Manipulating Text Files

- Irrespective of the role you play with Linux (system administrator, developer or user), you often need to browse through and parse text files, and/or extract data from them
- These are **file manipulation operations** and it is essential for the Linux user to become adept at performing certain operations on files
- Most of the time, such file manipulation is done at the command line, which allows users to perform tasks more efficiently than while using a GUI; the command line is also more suitable for automating often executed tasks
- Experienced system administrators write customized scripts to accomplish such repetitive tasks, standardized for each particular environment

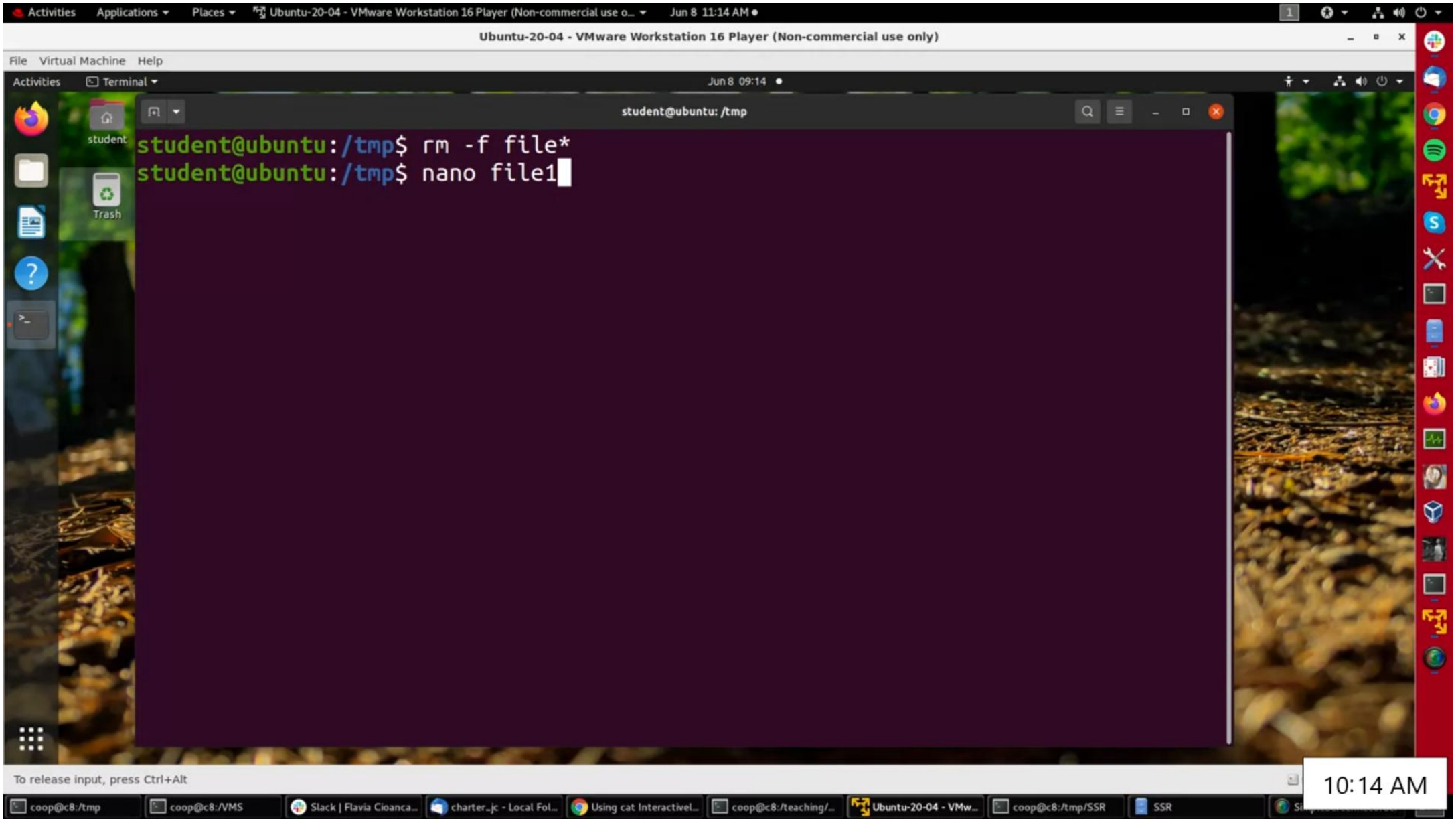
Manipulating Text Files

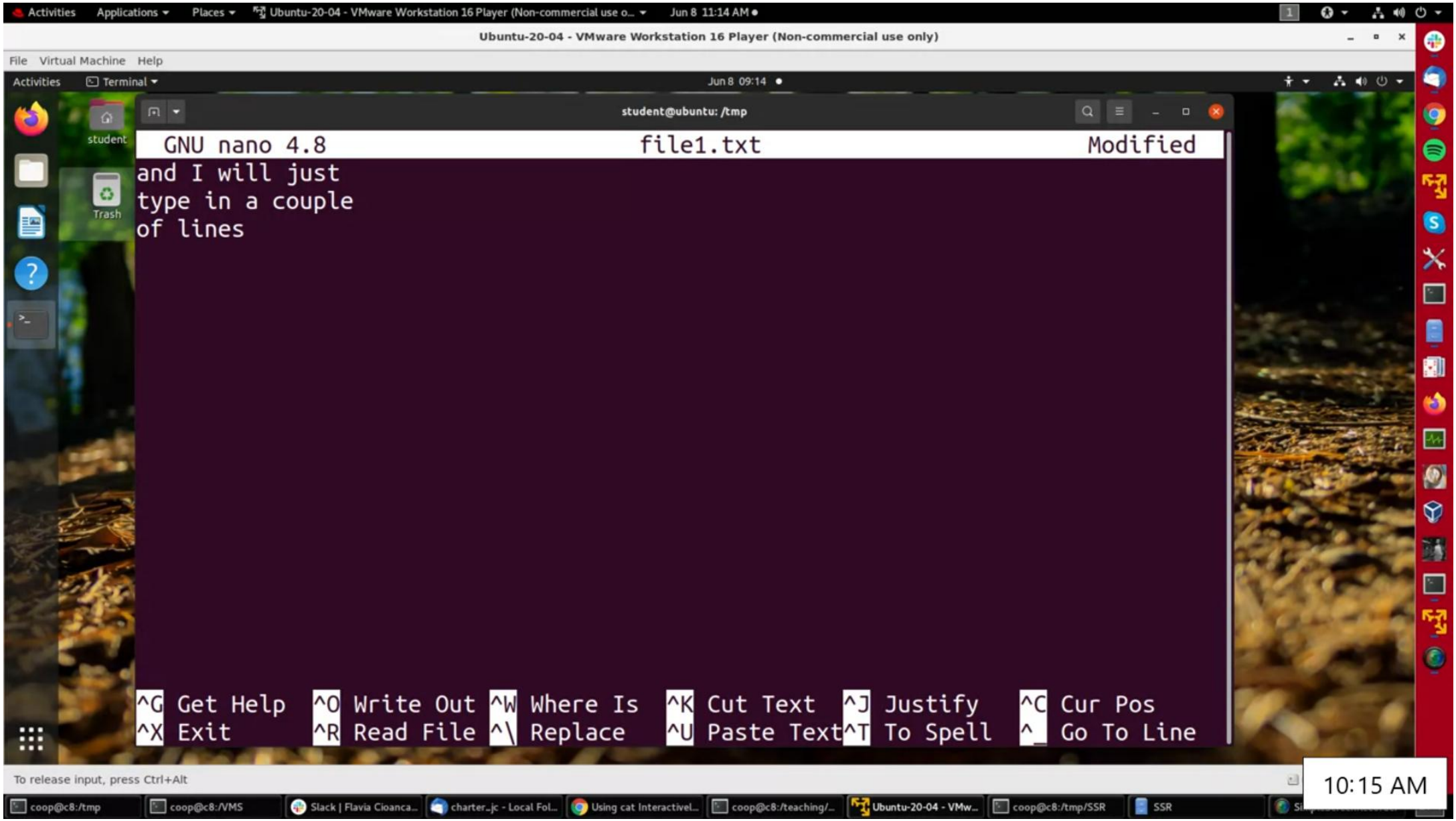
- Irrespective of the role you play with Linux (system administrator, developer or user), you often need to browse through and parse text files, and/or extract data from them
- These are **file manipulation operations** and it is essential for the Linux user to become adept at performing certain operations on files
- Most of the time, such file manipulation is done at the command line, which allows users to perform tasks more efficiently than while using a GUI; the command line is also more suitable for automating often executed tasks
- Experienced system administrators write customized scripts to accomplish such repetitive tasks, standardized for each particular environment

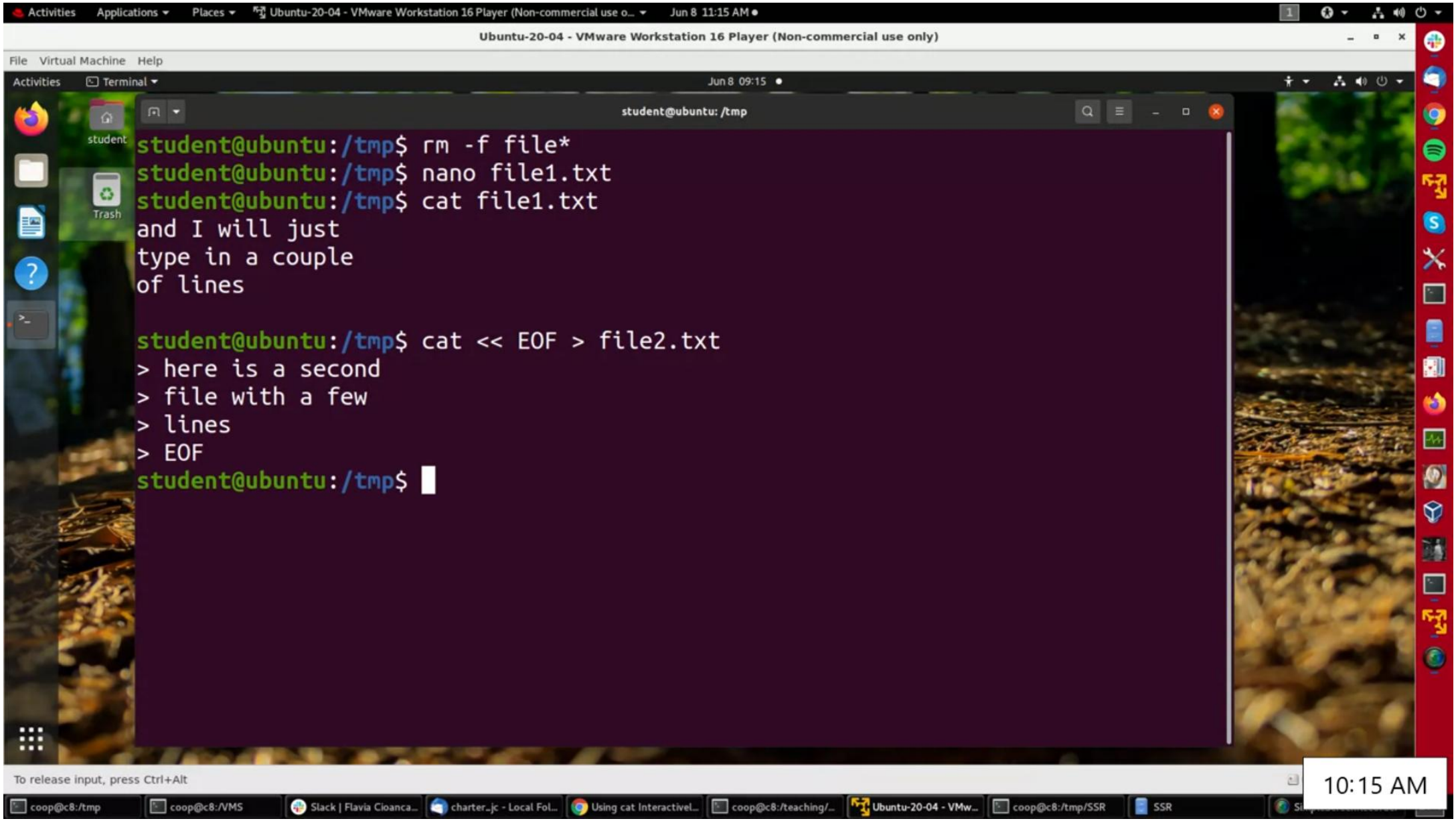
Command Line Utilities

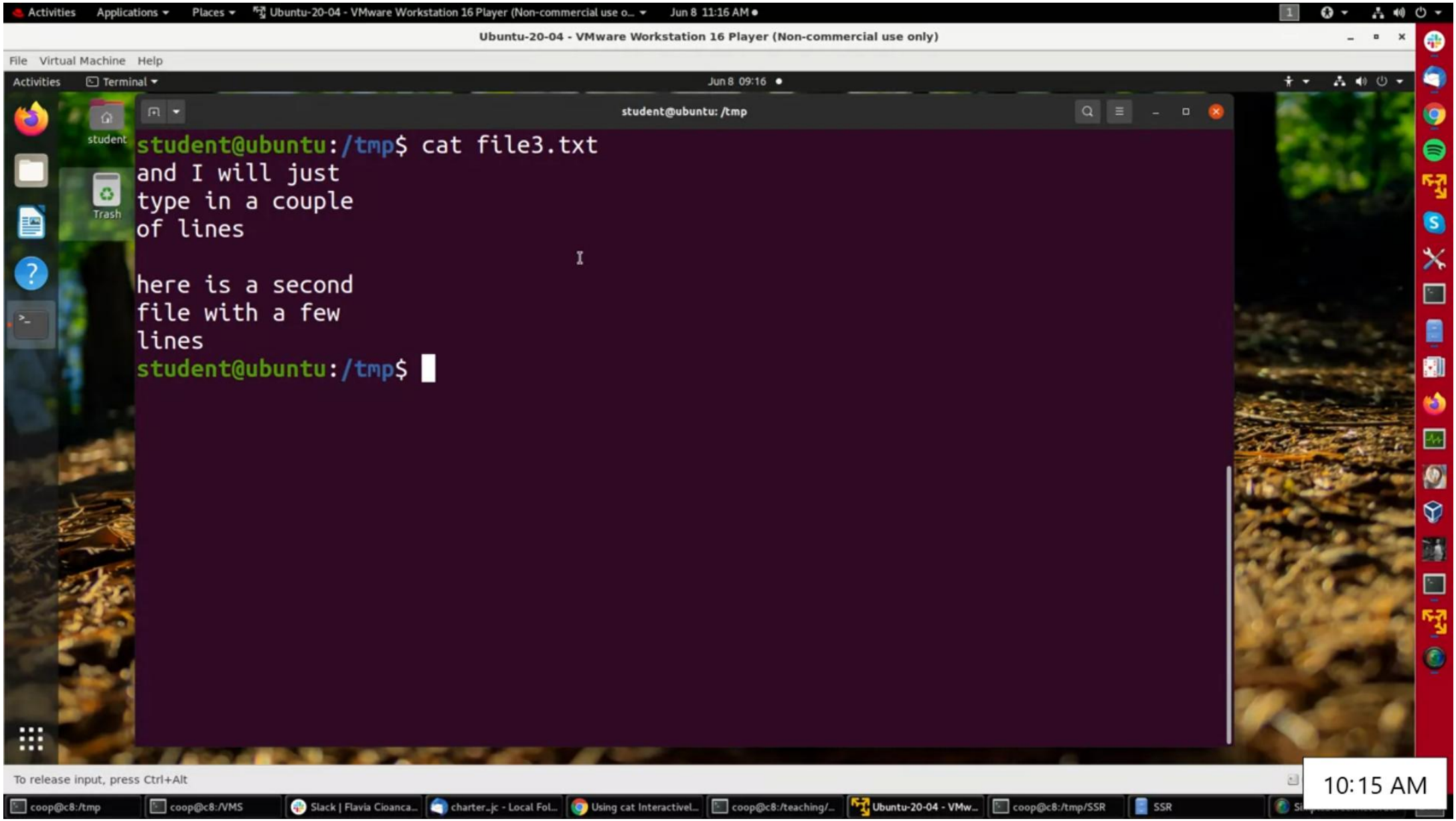
The basic Linux command line utilities are:

