packages

  - ✓ Installing software from source

  - ✓ Archiving software with `tar`

  - ✓ Backing up and restoring the file system with `dump` and `restore`
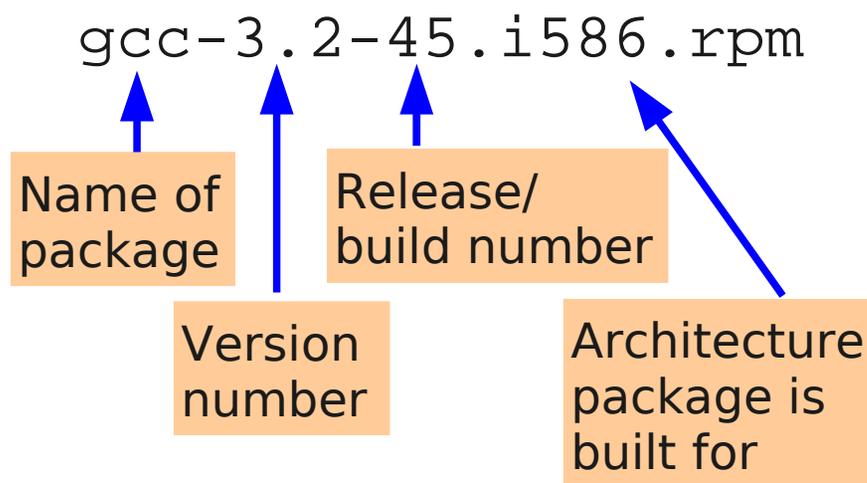
ict@innovation

Installing packages

- Installing packages
  - RedHat package manager (RPM)
  - RPM packages
  - Querying packages
  - Installing packages
  - Dependencies
  - Upgrading and uninstalling packages
  - Installing packages using YaST
  - Package selection using YaST
  - Installing packages using Konqueror
  - Downloading binary RPMs from the Internet
  - Installing from source – the pros and cons
  - Installing from source – the procedure

ict@innovation

FOSSFA

RedHat package manager (RPM)

- The RedHat package manager (RPM) partly automates the installation and removal of software packages

  - Widely used -- not just on RedHat linux -- used by SuSE, too

  - Primarily used to install binary packages which are pre-built for a specific linux distribution and version

- The SuSE linux installation CDs consist mainly of RPM packages

- RPM maintains a database of packages currently installed
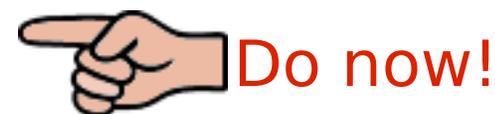
ict@innovation

RPM packages

- An RPM package contains several elements
  - The files that make up the package (executables, configuration files, documentation, etc)
  - Installation scripts
  - Package description
  - Dependency information
- RPM packages have file names of the form:

```
gcc-3.2-45.i586.rpm
```

Name of package

Version number

Release/ build number

Architecture package is built for

ict@innovation

Querying packages

- The `rpm` command lets you query the currently-installed packages

```
$ rpm -qa
```
⟵ List all currently installed packages

```
$ rpm -qf /bin/bash
```
⟵ Which package does this file come from?

```
$ rpm -qi logsurfer
```
⟵ Print general description of package

```
$ rpm -ql logsurfer
```
⟵ List files contained in package

- With the `-p` flag, `rpm` can query uninstalled packages (rpm files)

```
$ rpm -qip gcc-3.2-45.i586.rpm
```
⟵ Print general description of an uninstalled package

```
$ rpm -qlp gcc-3.2-45.i586.rpm
```
⟵ List the files contained in an uninstalled package

ict@innovation

Installing packages

- Packages are installed using:

  ```
  # rpm -ivh package_name.rpm
  ```

  - `-i` flag means install, `-v`, `-h` flags just give more verbose output

- Example:

  ```
  # rpm -qa | grep yacc              ← Package is not currently installed
  # cd /media/cdrom/suse/i586/
  # rpm -ivh yacc-91.7.30-764.i586.rpm
  Preparing...
              ################################# [100%]
     1:yacc
              ################################# [100%]
  # rpm -qa | grep yacc
  yacc-91.7.30-764                   ← Package is now installed
  ```