- **`cat`**
- **`echo`**
- **`head`**
- **`tail`**
- **`sed`**
- **`awk`**


Activities
Applications
Places
Ubuntu-20-04 - VMware Workstation 16 Player (Non-commercial use o...
Jun 8 11:14 AM
Ubuntu-20-04 - VMware Workstation 16 Player (Non-commercial use only)
File
Virtual Machine
Help
Activities
Terminal
Jun 8 09:14
student@ubuntu: /tmp
student@ubuntu:/tmp$ rm -f file*
student@ubuntu:/tmp$ nano file1
student
Trash
To release input, press Ctrl+Alt
10:14 AM
coop@c8:/tmp
coop@c8:/VMS
Slack | Flavia Cioanca...
charter_jc - Local Fol...
Using cat Interactivel...
coop@c8:/teaching/...
Ubuntu-20-04 - VMw...
coop@c8:/tmp/SSR
SSR


Activities
Applications
Places
Ubuntu-20-04 - VMware Workstation 16 Player (Non-commercial use o...
Jun 8 11:14 AM
Ubuntu-20-04 - VMware Workstation 16 Player (Non-commercial use only)
File Virtual Machine Help
Activities
Terminal
Jun 8 09:14
student@ubuntu: /tmp
GNU nano 4.8 file1.txt Modified
and I will just
type in a couple
of lines
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
student
Trash
To release input, press Ctrl+Alt
10:15 AM
coop@c8:/tmp
coop@c8:/VMS
Slack | Flavia Cioanca...
charter_jc - Local Fol...
Using cat Interactivel...
coop@c8:/teaching/...
Ubuntu-20-04 - VMw...
coop@c8:/tmp/SSR
SSR


Activities
Applications
Places
Ubuntu-20-04 - VMware Workstation 16 Player (Non-commercial use o...
Jun 8 11:15 AM
Ubuntu-20-04 - VMware Workstation 16 Player (Non-commercial use only)
File
Virtual Machine
Help
Activities
Terminal
Jun 8 09:15
student@ubuntu: /tmp
student
Trash
student@ubuntu:/tmp$ rm -f file*
student@ubuntu:/tmp$ nano file1.txt
student@ubuntu:/tmp$ cat file1.txt
and I will just
type in a couple
of lines
student@ubuntu:/tmp$ cat << EOF > file2.txt
> here is a second
> file with a few
> lines
> EOF
student@ubuntu:/tmp$
To release input, press Ctrl+Alt
10:15 AM
coop@c8:/tmp
coop@c8:/VMS
Slack | Flavia Cioanca...
charter_jc - Local Fol...
Using cat Interactivel...
coop@c8:/teaching/...
Ubuntu-20-04 - VMw...
coop@c8:/tmp/SSR
SSR


Activities
Applications
Places
Ubuntu-20-04 - VMware Workstation 16 Player (Non-commercial use o...
Jun 8 11:16 AM
Ubuntu-20-04 - VMware Workstation 16 Player (Non-commercial use only)
File
Virtual Machine
Help
Activities
Terminal
Jun 8 09:16
student@ubuntu: /tmp
student
Trash
student@ubuntu:/tmp$ cat file3.txt
and I will just
type in a couple
of lines
here is a second
file with a few
lines
student@ubuntu:/tmp$
To release input, press Ctrl+Alt
10:15 AM
coop@c8:/tmp
coop@c8:/VMS
Slack | Flavia Cioanca...
charter_jc - Local Fol...
Using cat Interactivel...
coop@c8:/teaching/...
Ubuntu-20-04 - VMw...
coop@c8:/tmp/SSR
SSR

Managing Large Files

- System administrators need to work with configuration files, text files, documentation files, and log files
- Some of these files may be large or become quite large as they accumulate data with time
- These files will require both viewing and administrative updating

10:15 AM

Example

A banking system might maintain one simple large log file to record details of all of one day's ATM transactions. Due to a security attack or a malfunction, the administrator might be forced to check for some data by navigating within the file.

In such cases, directly opening the file in an editor will cause issues, due to high memory utilization, as an editor will usually try to read the whole file into memory first.

What can be a possible solution?

`less` and `somefile`

- One can use **`less`** to view the contents of such a large file, scrolling up and down page by page, without the system having to place the entire file in memory before starting; this is much faster than using a text editor
- Viewing **`somefile`** can be done by typing either of the two following commands:

```
$ less somefile
$ cat somefile | less
```

- By default, man pages are sent through the **`less`** command
- You may have encountered the older **`more`** utility which has the same basic function but fewer capabilities: i.e. less is more!


Press Esc to exit full screen
Introduction to sed and awk
THE LINUX FOUNDATION
0:02 / 2:31
10:17 AM

Introduction to **sed** and **awk**

- It is very common to create and then repeatedly edit and/or extract contents from a file; in order to perform such operations you can use **sed** and **awk**
- Many Linux users and administrators will write scripts using comprehensive scripting languages such as **Python** and **perl**, rather than use **sed** and **awk** (and some other utilities we will discuss later)
- Using such utilities is certainly fine in most circumstances; you should always feel free to use the tools you are experienced with
- However, the utilities that are described here are much lighter; i.e. they use fewer system resources, and execute faster
- There are situations (such as during booting the system) where a lot of time would be wasted using the more complicated tools, and the system may not even be able to run them; the simpler tools will always be needed

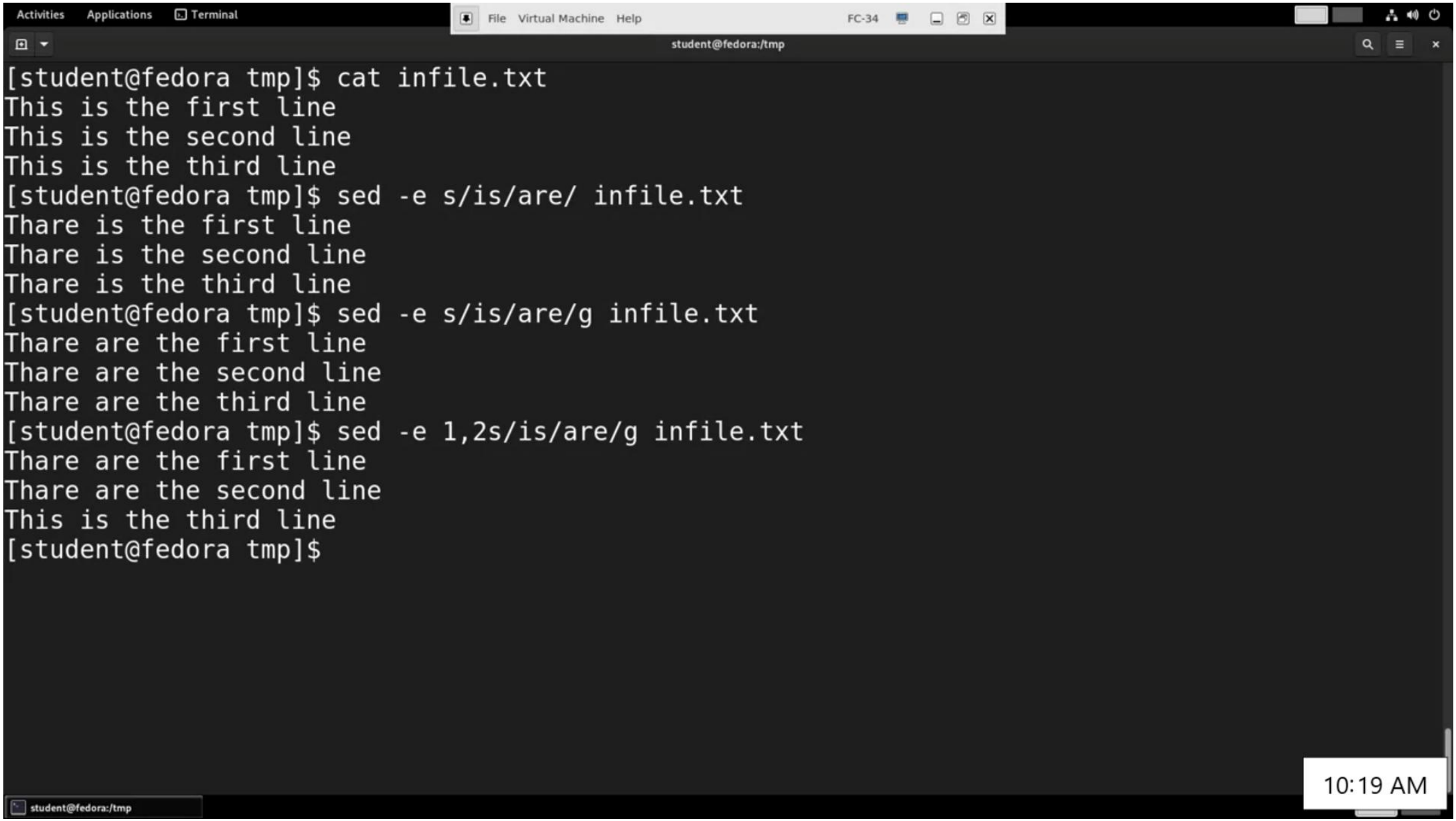
sed

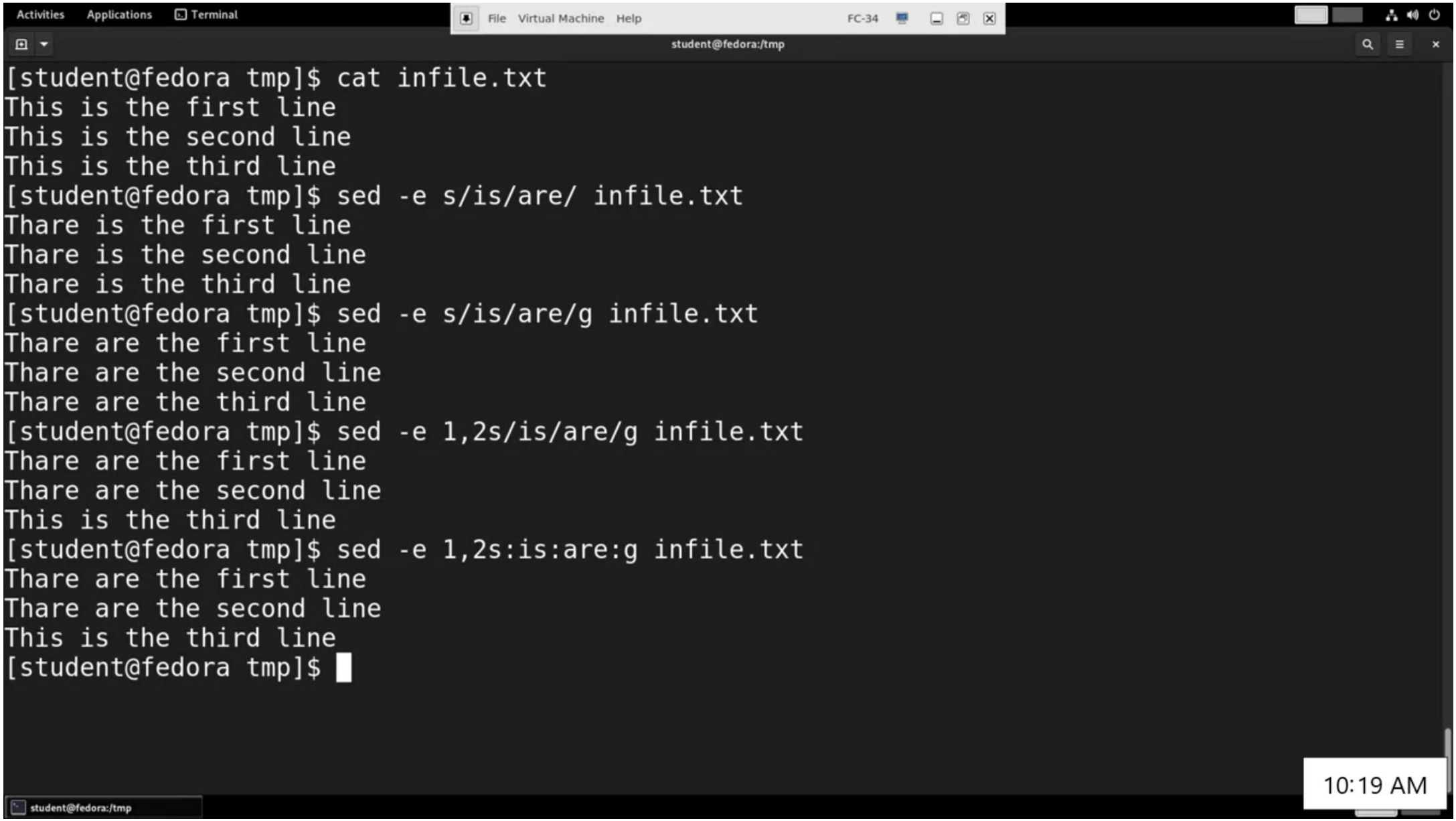
- **sed** is a powerful text processing tool and one of the oldest, earliest and most popular UNIX utilities; its name is an abbreviation for stream editor
- It is used to modify the contents of a file, usually placing the contents into a new file, and it can also filter text, as well as perform substitutions in data streams
- How it works:
  - Data from an input source/file (or stream) is taken and moved to a working space
  - The entire list of operations/modifications is applied over the data in the working space and the final contents are moved to the standard output space (or stream)

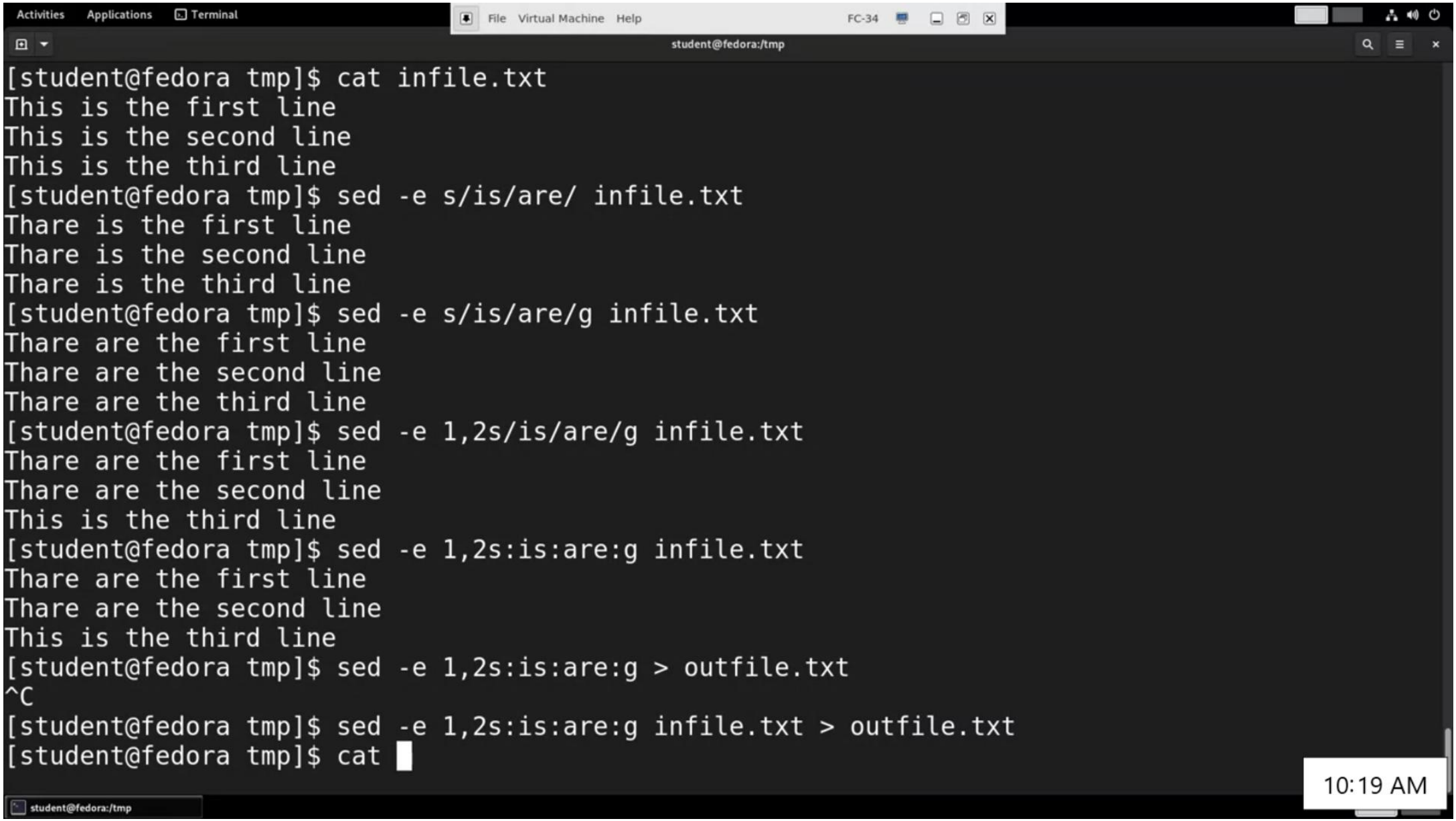
10:17 AM

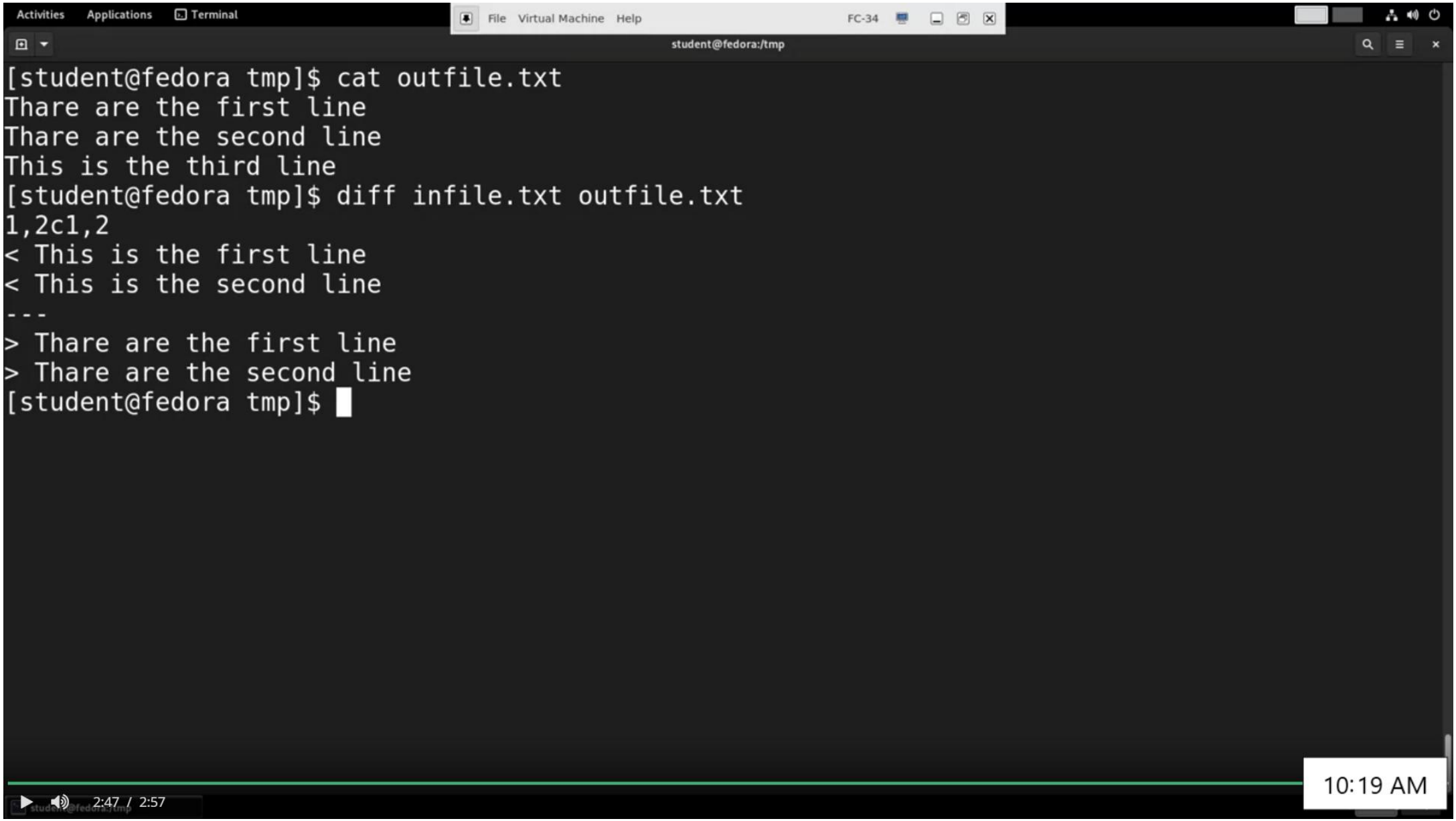
awk

- **awk** was created at Bell Labs in the 1970s and derived its name from the last names of its authors: Alfred Aho, Peter Weinberger, and Brian Kernighan
  - Used to extract and then print specific contents of a file and is often used to construct reports
  - A powerful utility and interpreted programming language
  - Used to manipulate data files, retrieving, and processing text
  - Works well with fields (containing a single piece of data, essentially a column) and records (a collection of fields, essentially a line in a file)


Activities
Applications
Terminal
File
Virtual Machine
Help
FC-34
student@fedora:/tmp
[student@fedora tmp]$ cat infile.txt
This is the first line
This is the second line
This is the third line
[student@fedora tmp]$ sed -e s/is/are/ infile.txt
Thare is the first line
Thare is the second line
Thare is the third line
[student@fedora tmp]$ sed -e s/is/are/g infile.txt
Thare are the first line
Thare are the second line
Thare are the third line
[student@fedora tmp]$ sed -e 1,2s/is/are/g infile.txt
Thare are the first line
Thare are the second line
This is the third line
[student@fedora tmp]$
student@fedora:/tmp
10:19 AM


Activities
Applications
Terminal
File
Virtual Machine
Help
FC-34
student@fedora:/tmp
[student@fedora tmp]$ cat infile.txt
This is the first line
This is the second line
This is the third line
[student@fedora tmp]$ sed -e s/is/are/ infile.txt
Thare is the first line
Thare is the second line
Thare is the third line
[student@fedora tmp]$ sed -e s/is/are/g infile.txt
Thare are the first line
Thare are the second line
Thare are the third line
[student@fedora tmp]$ sed -e 1,2s/is/are/g infile.txt
Thare are the first line
Thare are the second line
This is the third line
[student@fedora tmp]$ sed -e 1,2s:is:are:g infile.txt
Thare are the first line
Thare are the second line
This is the third line
[student@fedora tmp]$
student@fedora:/tmp
10:19 AM


Activities
Applications
Terminal
File
Virtual Machine
Help
FC-34
student@fedora:/tmp
[student@fedora tmp]$ cat infile.txt
This is the first line
This is the second line
This is the third line
[student@fedora tmp]$ sed -e s/is/are/ infile.txt
Thare is the first line
Thare is the second line
Thare is the third line
[student@fedora tmp]$ sed -e s/is/are/g infile.txt
Thare are the first line
Thare are the second line
Thare are the third line
[student@fedora tmp]$ sed -e 1,2s/is/are/g infile.txt
Thare are the first line
Thare are the second line
This is the third line
[student@fedora tmp]$ sed -e 1,2s:is:are:g infile.txt
Thare are the first line
Thare are the second line
This is the third line
[student@fedora tmp]$ sed -e 1,2s:is:are:g > outfile.txt
^C
[student@fedora tmp]$ sed -e 1,2s:is:are:g infile.txt > outfile.txt
[student@fedora tmp]$ cat
student@fedora:/tmp
10:19 AM


Activities
Applications
Terminal
File
Virtual Machine
Help
FC-34
student@fedora:/tmp
[student@fedora tmp]$ cat outfile.txt
Thare are the first line
Thare are the second line
This is the third line
[student@fedora tmp]$ diff infile.txt outfile.txt
1,2c1,2
< This is the first line
< This is the second line

> Thare are the first line
> Thare are the second line
[student@fedora tmp]$
2:47 / 2:57
10:19 AM

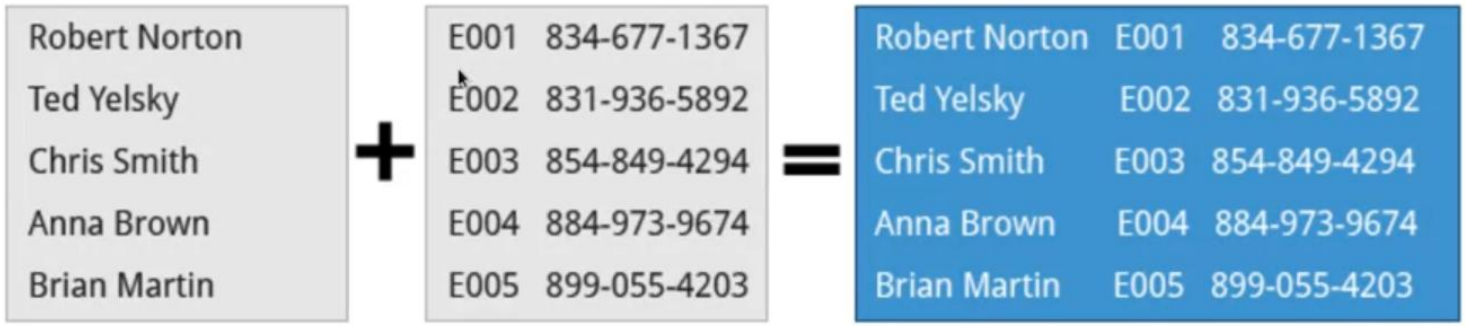
File Manipulation Utilities

- In managing your files, you may need to perform many tasks, such as sorting data and copying data from one location to another
- Linux provides several file manipulation utilities that you can use while working with text files, including:
  - **`sort`**
  - **`uniq`**
  - **`paste`**
  - **`join`**
  - **`split`**

Example

Suppose you have a file that contains the full name of all employees and another file that lists their phone numbers and Employee IDs. You want to create a new file that contains all the data listed in three columns: name, employee ID, and phone number.

How can you do this effectively without investing too much time?



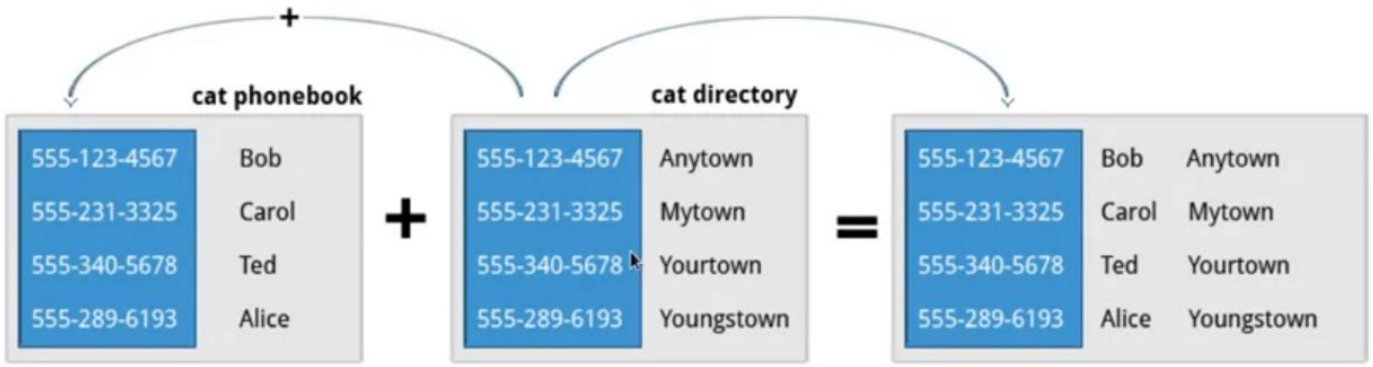
paste

- **paste** can be used to create a single file containing all three columns
- The different columns are identified based on delimiters (spacing used to separate two fields); for example, delimiters can be a blank space, a tab, or an Enter
- In the image provided, a single space is used as the delimiter in all files
- **paste** accepts the following options:
  - **-d** delimiters: specify a list of delimiters to be used instead of tabs for separating consecutive values on a single line; each delimiter is used in turn and when the list has been exhausted, **paste** begins again at the first delimiter
  - **-s** causes **paste** to append the data in series rather than in parallel; that is, in a horizontal rather than vertical fashion

Example

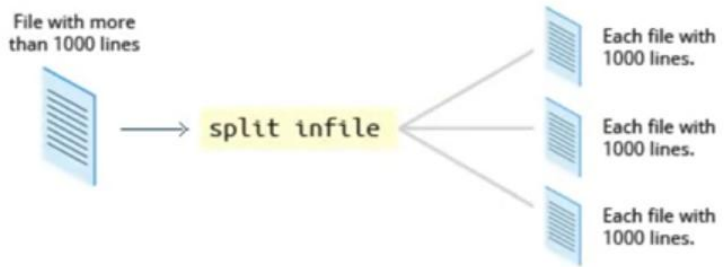
Suppose you have two files with some similar columns. You have saved employees' phone numbers in two files, one with their first name and the other with their last name. You want to combine the files without repeating the data of common columns.

How do you achieve this?



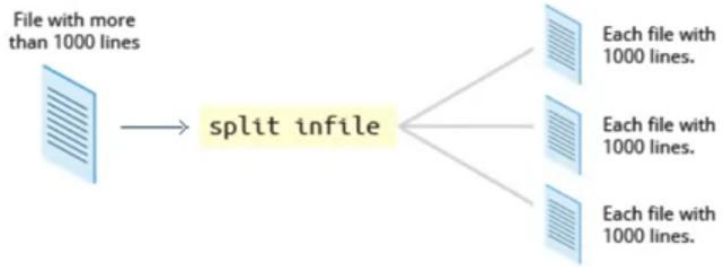
split

- **split** is used to break up (or split) a file into equal-sized segments for easier viewing and manipulation, and is generally used only on relatively large files
- By default, **split** breaks up a file into 1000-line segments; the original file remains unchanged, and a set of new files with the same name plus an added prefix is created
  - By default, the **x** prefix is added
  - To split a file into segments, use the command **split infile**
  - To split a file into segments using a different prefix, use the command **split infile <Prefix>**



split

- **split** is used to break up (or split) a file into equal-sized segments for easier viewing and manipulation, and is generally used only on relatively large files
- By default, **split** breaks up a file into 1000-line segments; the original file remains unchanged, and a set of new files with the same name plus an added prefix is created
  - By default, the **x** prefix is added
  - To split a file into segments, use the command **split infile**
  - To split a file into segments using a different prefix, use the command **split infile <Prefix>**



bash Script

After preparing a bash script and putting in a file, say **my_script.sh**, it can be invoked in either of two ways:

- Typing **bash my_script.sh**
- Have the first line of **my_script.sh** be **#!/bin/sh**
  - This will work with any other interpreter such as csh, perl, etc.
  - This is an exception to the rule that the character **#** is used to demark comments

  Then make the script executable by doing **chmod +x my_script.sh** and then just running **./my_script.sh**

Environment Variables

- There are some special environment variables that can be used inside a script:
  - **$0** is the command name
  - **$1 $2** ... are the command arguments
  - **$*** represents them all
  - **$@** represents them all, preserving the grouping of quoted arguments
  - **$#** gives the number of arguments
- Note that you should often enclose these variables in double quotes; i.e. you should say "**$@**" to preserve the argument grouping
- In addition, you will get syntax errors when doing comparisons if the string is empty if you do not use double quotes

Functions

- bash permits use of subprograms or functions
- Alternatively, one script can call another script, but this can make passing information a little more complex, and increases the number of files; proper use of functions can make scripts much simpler
- The basic form of a function is:

```
fun_foobar(){
      statements
}
```

which will not look strange to any C programmer

Functions (Cont.)