ict@innovation

Dependencies

- Software is often dependent on other software
  - Programs are dependent on the libraries against which they are linked (linux uses dynamic linking to shared libraries)
  - Graphical tools may be dependent on underlying command-line tools
  - Program will not work if the supporting software is not available
- One of RPM's benefits is its dependency checking
  - Will not allow installation of a package which has missing dependencies
- Example

```
# rpm -i totem-0.99.2-77.i586.rpm
error: Failed dependencies:
    libgnome-desktop-2.so.2 is needed by totem-0.99.2-77
```

ict@innovation

FOSSFA

Upgrading and uninstalling packages

- Packages can be upgraded by using `rpm -U` instead of `rpm -i`

  - The older version of the package is automatically removed

  - If no older version is installed, `rpm -U` behaves just like `rpm -i`

- Packages can be uninstalled using `rpm -e package_name`:

```
# rpm -qa | grep yacc
yacc-91.7.30-764
# rpm -e yacc
# rpm -qa | grep yacc
```

Note that you need only specify the base name of the package

- RPM will not allow removal of a package on which other currently installed packages depend:

```
# rpm -e  m4
error: Failed dependencies:
        m4 is needed by (installed) autoconf-2.57-137
        m4 is needed by (installed) bison-1.75-109
```

ict@innovation

Exercise: Installing a package with RPM

- In this exercise we will use RPM to install the GNU C compiler from the United Linux CDs. You must be logged in as root for this. Then we will write and compile a tiny program, to verify that it works.

1. Verify that `gcc` is not installed by trying to run it (with no arguments). You should receive a message to say that the command is not found

2. Insert United Linux CD2, and mount it on `/media/cdrom`

3. `cd` to the directory `/media/cdrom/UnitedLinux/i586` and list the contents. What files do you see?

4. Run the query commands (`rpm -qip` and `rpm -qlp`) on one of the packages

5. The `gcc` compiler is in the package `gcc-3.2-45.i586.rpm`. Try to install this package, using the example in the notes. What happens?

6. Install the missing supporting library which was reported by rpm. (It is in the same directory), then repeat the installation of the `gcc` compiler. It should work this time

ict@innovation

7. Return to your tux account. `cd` to your home directory and create a file called `greet.c` with the contents:

```
main()
{
    printf("hello world\n");
}
```
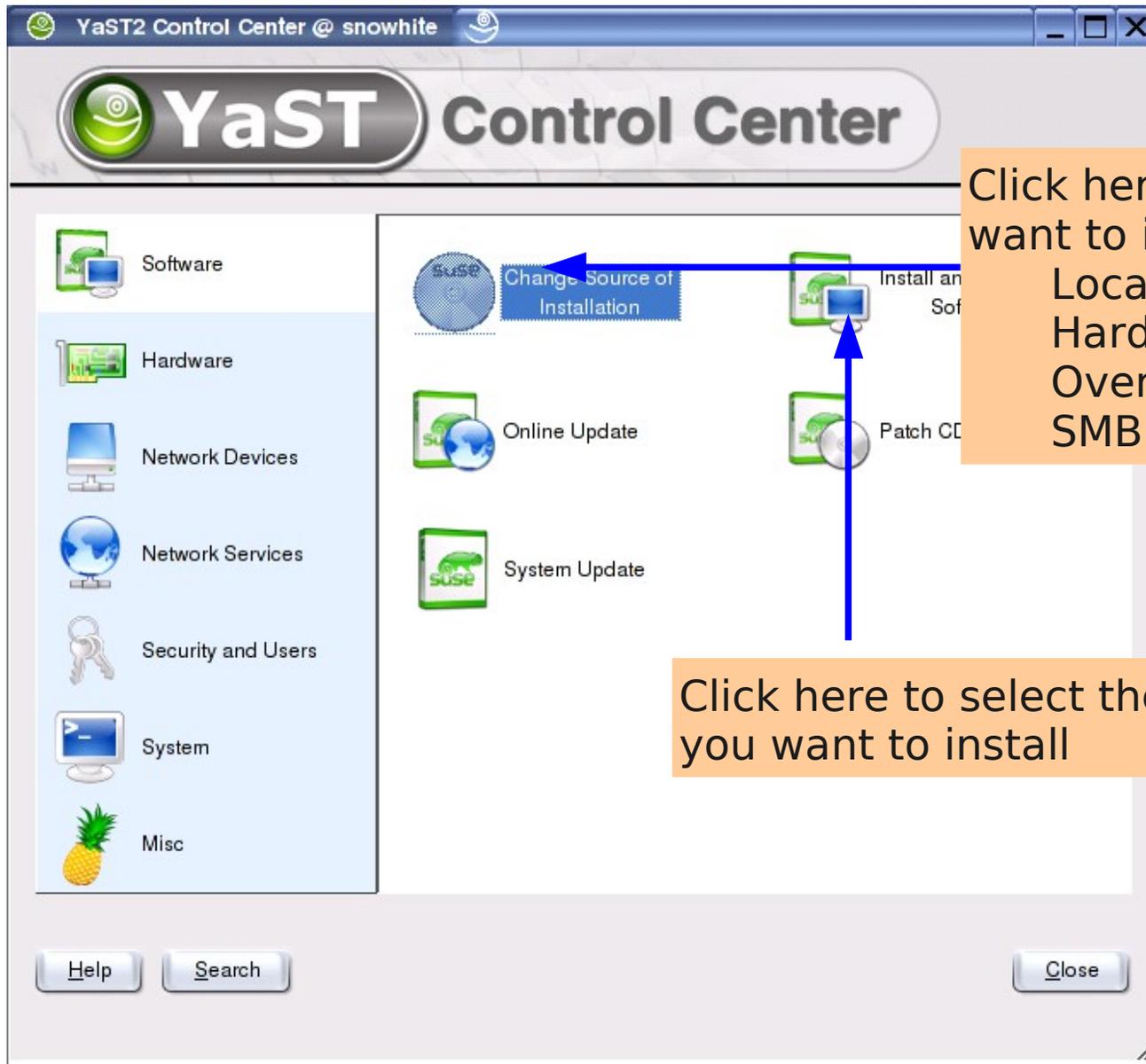
8. Compile the file with the command:

   `$ gcc greet.c -o greet`

9. Run the program with the command:

   `$ ./greet`

- Does it work? If so, congratulations! You have successfully installed the C compiler from an RPM package

ict@innovation

FOSSFA

Installing packages using YaST

YaST2 Control Center @ snowhite

YaST Control Center

- Software
- Hardware
- Network Devices
- Network Services
- Security and Users
- System
- Misc

- Change Source of Installation
- Install and Remove Software
- Online Update
- Patch CD
- System Update

Help   Search   Close

Click here to select the place(s) you want to install software from. Can be:
Local CD or DVD
Hard drive
Over network using FTP, HTTP, SMB or NFS

Click here to select the packages you want to install

ict@innovation

FOSSFA

Package selection using YaST
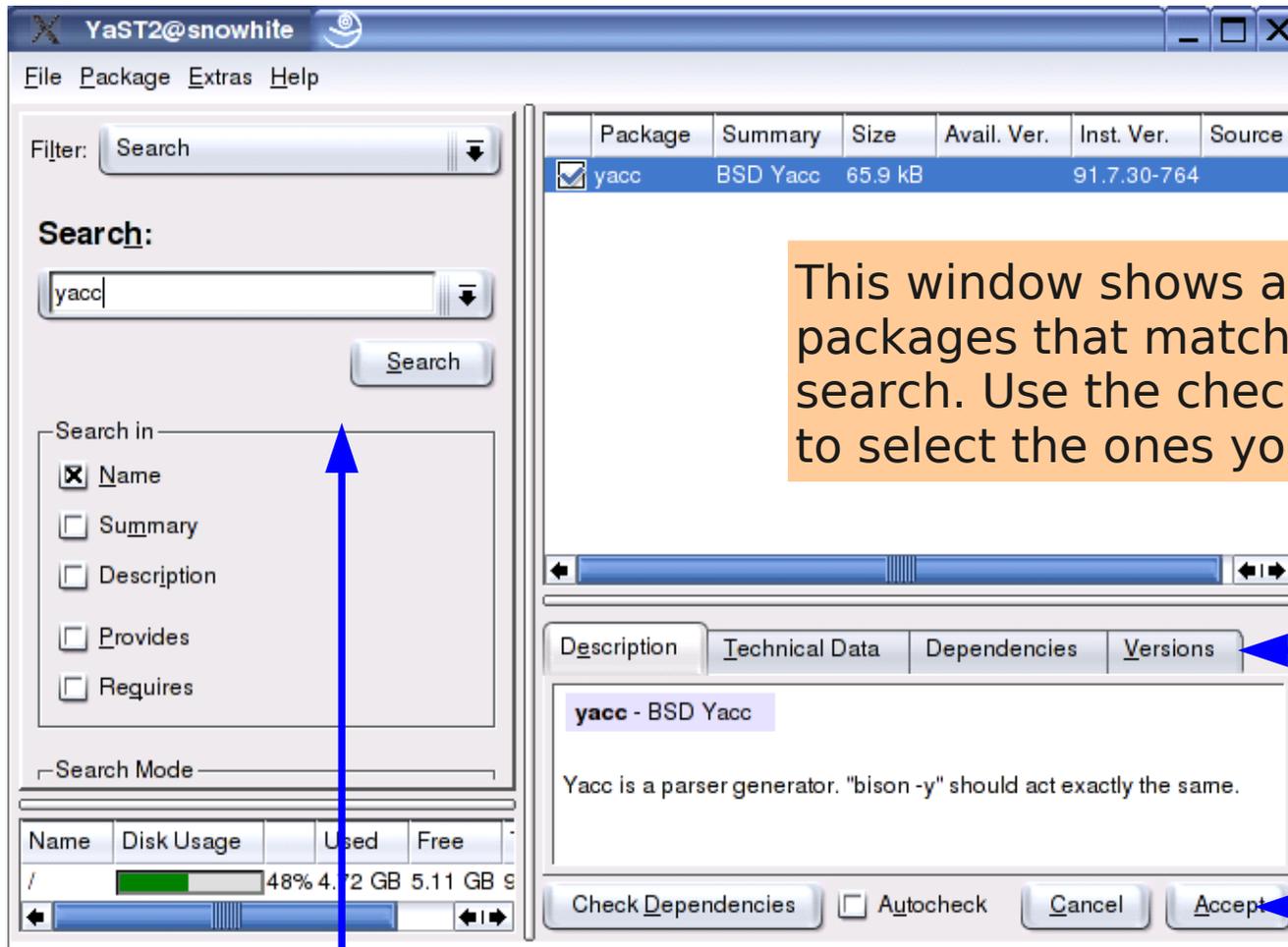
This window shows all available packages that match your search. Use the check boxes to select the ones you want
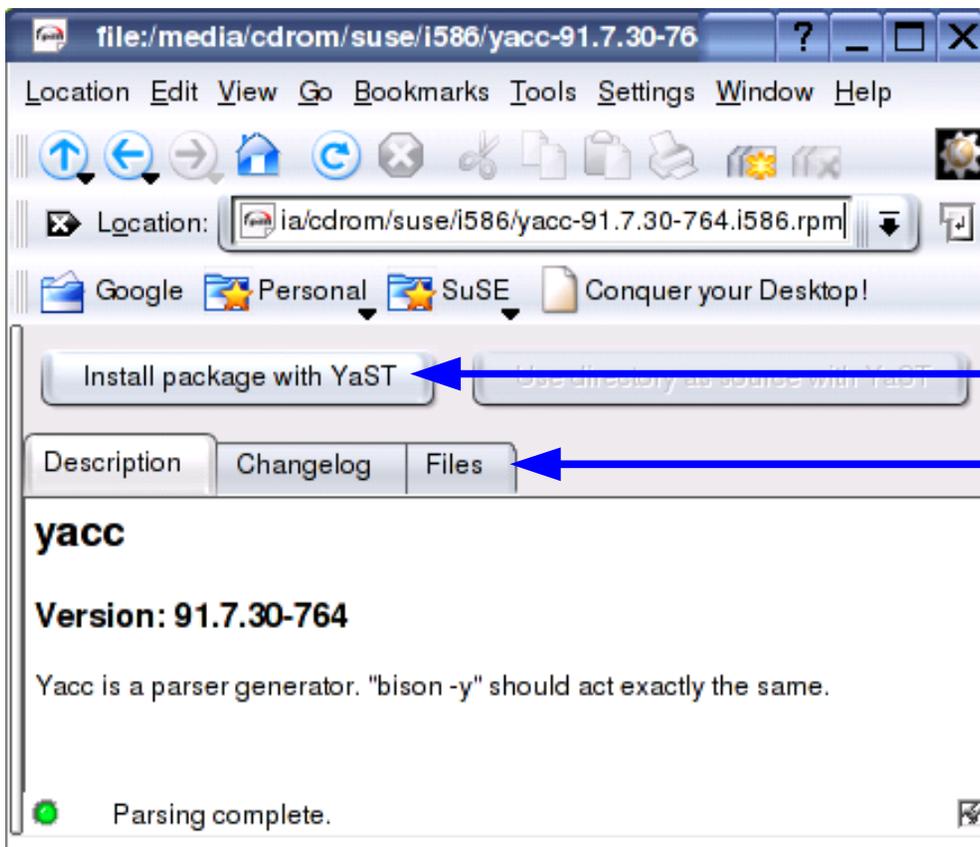
These tabs display info about the selected package