A few things to remember about functions include:

- No arguments are enclosed in the parentheses
- Functions must always be defined **before** they are used, because scripts are interpreted, not compiled
- Functions leave positional arguments unchanged, but can reset other variables; a **`local var1 var2`** ... statement can make variables local

Types of Files

- Normal files (**-**) and directories (**d**) are the basic entries in the filesystem structure
- Symbolic links (**l**) point somewhere else on the system, such that a reference to either the link or its target is equivalent
- Named pipes (**p**), also called FIFOs, for First-In, First-Out, are used for inter-process communication
- Unix domain sockets (**s**) are used for a similar purpose, but do this through networking infrastructure, and have some more capabilities than FIFOs

Types of Files (Cont.)

- Block device nodes (**b**) and character device nodes (**c**) are used to communicate with either hardware devices, or software pseudo-devices
  - Block devices do reads and writes in predetermined fixed size chunks, and usually correspond to storage media such as disks and CD-ROMs; the I/O is generally buffered and cached
  - Character devices are addressed with streams of data, and correspond to devices such as sound cards, serial ports, etc; the I/O is generally not buffered or cached

  Device nodes are almost always placed in the **/dev** directory

Everything Is a File

- In describing UNIX-like operating systems, such as Linux, one often hears the old saying: *Everything is a file*
- This fits in with the philosophy that the same basic tools can be used for a wide variety of purposes, whether you are dealing with normal files, devices, etc., and can be strung together to accomplish non-trivial tasks.
- There are exceptions to this model; for example, network devices in Linux have no corresponding filesystem entry
- Note that in Linux file extensions rarely play a defining role; the type of a file is determined by examining its contents rather than its extension

Everything Is a File

- In describing UNIX-like operating systems, such as Linux, one often hears the old saying: *Everything is a file*
- This fits in with the philosophy that the same basic tools can be used for a wide variety of purposes, whether you are dealing with normal files, devices, etc., and can be strung together to accomplish non-trivial tasks.
- There are exceptions to this model; for example, network devices in Linux have no corresponding filesystem entry
- Note that in Linux file extensions rarely play a defining role; the type of a file is determined by examining its contents rather than its extension

Access Rights

- When you do an **`ls -l as`** in:

  ```
  $ ls -l a_file
  -rw-rw-r-- 1 coop aproject 1601 Mar 9 15:04 a_file
  ```

  after the first character there are nine more which indicate the access rights granted to potential file users
- These are arranged in three groups of three:
  - **owner**: the user who owns the file (also called **user**)
  - **group**: the group of users who have access
  - **world**: the rest of the world (also called **other**)

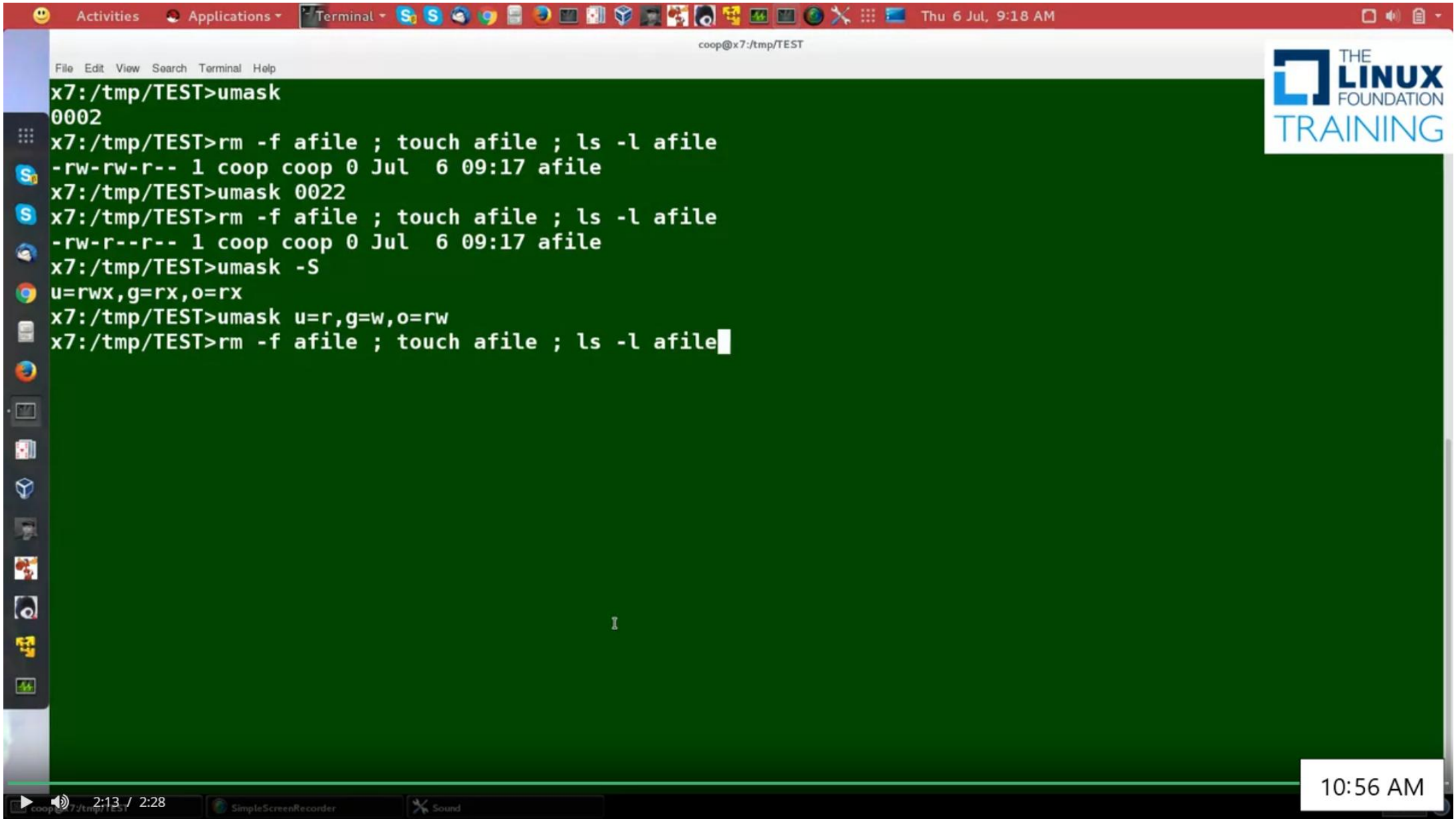
  In the above listing, the user is **`coop`** and the group is **`aproject`**

read, write, execute

- Each of the triplets can have each of the following values set:
  - r: read access is allowed
  - w: write access is allowed
  - x: execute access is allowed
- If the permission is not allowed, a - appears instead of one of these characters
- Thus, in the preceding example,

```
$ ls -l a_file
-rw-rw-r-- 1 coop aproject 1601 Mar 9 15:04 a_file
```

the user **coop** and members of the group **aproject** have read and write access, while anyone else has only read access


Activities
Applications
Terminal
Thu 6 Jul, 9:18 AM
coop@x7:/tmp/TEST
File Edit View Search Terminal Help
x7:/tmp/TEST>umask
0002
x7:/tmp/TEST>rm -f afile ; touch afile ; ls -l afile
-rw-rw-r-- 1 coop coop 0 Jul 6 09:17 afile
x7:/tmp/TEST>umask 0022
x7:/tmp/TEST>rm -f afile ; touch afile ; ls -l afile
-rw-r--r-- 1 coop coop 0 Jul 6 09:17 afile
x7:/tmp/TEST>umask -S
u=rwx,g=rx,o=rx
x7:/tmp/TEST>umask u=r,g=w,o=rw
x7:/tmp/TEST>rm -f afile ; touch afile ; ls -l afile
THE LINUX FOUNDATION TRAINING
2:13 / 2:28
SimpleScreenRecorder
Sound
10:56 AM

Virtual File System (VFS)

- Linux implements a **Virtual File System** (**VFS**), as do all modern operating systems
- For the most part neither the specific filesystem or actual physical media and hardware need be addressed by filesystem operations
- Furthermore, network filesystems (such as NFS) can be handled transparently
- This permits Linux to work with more filesystem varieties than any other operating system
- This democratic attribute has been a large factor in its success
- Most filesystems have full read/write access while a few have only read access and perhaps experimental write access

Virtual File System (Cont.)

- Some filesystem types, especially non-UNIX based ones, may require more manipulation in order to be represented in the VFS
- For example, variants such as vfat do not have distinct read/write/execute permissions for the owner/group/world fields; the VFS has to make an assumption about how to specify distinct permissions for the three types of user, and such behavior can be influenced by mounting operations
- Even more drastically, such filesystems store information about files in a File Allocation Table (FAT) at the beginning of the disk, rather than in the directories themselves, a basically different architectural method
- A number of newer high performance filesystems include full journaling capability

ext2, ext3, and ext4 Filesystems

- The native filesystems for Linux are **ext2** and its journaling descendants, **ext3** and **ext4**
- Each has its associated utility for formatting the filesystem (e.g. mkfs.ext3) and for checking the filesystem (e.g. fsck.ext4)
- Additionally, many parameters can be reset or tuned after filesystem creation with the program tune2fs; while defragmentation is generally not necessary, one can use e4defrag to do so
- The ext4 filesystem had its experimental designation removed with the 2.6.28 kernel release
- Pre-existing ext3 partitions could be migrated in place to ext4 without risk; with new features to be used only in subsequent file system operations
- One immediately obvious improvement is fast fsck; the speed of a filesystem check can easily go up by an order of magnitude or more

Journaling Filesystems

- Journaling filesystems recover from system crashes or ungraceful shutdowns with little or no corruption, and they do so very rapidly
- While this comes at the price of having some more operations to do, additional enhancements can more than offset the price
- In a journaling filesystem, operations are grouped into transactions
- A transaction must be completed without error, atomically; otherwise the filesystem is not changed
- A log file is maintained of transactions; when an error occurs, usually only the last transaction needs to be examined

Journaling Filesystems Freely Available Under Linux

The following journaling filesystems are freely available under Linux:

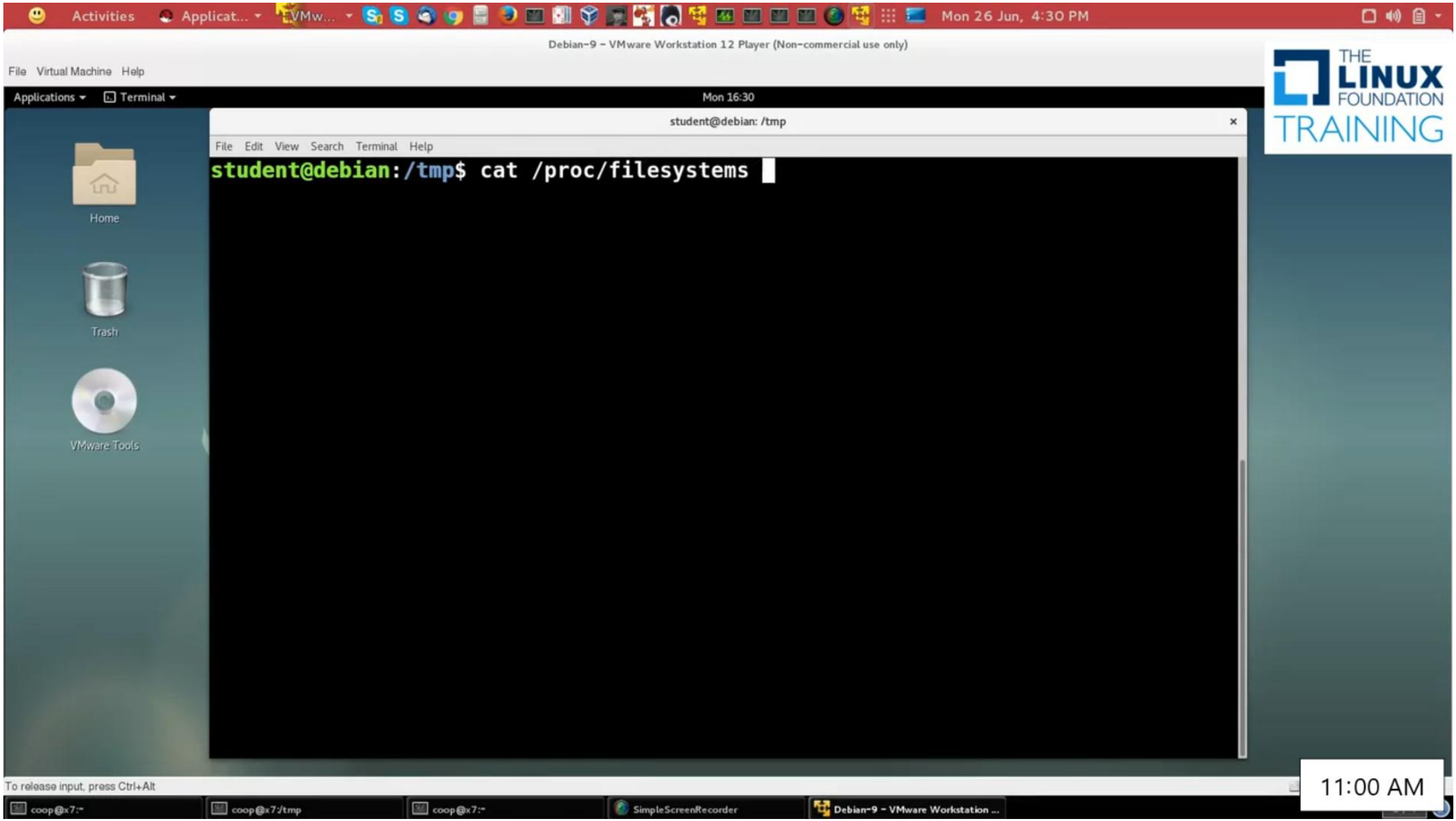
- The **ext3** filesystem is an extension of the ext2 filesystem
- The **ext4** filesystem is an extension of the ext3 filesystem, and was included in the mainline kernel first as an experimental branch, and then as a stable production feature in kernel 2.6.28 (features: extents, 48-bit block numbers, and up to 16 TB size)
- The **Reiser** filesystem was the first journaling implementation in Linux, but has lost its leadership and development has stalled
- The **JFS** filesystem is a product of IBM; has been ported from IBM's AIX OS
- The **XFS** filesystem is a product of SGI; has been ported from SGI's IRIX OS
- The **btrfs** filesystem (**B TR**ee filesystem) is the most recent, is native to Linux, and has many advanced features