Click here to install the selected packages

This pane lets you locate packages
   - by name
   - by functional groups

Installing packages using Konqueror

- Selecting an RPM file in Konqueror will display package information and give you the option to install it

Click here to install the package using YaST

These tabs display package information (Similar to using `rpm -qip, rpm -qlp`)

ict@innovation

Downloading binary RPMs from the Internet

- There is a lot of free software available for linux which is not provided on the SuSE distribution

    - `www.rpmfind.net` is a good place to look

- If you download binary RPMs, be sure to get the right version

    - It needs to be compiled for the right architecture (i586, sparc, alpha, etc);

        - This is usually apparent from the file name

    - In many cases it needs to be configured for the correct distribution. For example, RedHat and SuSE put many configuration files in different places; packages built for one distribution will often not work on another

        - This is not always apparent from the file name

ict@innovation

Installing from source: the pros and cons

- If you have `gcc` and related development tools installed, it's easy to download and build from source code

- Advantages

  - You can install software for which no binary RPM is available

  - You can get the latest version (the SuSE RPMs are likely not the latest)

  - The same source distribution can be built on many platforms

  - You have greater choice in configuring the software

    - Where the pieces will be placed in the file system
    - Which features are compiled in and which are excluded

- Disadvantages

  - You need the compiler, etc. installed

  - Your system may become more difficult to maintain and upgrade, especially if support is provided by a third party

ict@innovation

FOSSFA

Installing from source: the procedure

- Source distributions are most commonly supplied as "tarballs"

    - Compressed archives built using the `tar` utility

- Here is the command sequence for building the `net-snmp` package from source; it is very typical

- First, download the tarball `net-snmp-5.1.tar.gz` from a suitable internet archive such as `sourceforge.net`, into `/tmp` (say)

```
$ cd ~/build
$ tar zxvf /tmp/net-snmp-5.1.tar.gz
$ cd netsnmp-5.1
$ ./configure
$ make
$ su -c 'make install'
```

Extract the archive into some convenient directory

This will probe your system and create a 'makefile' suitable for building the software on your machine

Build the software

Install the software (usually needs to be done as root)

Archiving and backup tools

- Archiving and
  backup tools

  - Creating archives with `tar`

  - `tar` examples

  - `dump` and `restore`

  - Creating a dump

  - Dump regime

  - Doing a full restore

  - Using restore interactively

  - Third-party backup solutions

ict@innovation

Creating archives with `tar`

- `tar` (originally, "tape archiver") is a popular tool for archiving files

- Archives ownership, timestamps, etc. as well as a file's data

- Options include:

Option	Meaning
c	Create an archive
t	List the contents of an archive
x	Extract files from an archive. Files are extracted to the current directory
r	Append to archive
u	Append to archive if file newer than copy already in archive (You must specify exactly one of c, t, x, r or u)
f	The following argument specifies the name of the archive
v	Verbose mode: list the files being added / extracted
z	Compress/uncompress the archive using gzip compression
j	Compress/uncompress the archive using bz2 compression

ict@innovation

FOSSFA

`tar` examples

- Archive the current directory, writing the archive to `/tmp/foo.tar`:

  ```
  $ tar cf  /tmp/foo.tar .
  ```

- Archive the directory `/home/tux`, writing the archive (compressed) to `/tmp/tux.tar` and listing the files added to the archive

  ```
  $ tar czvf /tmp/tux.tar.gz  /home/tux
  ```

- List the files in the archive `foo.tar`:

  ```
  $ tar tvf /tmp/foo.tar
  ```

- Extract all the files from the compressed archive `tux.tar.gz`:

  ```
  $ tar xzvf /tmp/tux.tar.gz
  ```

- Extract one file from the archive

  ```
  $ tar xzvf /tmp/tux.tar.gz shopping
  ```