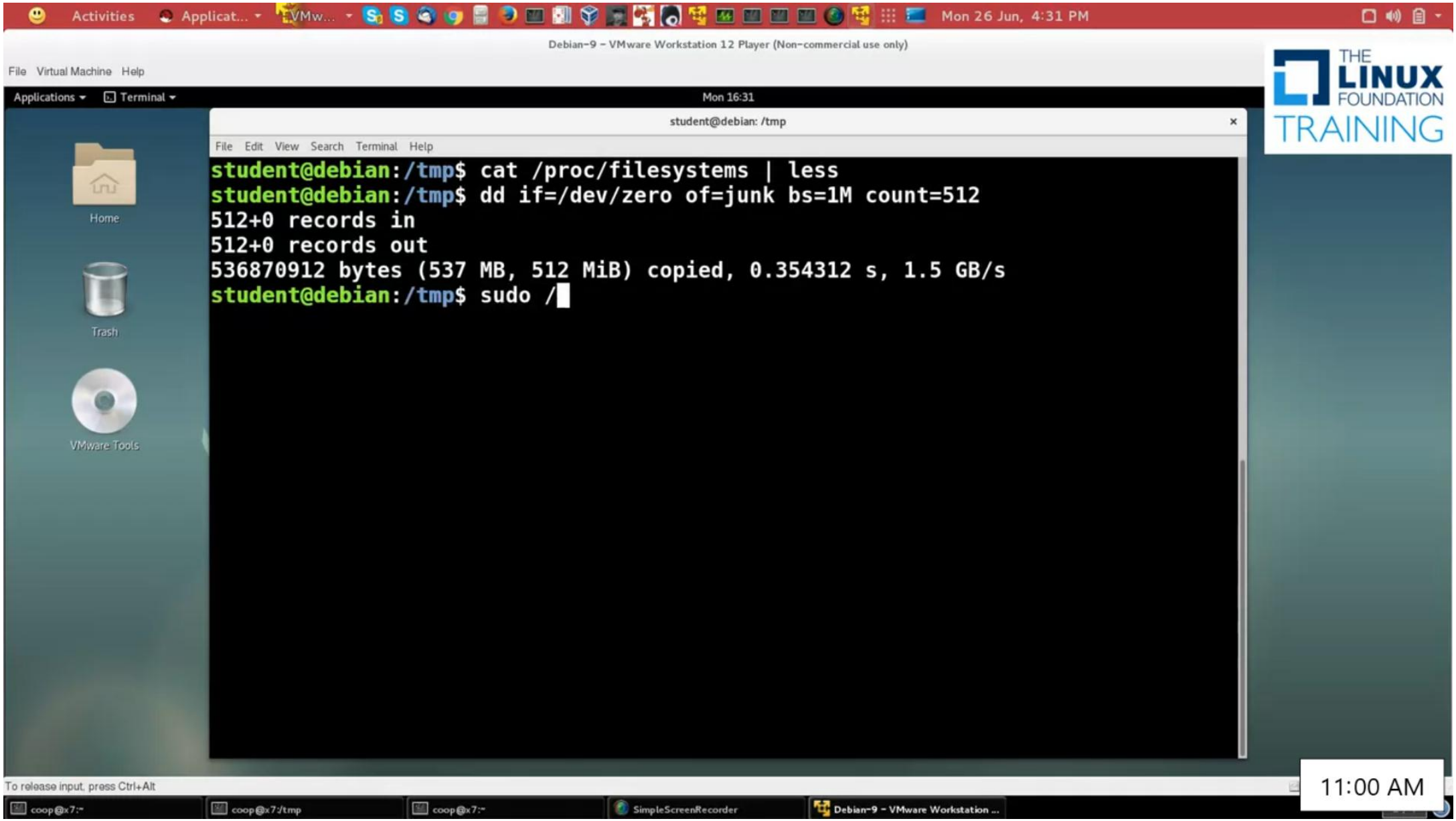
btrfs

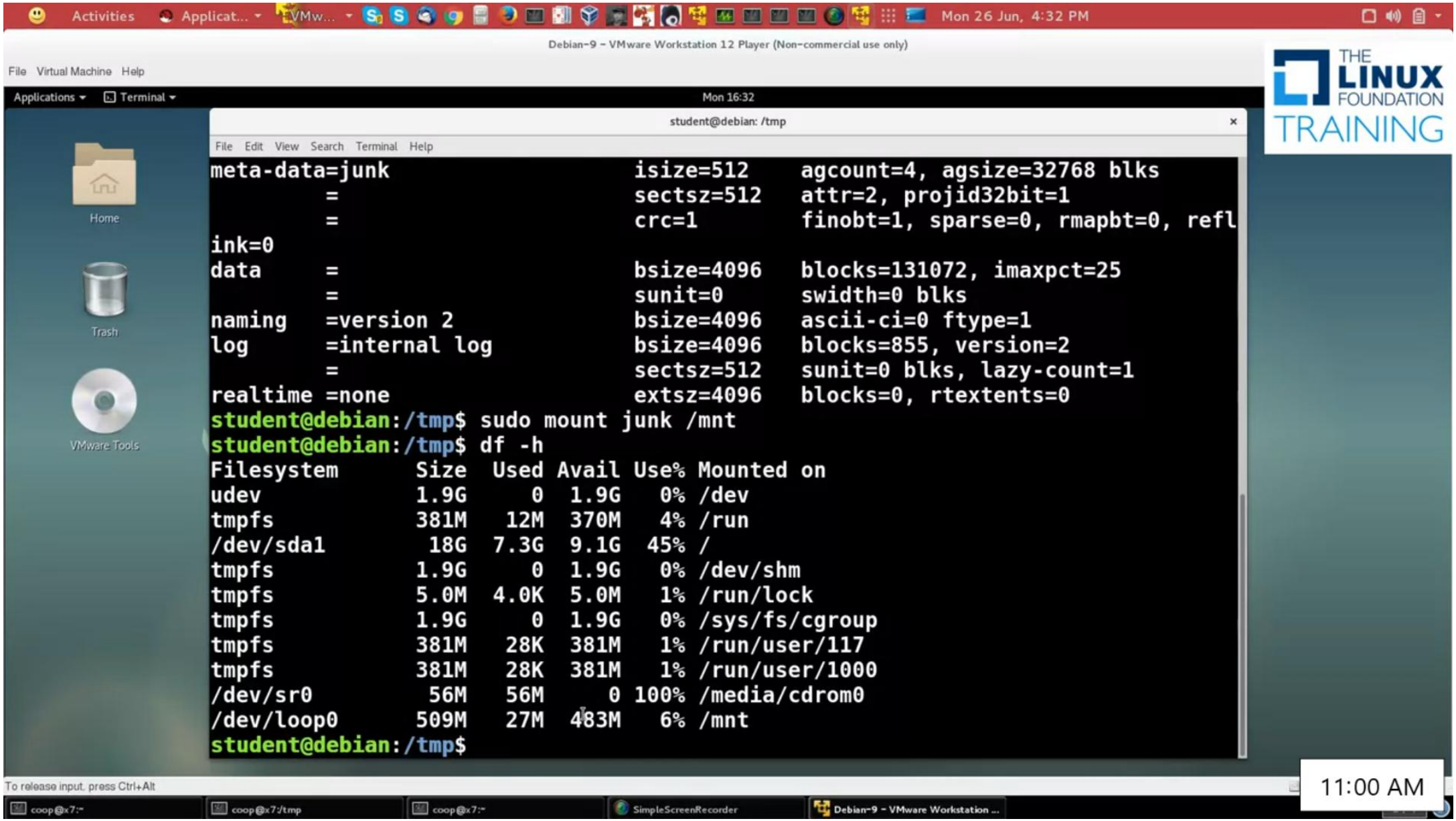
- Both Linux developers and Linux users with high performance and high capacity or specialized needs are following the development and gradual deployment of the **btrfs** filesystem, which was created by Chris Mason
- While btrfs has been in the mainline kernel since 2.6.29, it has generally been viewed as experimental; although it has been used in new products
- One of the main features is the ability to take frequent snapshots of entire filesystems, or sub-volumes of entire filesystems in virtually no time
- Because btrfs makes extensive use of **COW** (Copy on Write) techniques , such a snapshot does not involve any more initial space for data blocks or any I/O activity except for some metadata updating

btrfs (Cont.)

- One can easily revert to the state described by earlier snapshots and even induce the kernel to reboot off an earlier root filesystem snapshot
- btrfs maintains its own internal framework for adding or removing new partitions and/or physical media to existing filesystems, much as LVM (Logical Volume Management) does
- Before btrfs becomes suitable for day-to-day use in critical filesystems, some tasks require finishing; for example, as of now the fsck (filesystem checking) program is read-only, which means you can find out what is wrong, but you cannot easily fix it!


Activities
Applicat...
VMw...
Mon 26 Jun, 4:30 PM
Debian-9 - VMware Workstation 12 Player (Non-commercial use only)
File Virtual Machine Help
Applications
Terminal
Mon 16:30
student@debian: /tmp
File Edit View Search Terminal Help
student@debian:/tmp$ cat /proc/filesystems
Home
Trash
VMware Tools
THE LINUX FOUNDATION TRAINING
To release input, press Ctrl+Alt
coop@x7:~
coop@x7:/tmp
coop@x7:~
SimpleScreenRecorder
Debian-9 - VMware Workstation ...
11:00 AM


Activities
Applicat...
VMw...
Mon 26 Jun, 4:31 PM
Debian-9 - VMware Workstation 12 Player (Non-commercial use only)
File Virtual Machine Help
Applications
Terminal
Mon 16:31
student@debian: /tmp
File Edit View Search Terminal Help
student@debian:/tmp$ cat /proc/filesystems | less
student@debian:/tmp$ dd if=/dev/zero of=junk bs=1M count=512
512+0 records in
512+0 records out
536870912 bytes (537 MB, 512 MiB) copied, 0.354312 s, 1.5 GB/s
student@debian:/tmp$ sudo /
Home
Trash
VMware Tools
THE LINUX FOUNDATION TRAINING
To release input, press Ctrl+Alt
coop@x7:~
coop@x7:/tmp
coop@x7:~
SimpleScreenRecorder
Debian-9 - VMware Workstation ...
11:00 AM


Activities
Applicat...
VMw...
Mon 26 Jun, 4:32 PM
Debian-9 - VMware Workstation 12 Player (Non-commercial use only)
File Virtual Machine Help
Applications
Terminal
Mon 16:32
student@debian: /tmp
File Edit View Search Terminal Help
meta-data=junk isize=512 agcount=4, agsize=32768 blks
= sectsz=512 attr=2, projid32bit=1
= crc=1 finobt=1, sparse=0, rmapbt=0, refl
ink=0
data = bsize=4096 blocks=131072, imaxpct=25
= sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0 ftype=1
log =internal log bsize=4096 blocks=855, version=2
= sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
student@debian:/tmp$ sudo mount junk /mnt
student@debian:/tmp$ df -h
Filesystem Size Used Avail Use% Mounted on
udev 1.9G 0 1.9G 0% /dev
tmpfs 381M 12M 370M 4% /run
/dev/sda1 18G 7.3G 9.1G 45% /
tmpfs 1.9G 0 1.9G 0% /dev/shm
tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs 1.9G 0 1.9G 0% /sys/fs/cgroup
tmpfs 381M 28K 381M 1% /run/user/117
tmpfs 381M 28K 381M 1% /run/user/1000
/dev/sr0 56M 56M 0 100% /media/cdrom0
/dev/loop0 509M 27M 483M 6% /mnt
student@debian:/tmp$
Home
Trash
VMware Tools
THE LINUX FOUNDATION TRAINING
To release input, press Ctrl+Alt
coop@x7:~
coop@x7:/tmp
coop@x7:~
SimpleScreenRecorder
Debian-9 - VMware Workstation ...
11:00 AM

Redundant Array of Independent Disks (RAID)

1:09 PM

- RAID (Redundant Array of Independent Disks) spreads I/O over multiple spindles, or disks; this can really increase performance in modern disk controller interfaces, such as SCSI which can perform the work in parallel efficiently
- RAID can be implemented either in software (it is a mature part of the Linux kernel) or in hardware
- If your hardware RAID is known to be of good quality it should be more efficient than using software RAID
- With the hardware implementation the operating system is actually not directly aware of using RAID; it is transparent
    - For example, three 512 GB hard drives (two for data, one for parity) configured with RAID-5 will just look like a single 1 TB disk

Features of RAID

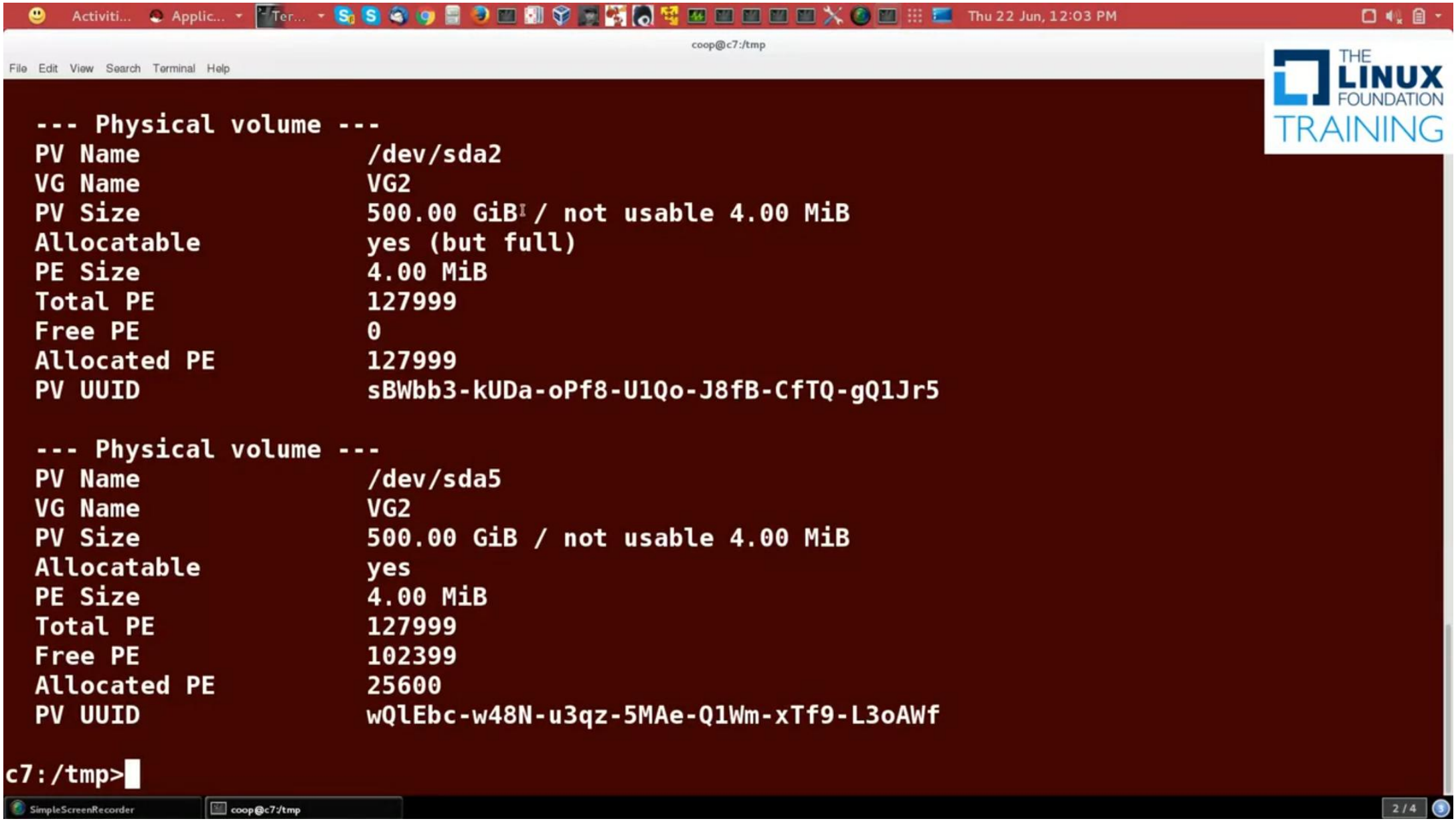
- Three essential features of RAID are:
  - **Mirroring**: writing the same data to more than one disk
  - **Striping**: splitting of data to more than one disk
  - **Parity**: extra data is stored to allow problem detection and repair, yielding fault tolerance
- Thus use of RAID can improve both performance and reliability
- There are a number of RAID specifications of increasing complexity and use; the most commonly used are levels 0, 1, and 5

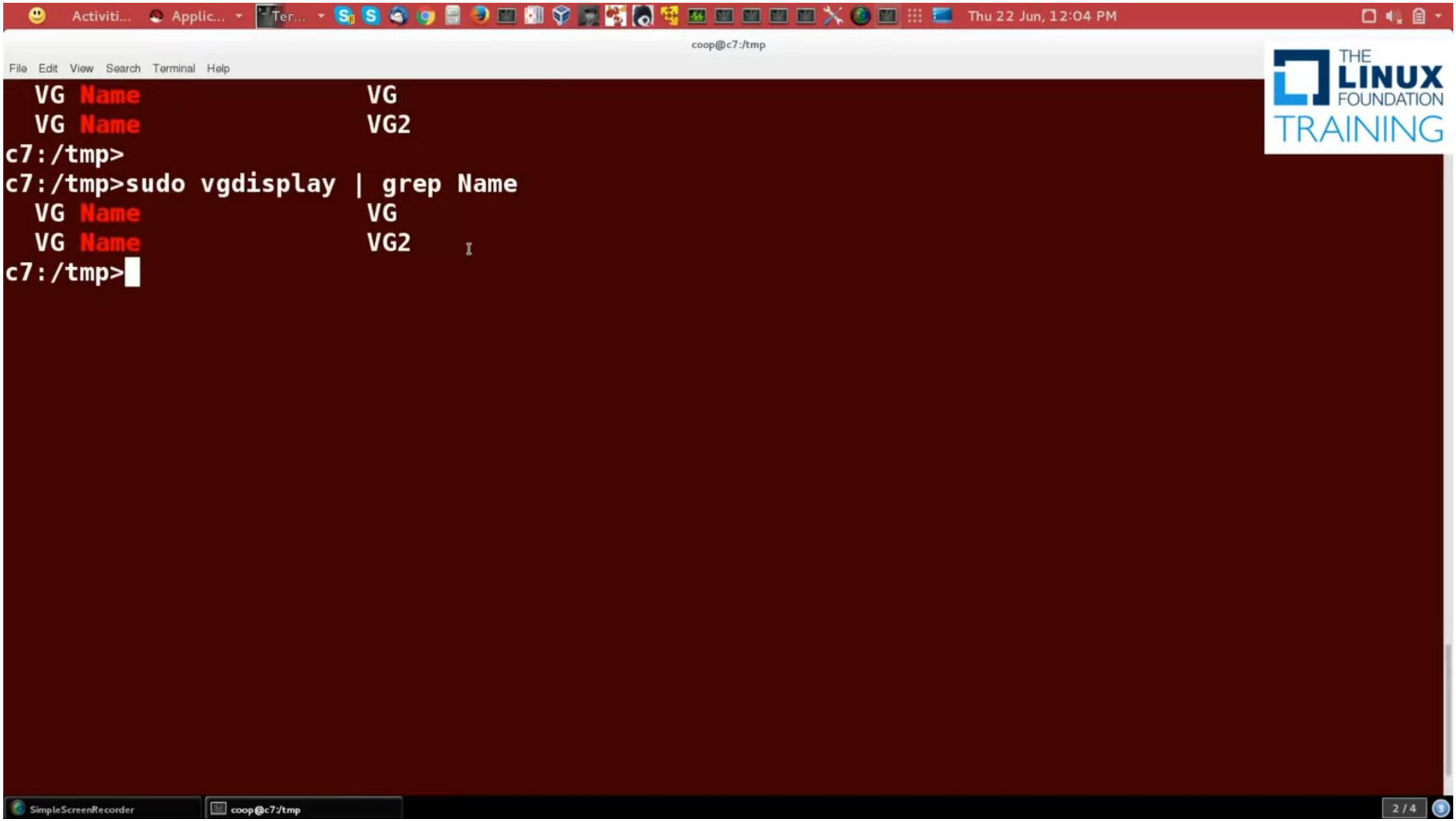
Logical Volume Management (LVM)

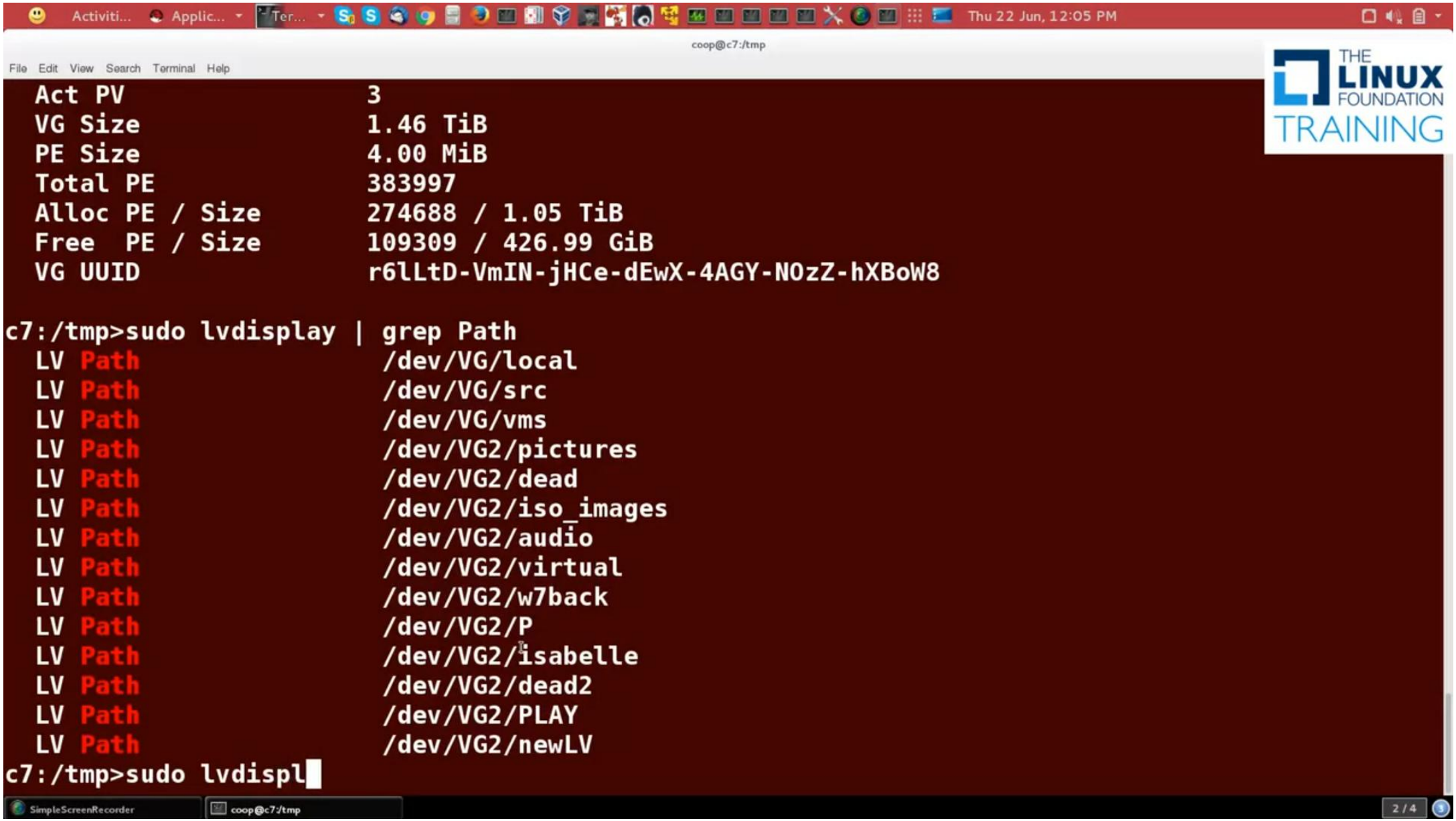
- Using **LVM** (**L**ogical **V**olume **M**anagement) breaks up one virtual partition into multiple chunks, which can be on different physical volumes
- There are many advantages to using LVM; in particular it becomes really easy to change the size of the logical partitions and filesystems, to add more space, rearrange things, etc.
- One or more physical disk partitions, or **physical volumes**, are grouped together into a **volume group**
- Then, the volume group is subdivided into logical volumes which appear to the system as disk partitions and are then formatted to contain mountable filesystems

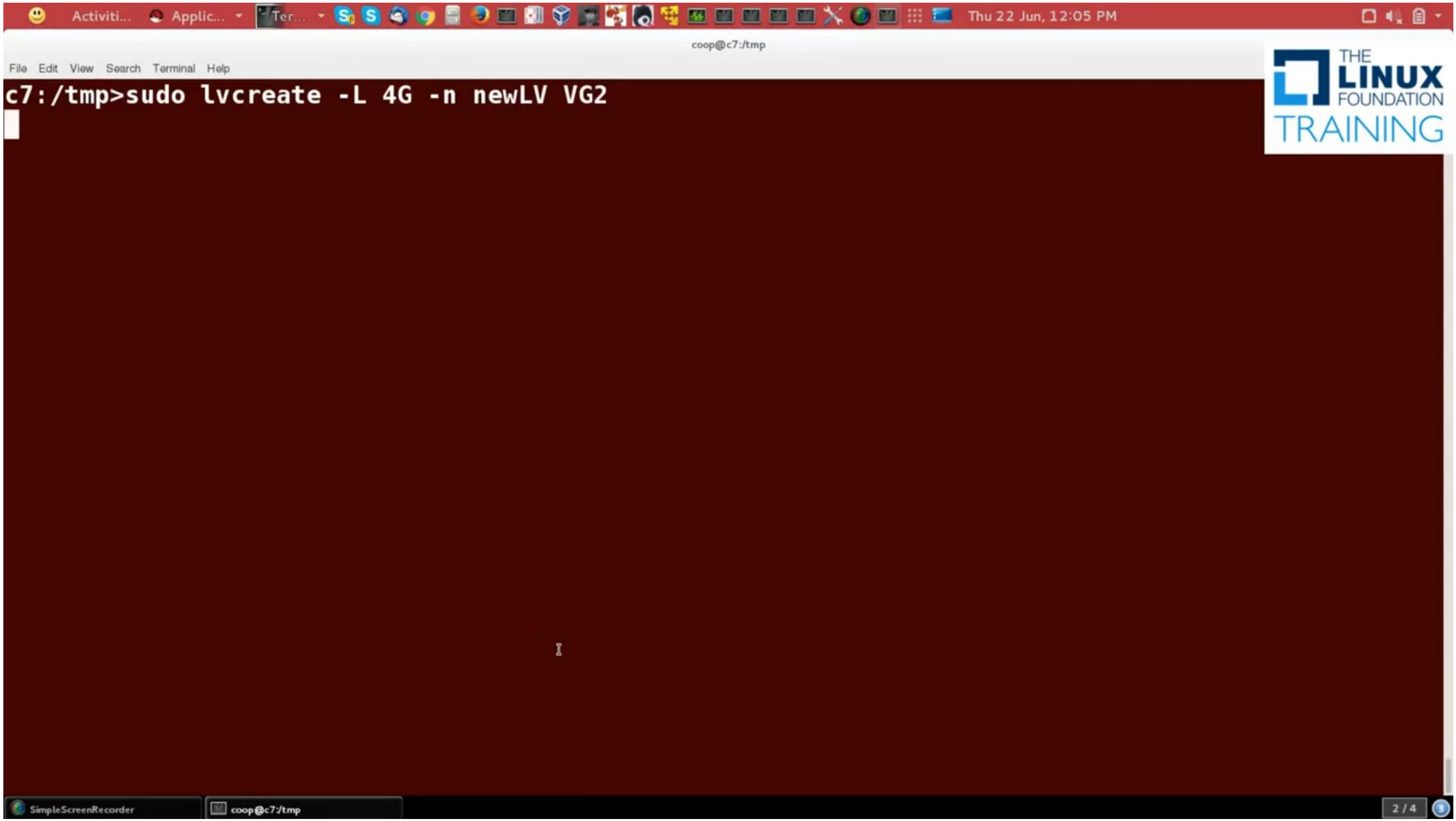
Logical Volume Management (Cont.)

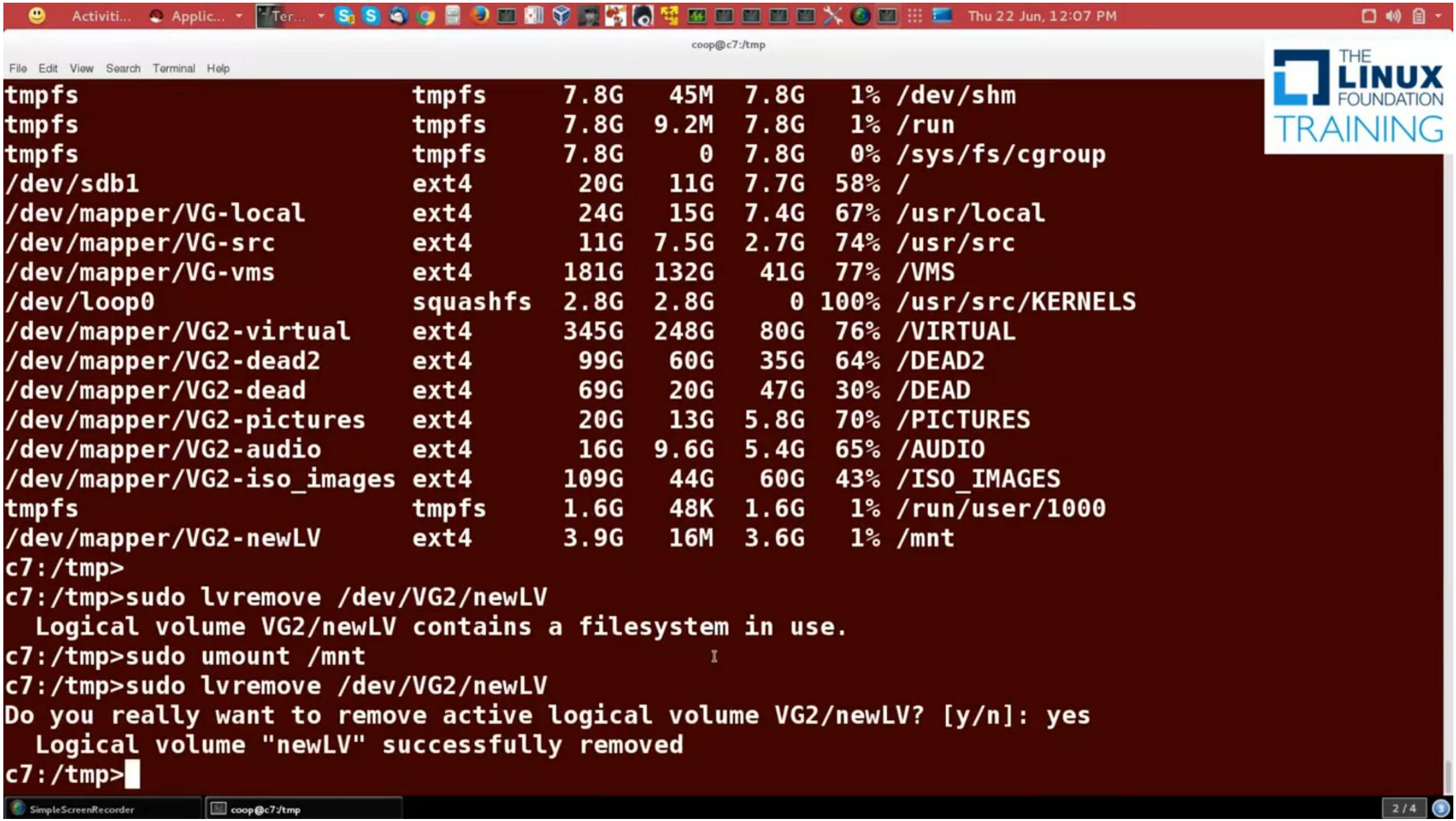
- There are a variety of command line utilities tasked to create, delete, resize, etc. physical and logical volumes; fortunately, these command line utilities are not hard to use and are quite flexible
- Performance changes do occur
- There is a definite additional cost that comes from the overhead of the LVM layer; however, even on non-RAID systems, if you use striping in the setup you can achieve some parallelization improvements
- Such striping does no good if it is within the same physical disk, however


Activiti... Applic... Ter... Thu 22 Jun, 12:03 PM
coop@c7:/tmp
File Edit View Search Terminal Help
--- Physical volume ---
PV Name /dev/sda2
VG Name VG2
PV Size 500.00 GiB / not usable 4.00 MiB
Allocatable yes (but full)
PE Size 4.00 MiB
Total PE 127999
Free PE 0
Allocated PE 127999
PV UUID sBWbb3-kUDa-oPf8-U1Qo-J8fB-CfTQ-gQ1Jr5
--- Physical volume ---
PV Name /dev/sda5
VG Name VG2
PV Size 500.00 GiB / not usable 4.00 MiB
Allocatable yes
PE Size 4.00 MiB
Total PE 127999
Free PE 102399
Allocated PE 25600
PV UUID wQlEbc-w48N-u3qz-5MAe-Q1Wm-xTf9-L3oAWf
c7:/tmp>
SimpleScreenRecorder
coop@c7:/tmp
2 / 4