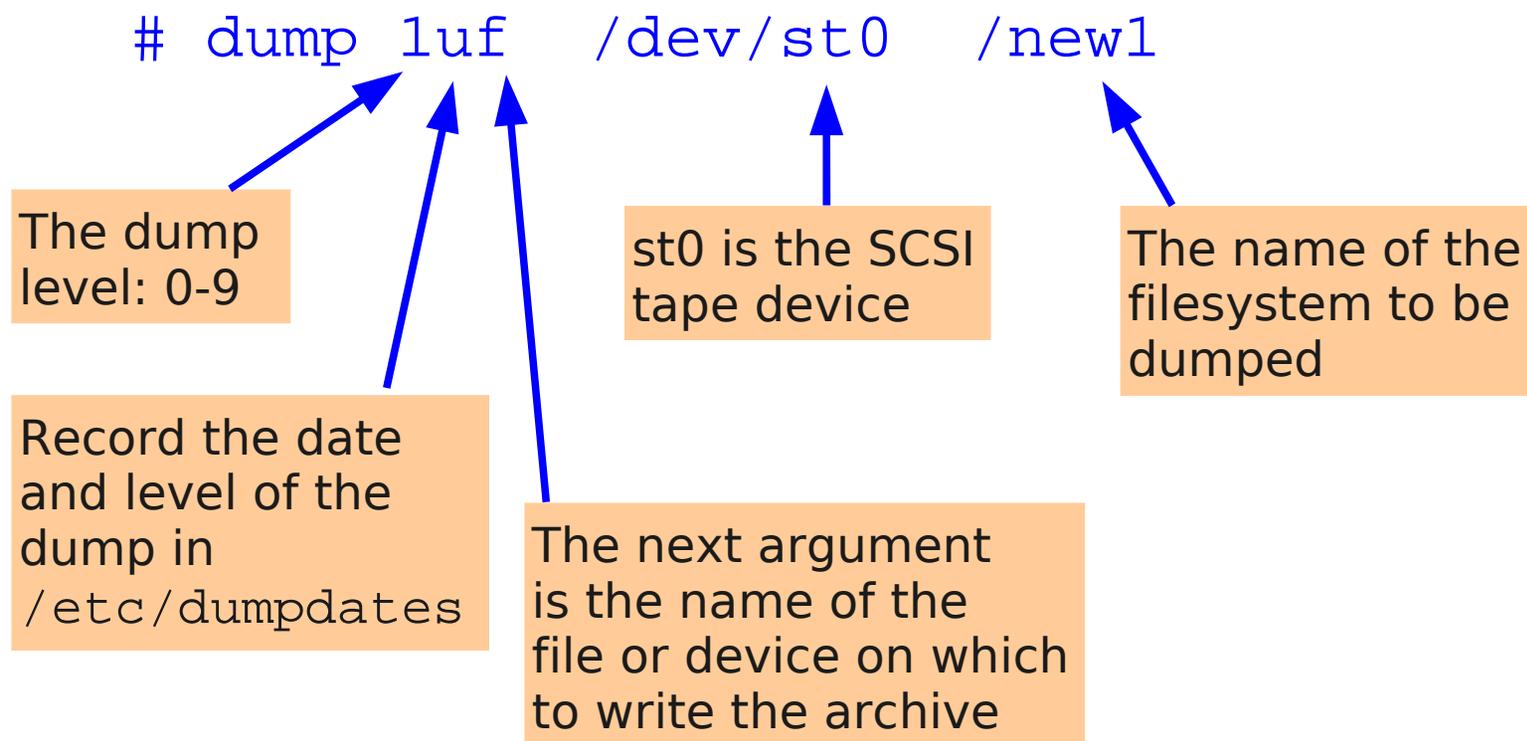ict@innovation

Exercise: Using `tar`

1. Following the examples in the notes, create two tar archives of the files in `/etc`:

   – The first should be called `/tmp/etc.tar` and should be uncompressed

   – The second should be called `/tmp/etc.tar.gz` and should be gzipped

2. Compare the sizes of the files `/tmp/etc.tar` and `/tmp/etc.tar.gz`

   – What compression factor is achieved? _____

3. List the files in the compressed archive and verify that the file `motd` is present (hint: use `grep`)

4. Remove the file `/etc/motd`, using `rm`

5. Restore the file `/etc/motd` from the compressed archive and verify that it is now present

• Congratulations! You have created a tar archive and restored a file from it.

ict@innovation

Dump and restore

- The programs `dump` and `restore` provide a way to efficiently backup a filesystem so that it can be restored following a catastrophic failure

- The contents of a dump depend upon the *dump level* (an integer in the range 0 to 9)

  - A level 0 dump dumps everything

  - A level 1 dump dumps everything not on the most recent level 0 dump

  - A level 2 dump dumps everything not on the most recent level 1 dump

  - … and so on

- Incremental dumps are efficient because most of a fileystem changes infrequently; an incremental dump only contains the changes since the most recent dump at a lower level

- When restoring a file system from an incremental dump, start with the level 0 dump and work through the most recent dumps at each higher level.

ict@innovation

FOSSFA

Creating a `dump`

- A typical `dump` command:

    ```
    # dump 1uf  /dev/st0  /new1
    ```

    The dump level: 0-9

    Record the date and level of the dump in `/etc/dumpdates`

    The next argument is the name of the file or device on which to write the archive

    st0 is the SCSI tape device

    The name of the filesystem to be dumped

- It is recommended that a filesystem should be unmounted (or mounted read-only) whilst a dump is in progress

ict@innovation

Dump regime

- A pre-planned regime is needed for effective use of `dump`

- Example

    - Do a level 0 dump every month

    - Do a level 1 dump every week

    - Do a level 2 dump every day

    - Keep enough tapes/CDs that you can restore even if one is faulty

- Label the tapes or CDs (*not* the boxes!)

- Routinely verify that you can read the backups

- Do not try to use media beyond their working life

- Keep backup media off-site

    - Consider the security implications – anyone with physical access to the media can read them

Doing a full `restore`

- The `restore` command is used to recover files from a dump

  - To restore an entire partition after a catastrophic failure

  - To restore individual files

- To restore an entire partition:

  - Rebuild an empty file system on the partition

  - Restore each dump level, starting with level 0

- Example:

```
# mke2fs  /dev/hda5
# mount /dev/hda5 /new1
# cd /new1
# restore rf /dev/st0
# restore rf /dev/st0
```

Build a new file system

Restore the level 0 dump

Restore the level 1 dump

ict@innovation

FOSSFA

Using `restore` interactively

- `restore -i` starts an interactive mode and provides a shell-like interface that lets you move around within the directory structure on the dump, list the files, and select the ones you want to extract

- Interactive commands include:

  - `cd`   Change directory (within the dump)

  - `ls`   List the files in the current directory (within the dump)

  - `add`   Add a file to the list of files to be extracted

  - `delete`   Remove a file from the list of files to be extracted

  - `extract` Extract the selected files

  - `quit`   Exit from restore

ict@innovation

Exercise: Using `dump` and `restore`

- In this exercise, we will make a level 0 dump of the `/new1` partition. Then, we will destroy the contents of the partition (by re-building an empty file system on it) and restore the contents from the dump

1. Following the example in the notes, make a level 0 dump of the filesystem on `/new1`, storing the dump in a file in `/tmp`

2. Unmount the file system mounted on `/new1`

3. Destroy the file system on `/dev/hda5` by creating a new, empty file system on it. Take care to use the correct partition name:

   `# mke2fs /dev/hda5`

4. Remount the `hda5` partition onto `/new1` and list the contents to verify that it is empty

5. Following the example in the notes, `cd` into `/new1` and restore the level 0 dump. Verify that the files have been restored.

ict@innovation

Third-party backup solutions

- There are many free and commercial backup products for linux
  - Build on, or replace traditional tools such as `tar` and `dump/restore`
  - Many offer backups across the network to a central backup server
- Amanda, originally from Univ. of Maryland (`www.amanda.org`)
- Afbackup (`sourceforge.net/projects/afbackup`)
- Integrity (`www.availsolutions.com`)
- Mondo (`www.microwerks.net/~hugo`)
- many others ...

Quiz

- What does "RPM" stand for?

- What command would you use to obtain a list of the files contained in the (uninstalled) package `foo-1.2-23.rpm`?

- What command will list all the RPM packages currently installed?

- Give one advantage, and one disadvantage, of installing from source code, compared with installing from a binary RPM

- What commands would you use to extract a compressed tar archive called `foo.tar.gz` into the directory `/tmp/build`?

- True or false?

  - When you create a dump, the lower the dump level, the bigger the dump will be

  - You need the gcc compiler installed before you can install RPM packages

ict@innovation