Activiti...
Applic...
Ter...
Thu 22 Jun, 12:04 PM
coop@c7:/tmp
File Edit View Search Terminal Help
VG Name VG
VG Name VG2
c7:/tmp>
c7:/tmp>sudo vgdisplay | grep Name
VG Name VG
VG Name VG2
c7:/tmp>
THE LINUX FOUNDATION TRAINING
SimpleScreenRecorder
coop@c7:/tmp
2 / 4


Activiti... Applic... Ter... Thu 22 Jun, 12:05 PM
coop@c7:/tmp
File Edit View Search Terminal Help
Act PV 3
VG Size 1.46 TiB
PE Size 4.00 MiB
Total PE 383997
Alloc PE / Size 274688 / 1.05 TiB
Free PE / Size 109309 / 426.99 GiB
VG UUID r6lLtD-VmIN-jHCe-dEwX-4AGY-NOzZ-hXBoW8
c7:/tmp>sudo lvdisplay | grep Path
LV Path /dev/VG/local
LV Path /dev/VG/src
LV Path /dev/VG/vms
LV Path /dev/VG2/pictures
LV Path /dev/VG2/dead
LV Path /dev/VG2/iso_images
LV Path /dev/VG2/audio
LV Path /dev/VG2/virtual
LV Path /dev/VG2/w7back
LV Path /dev/VG2/P
LV Path /dev/VG2/isabelle
LV Path /dev/VG2/dead2
LV Path /dev/VG2/PLAY
LV Path /dev/VG2/newLV
c7:/tmp>sudo lvdispl
SimpleScreenRecorder
coop@c7:/tmp
2 / 4


Activiti...
Applic...
Ter...
Thu 22 Jun, 12:05 PM
coop@c7:/tmp
File Edit View Search Terminal Help
c7:/tmp>sudo lvcreate -L 4G -n newLV VG2
THE LINUX FOUNDATION TRAINING
SimpleScreenRecorder
coop@c7:/tmp
2 / 4


Activiti... Applic... Ter... Thu 22 Jun, 12:07 PM
coop@c7:/tmp
File Edit View Search Terminal Help
tmpfs tmpfs 7.8G 45M 7.8G 1% /dev/shm
tmpfs tmpfs 7.8G 9.2M 7.8G 1% /run
tmpfs tmpfs 7.8G 0 7.8G 0% /sys/fs/cgroup
/dev/sdb1 ext4 20G 11G 7.7G 58% /
/dev/mapper/VG-local ext4 24G 15G 7.4G 67% /usr/local
/dev/mapper/VG-src ext4 11G 7.5G 2.7G 74% /usr/src
/dev/mapper/VG-vms ext4 181G 132G 41G 77% /VMS
/dev/loop0 squashfs 2.8G 2.8G 0 100% /usr/src/KERNELS
/dev/mapper/VG2-virtual ext4 345G 248G 80G 76% /VIRTUAL
/dev/mapper/VG2-dead2 ext4 99G 60G 35G 64% /DEAD2
/dev/mapper/VG2-dead ext4 69G 20G 47G 30% /DEAD
/dev/mapper/VG2-pictures ext4 20G 13G 5.8G 70% /PICTURES
/dev/mapper/VG2-audio ext4 16G 9.6G 5.4G 65% /AUDIO
/dev/mapper/VG2-iso_images ext4 109G 44G 60G 43% /ISO_IMAGES
tmpfs tmpfs 1.6G 48K 1.6G 1% /run/user/1000
/dev/mapper/VG2-newLV ext4 3.9G 16M 3.6G 1% /mnt
c7:/tmp>
c7:/tmp>sudo lvremove /dev/VG2/newLV
Logical volume VG2/newLV contains a filesystem in use.
c7:/tmp>sudo umount /mnt
c7:/tmp>sudo lvremove /dev/VG2/newLV
Do you really want to remove active logical volume VG2/newLV? [y/n]: yes
Logical volume "newLV" successfully removed
c7:/tmp>
SimpleScreenRecorder
coop@c7:/tmp
2 / 4

gcc

- **gcc** is the **GNU C** compiler; its proper name is the **G**NU **C**ompiler **C**ollection
- It can be invoked as gcc or cc, and can compile programs written in C, C++, and Objective C
- **g++** is the C++ compiler; it can also be invoked as c++
- gcc works closely with the GNU libc, **glibc**, and the debugger, **gdb**
- Virtually every operating system you can think of has a version of gcc, and it can be used for cross-compilation on different architectures

gcc (Cont.)

- gcc also forms the back end for compiler front ends in Ada95 (package gcc-gnat), Fortran (package gcc-gfortran), and Pascal (package gcc-gpc); i.e. first there is a translation to the C language, and then a back end (silently) invokes gcc
- The gcc-java package supplies gcj which adds support for compiling Java programs and bytecode into native code, but is no longer available on some recent Linux distributions as it is considered obsolete

Compiling Stages

- Invoking gcc actually entails a number of different programs or stages, each of which has its own **man** page, and can be independently and directly invoked; these are:

Stage	Command	Default Input	Default Output	**-W** Switch
Preprocessing	cpp	`.c`	`.i`	`-Wp....`
Compilation	gcc	`.i`	`.s`	N/A
Assembly	as	`.s`	`.o`	`-Wa....`
Linking	ld	`.o`	`a.out`	`-Wl....`

- Depending on your Linux distribution, details about the gcc installation and defaults can be found in the **`/usr/lib/gcc`**, **`/usr/lib64/gcc`** and/or **`/usr/libexec/gcc`** directories

LLVM

- The **LLVM** project provides an alternative C/C++ compiler called clang which is rapidly becoming a real alternative to gcc. It is already used exclusively by projects such as FreeBSD, and Mandriva Linux. Several companies are moving over to using clang to replace their current toolchains with some companies like Google and Apple moving 100% of their software development over to clang.
- At this time all IOS (iPhone/iPad) and all Android devices (phones and tablets) are now running clang compiled code.
- You can learn all about LLVM and clang at: https://llvm.org/.
- The ClangBuiltLinux project has continued the work of the prior LLVMLinux project upstreaming patches to the Linux kernel which enable it to be compiled with the clang compiler. Full details can be found at: https://clangbuiltlinux.github.io/.

Intel

- **Intel** has a mature set of compilers
- Evaluation copies can be downloaded for free, and a free non-commercial license can be obtained for learning purposes
- The Intel C compiler works well for compiling applications under Linux; it can be used to compile the kernel but it is not a trivial exercise and it is doubtful anyone is using it for this purpose in a production environment


Press Esc to exit full screen
Debugging with gdb
THE LINUX FOUNDATION

gdb

- **gdb** is the GNU debugger
- Upon launch, after processing all command line arguments and options, it loads commands from the file **`.gdbinit`** in the current working directory (if it exists)
- gdb allows you to step through C and C++ programs, setting breakpoints, displaying variables, etc. (actually it will work with programs in native Fortran and other languages that use gcc as a back end as well)
- In addition, gdb can properly debug multi-threaded programs
- Programs have to be compiled with the **`-g`** option for symbol and line number information to be available to gdb
- Note, however, you can still use gdb to get some information even if this has not been done; for instance, the **where** command often tells you exactly where the program bombed

Why Use Package Management?

- Originally, most Linux distributions were simply collections of **tarballs**, of either binaries or sources which required compilation; some distributions (**Slackware**) still work this way
- However, this method has many disadvantages:
  - Removing all files from a package can be difficult
  - One might accidentally delete a package that other packages need, or install a package that will not work because it needs other packages that have not been installed
  - A developer may lose track of what exact sources were used to build a particular binary package
  - Updates and upgrades can be very difficult, for a number of reasons:
    - Files which are no longer needed may remain on the system
    - In place upgrades done while the system is running may cause difficulties, even system crashes
    - The order of upgrading software packages may be important but not be explicitly considered
    - Software groups that require simultaneous updating may conflict with each other

RPM and APT

- Several systems have been developed in the attempt to improve on this process, these include:
  - **RPM** (Red Hat Package Manager): originally only Red Hat used RPM, but many other Linux distributors have used RPM, such as Fedora, and particularly SUSE and its derivatives, such as OpenSUSE
  - **APT** (Advanced Packaging Tool): produces deb packages; originally developed by Debian it is the foundation of Ubuntu and other Debian-derived distributions such as Linux Mint
- **alien** is a tool for converting back and forth between rpm and deb packages, or installing one kind of package on a system set up for the other; however, alien is not available on all Linux distributions

Packaging System Benefits for Developers

- Repeatable builds
- Generation of dependency data, such as what other packages are needed by a given package, and/or what other packages (or class of packages) may need a given package
- Inclusion of the pristine sources, which aids in configuration control and establishing a clear history of revision
- Full explanation of any patches that have been made to the upstream sources, together with instructions for how to proceed in building the package as well as in installing it

Packaging System Benefits for Developers

- Repeatable builds
- Generation of dependency data, such as what other packages are needed by a given package, and/or what other packages (or class of packages) may need a given package
- Inclusion of the pristine sources, which aids in configuration control and establishing a clear history of revision
- Full explanation of any patches that have been made to the upstream sources, together with instructions for how to proceed in building the package as well as in installing it

Packaging System Benefits for System Administrators and End Users

- Integrity of the installation can be verified in a uniform and rapid fashion
- Simple installation and removal methods
- Package updates and upgrades automatically preserve customized and modified configuration files
- Error-checking on install and remove (erase) ensures needed resources are not obliterated
- Users and administrators can do queries on matters including identifying what files are part of a package, or the inverse process of asking what package (if any) a given file is part of

Maintenance

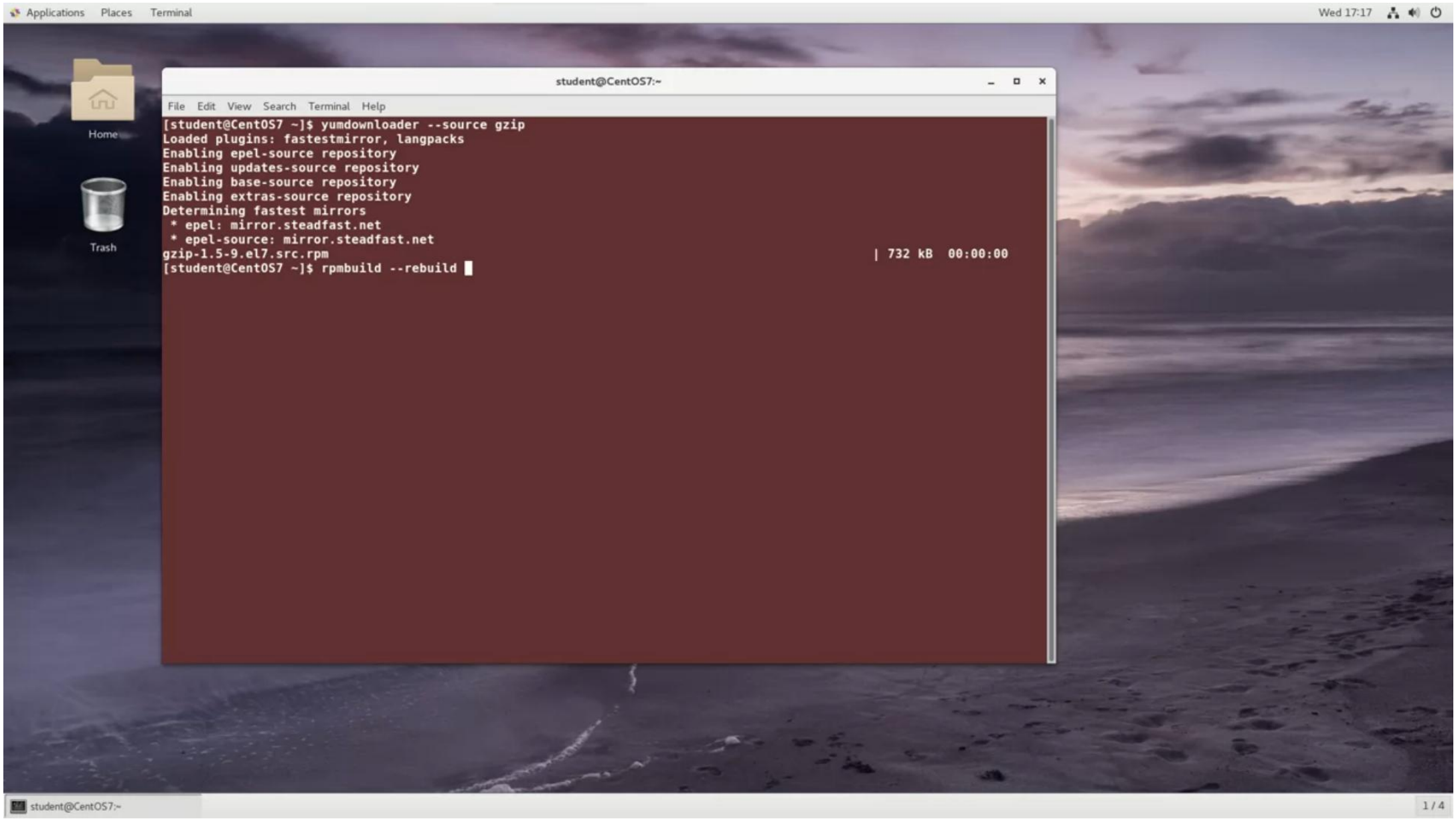
- Maintenance of software by using packaging systems is an essential task of any Linux distribution
- Every other function that must be accomplished requires coherent packaging, including:
  - Keeping a healthy bi-directional connection to the upstream developers, both incorporating patches from upstream as well as sending patches to the upstream maintainers
  - Establishing and deploying clear policies for dealing with configuration files, etc., as packages are updated
  - Updating and upgrading packages in timely fashion, both to fix outstanding bugs and security holes, and to incorporate new features
  - Most importantly, perhaps, ensuring proper package dependencies, including control the order of installation and removal, requiring various packages to be dealt with as one group, etc.
  - Maintaining (multiple) repositories, ensuring that they are secure and authoritative, as well as complete

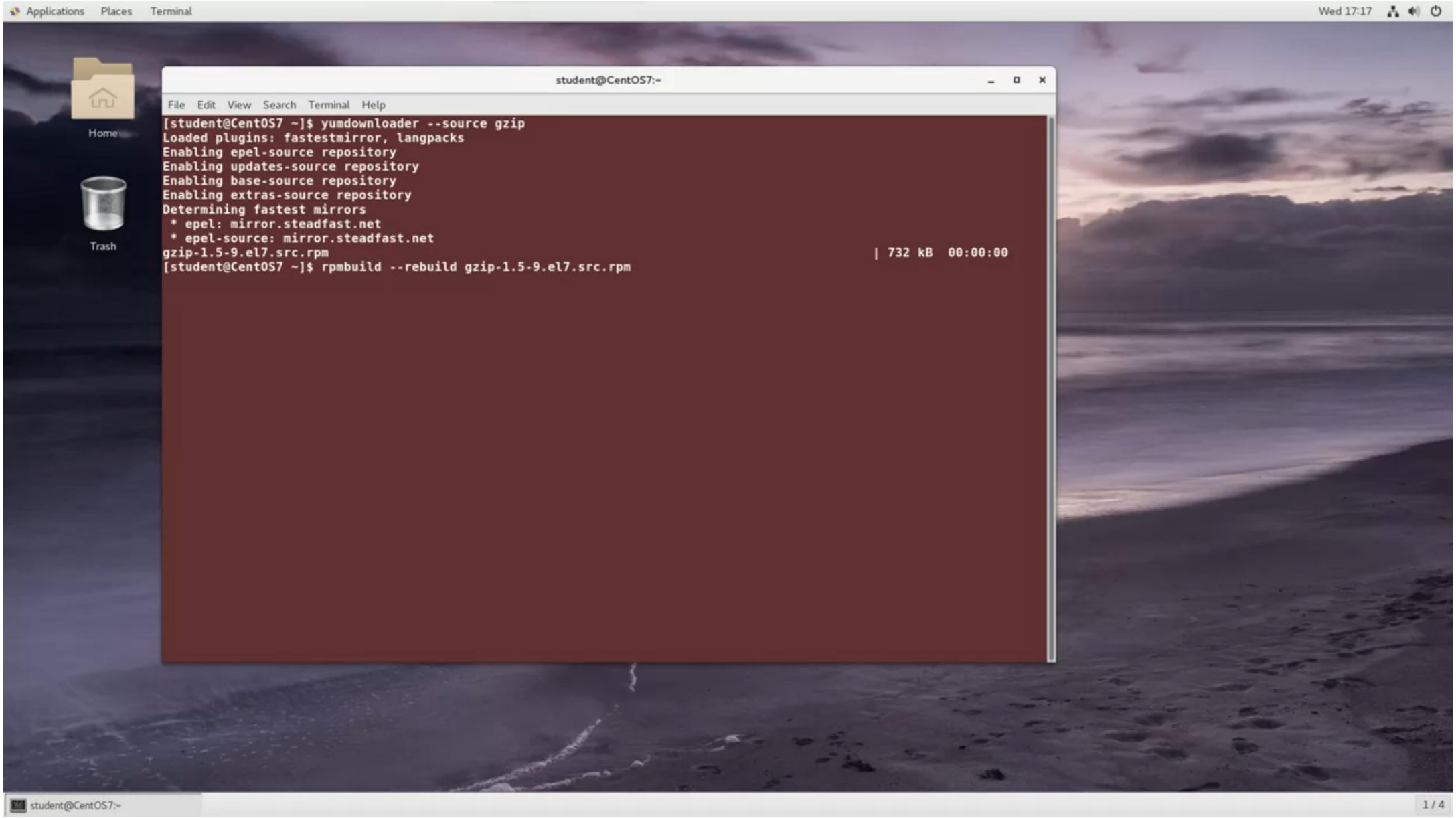
Maintenance (Cont.)

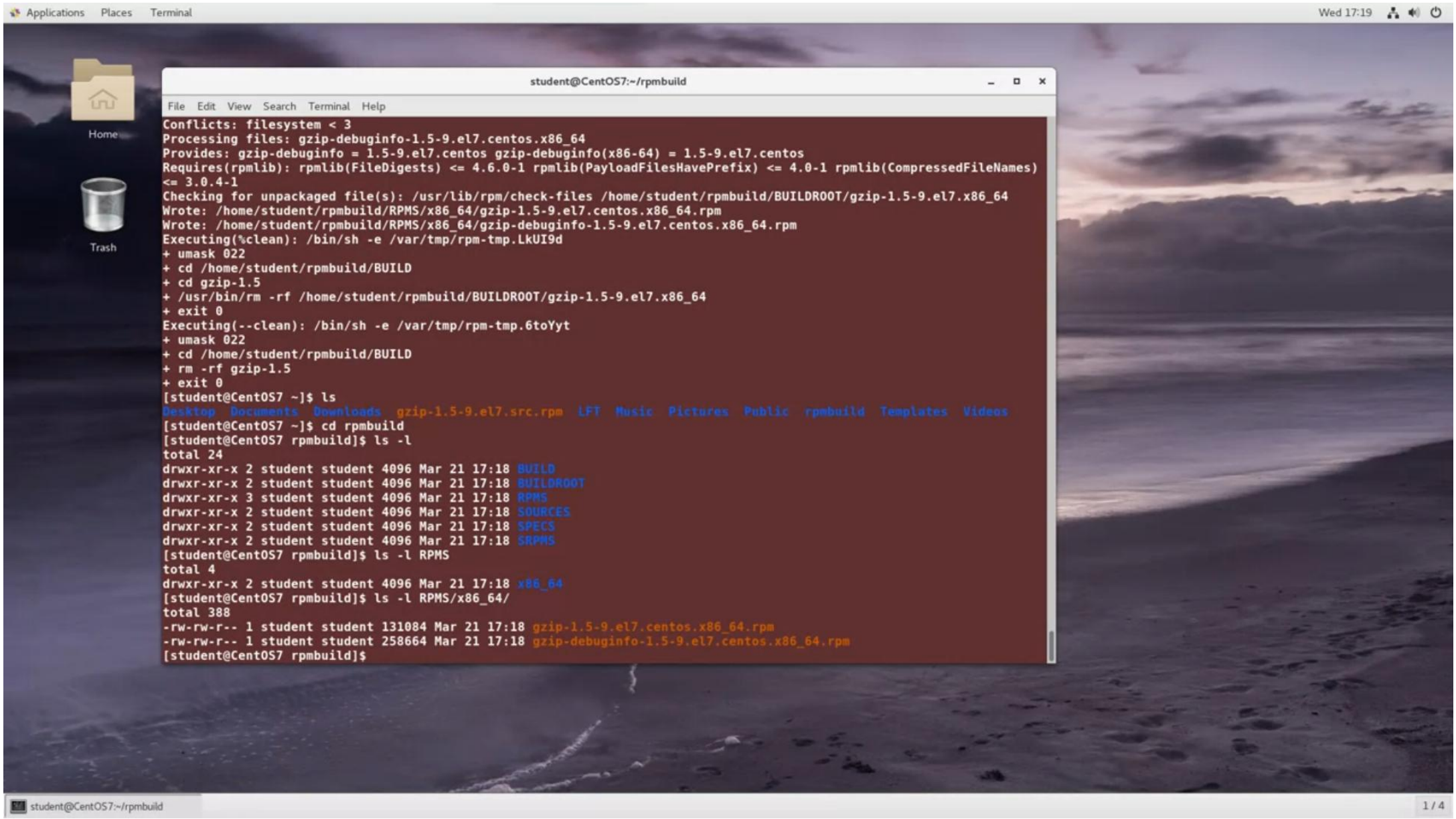
- Linux distributions include software from many sources
- Maintaining coherence, ensuring proper licensing compliance, etc., is actually a very difficult task.
- The package maintainer may be an employee of the distribution (especially for commercial distributions such as Red Hat or Ubuntu) or a "volunteer" for other distributions such as Debian or Fedora
- In either case, the maintainer has to think long-term, how to really do a good job on the package building so that updates, upgrades, use of different systems, can be done rather cleanly with time

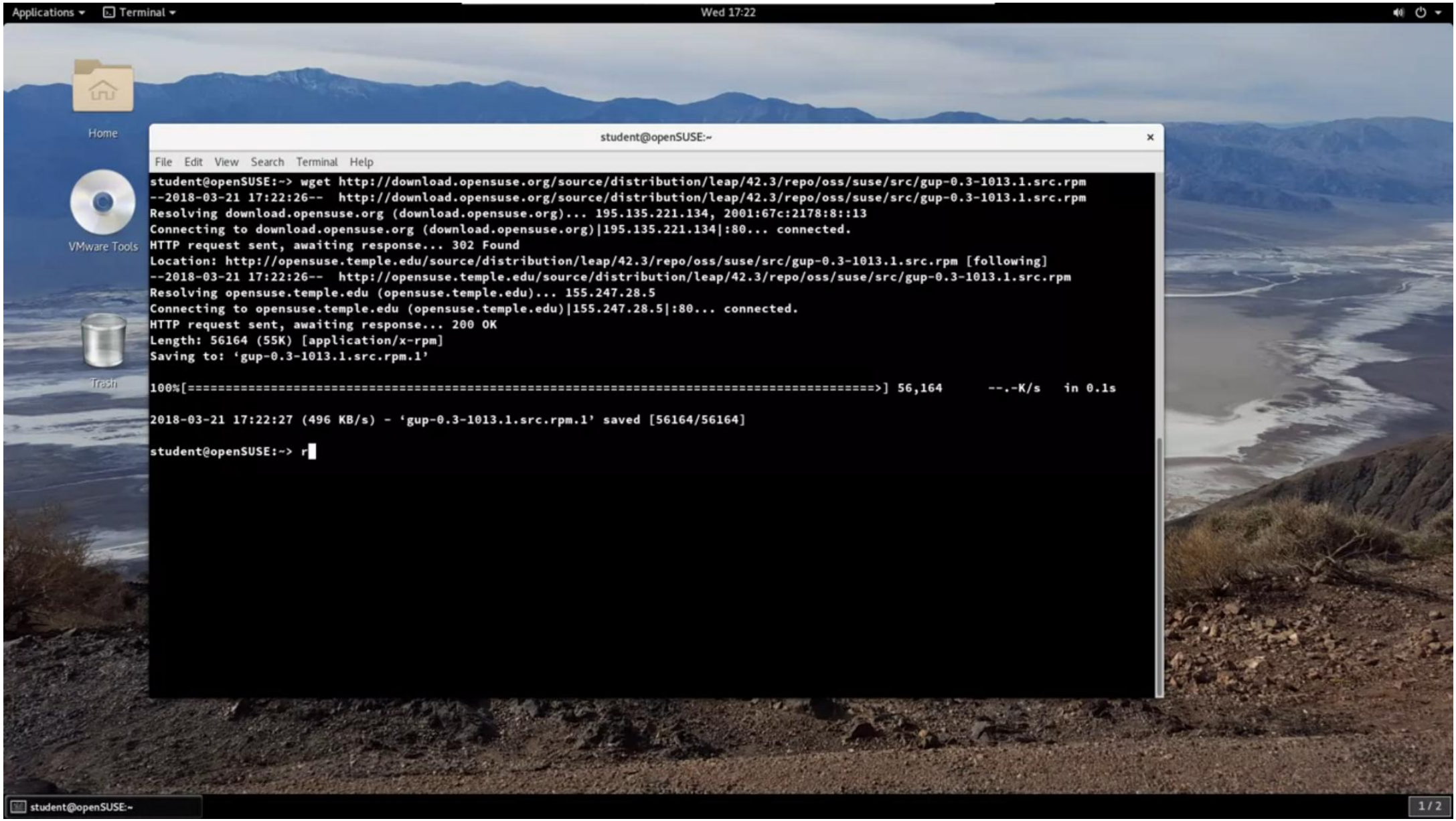
Maintenance (Cont.)

- Linux distributions include software from many sources
- Maintaining coherence, ensuring proper licensing compliance, etc., is actually a very difficult task.
- The package maintainer may be an employee of the distribution (especially for commercial distributions such as Red Hat or Ubuntu) or a "volunteer" for other distributions such as Debian or Fedora
- In either case, the maintainer has to think long-term, how to really do a good job on the package building so that updates, upgrades, use of different systems, can be done rather cleanly with time


Applications
Places
Terminal
Wed 17:17
Home
Trash
student@CentOS7:~
File Edit View Search Terminal Help
[student@CentOS7 ~]$ yumdownloader --source gzip
Loaded plugins: fastestmirror, langpacks
Enabling epel-source repository
Enabling updates-source repository
Enabling base-source repository
Enabling extras-source repository
Determining fastest mirrors
* epel: mirror.steadfast.net
* epel-source: mirror.steadfast.net
gzip-1.5-9.el7.src.rpm | 732 kB 00:00:00
[student@CentOS7 ~]$ rpmbuild --rebuild
student@CentOS7:~
1 / 4


Applications
Places
Terminal
Wed 17:17
Home
Trash
student@CentOS7:~
File Edit View Search Terminal Help
[student@CentOS7 ~]$ yumdownloader --source gzip
Loaded plugins: fastestmirror, langpacks
Enabling epel-source repository
Enabling updates-source repository
Enabling base-source repository
Enabling extras-source repository
Determining fastest mirrors
* epel: mirror.steadfast.net
* epel-source: mirror.steadfast.net
gzip-1.5-9.el7.src.rpm | 732 kB 00:00:00
[student@CentOS7 ~]$ rpmbuild --rebuild gzip-1.5-9.el7.src.rpm
student@CentOS7:~
1 / 4


Applications
Places
Terminal
Wed 17:19
Home
Trash
student@CentOS7:~/rpmbuild
File Edit View Search Terminal Help
Conflicts: filesystem < 3
Processing files: gzip-debuginfo-1.5-9.el7.centos.x86_64
Provides: gzip-debuginfo = 1.5-9.el7.centos gzip-debuginfo(x86-64) = 1.5-9.el7.centos
Requires(rpmlib): rpmlib(FileDigests) <= 4.6.0-1 rpmlib(PayloadFilesHavePrefix) <= 4.0-1 rpmlib(CompressedFileNames) <= 3.0.4-1
Checking for unpackaged file(s): /usr/lib/rpm/check-files /home/student/rpmbuild/BUILDROOT/gzip-1.5-9.el7.x86_64
Wrote: /home/student/rpmbuild/RPMS/x86_64/gzip-1.5-9.el7.centos.x86_64.rpm
Wrote: /home/student/rpmbuild/RPMS/x86_64/gzip-debuginfo-1.5-9.el7.centos.x86_64.rpm
Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.LkUI9d
+ umask 022
+ cd /home/student/rpmbuild/BUILD
+ cd gzip-1.5
+ /usr/bin/rm -rf /home/student/rpmbuild/BUILDROOT/gzip-1.5-9.el7.x86_64
+ exit 0
Executing(--clean): /bin/sh -e /var/tmp/rpm-tmp.6toYyt
+ umask 022
+ cd /home/student/rpmbuild/BUILD
+ rm -rf gzip-1.5
+ exit 0
[student@CentOS7 ~]$ ls
Desktop Documents Downloads gzip-1.5-9.el7.src.rpm LFT Music Pictures Public rpmbuild Templates Videos
[student@CentOS7 ~]$ cd rpmbuild
[student@CentOS7 rpmbuild]$ ls -l
total 24
drwxr-xr-x 2 student student 4096 Mar 21 17:18 BUILD
drwxr-xr-x 2 student student 4096 Mar 21 17:18 BUILDROOT
drwxr-xr-x 3 student student 4096 Mar 21 17:18 RPMS
drwxr-xr-x 2 student student 4096 Mar 21 17:18 SOURCES
drwxr-xr-x 2 student student 4096 Mar 21 17:18 SPECS
drwxr-xr-x 2 student student 4096 Mar 21 17:18 SRPMS
[student@CentOS7 rpmbuild]$ ls -l RPMS
total 4
drwxr-xr-x 2 student student 4096 Mar 21 17:18 x86_64
[student@CentOS7 rpmbuild]$ ls -l RPMS/x86_64/
total 388
-rw-rw-r-- 1 student student 131084 Mar 21 17:18 gzip-1.5-9.el7.centos.x86_64.rpm
-rw-rw-r-- 1 student student 258664 Mar 21 17:18 gzip-debuginfo-1.5-9.el7.centos.x86_64.rpm
[student@CentOS7 rpmbuild]$
student@CentOS7:~/rpmbuild
1 / 4


Applications
Terminal
Wed 17:22
Home
VMware Tools
Trash
student@openSUSE:~
File Edit View Search Terminal Help
student@openSUSE:~> wget http://download.opensuse.org/source/distribution/leap/42.3/repo/oss/suse/src/gup-0.3-1013.1.src.rpm
--2018-03-21 17:22:26-- http://download.opensuse.org/source/distribution/leap/42.3/repo/oss/suse/src/gup-0.3-1013.1.src.rpm
Resolving download.opensuse.org (download.opensuse.org)... 195.135.221.134, 2001:67c:2178:8::13
Connecting to download.opensuse.org (download.opensuse.org)|195.135.221.134|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://opensuse.temple.edu/source/distribution/leap/42.3/repo/oss/suse/src/gup-0.3-1013.1.src.rpm [following]
--2018-03-21 17:22:26-- http://opensuse.temple.edu/source/distribution/leap/42.3/repo/oss/suse/src/gup-0.3-1013.1.src.rpm
Resolving opensuse.temple.edu (opensuse.temple.edu)... 155.247.28.5
Connecting to opensuse.temple.edu (opensuse.temple.edu)|155.247.28.5|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 56164 (55K) [application/x-rpm]
Saving to: 'gup-0.3-1013.1.src.rpm.1'
100%[==========>] 56,164 --.-K/s in 0.1s
2018-03-21 17:22:27 (496 KB/s) - 'gup-0.3-1013.1.src.rpm.1' saved [56164/56164]
student@openSUSE:~> r
student@openSUSE:~
1 / 2