

**System Commands**  
**Professor. Gandham Phanikumar**  
**Metallurgical and Materials Engineering**  
**Indian Institute of Technology, Madras**  
**Some Command Line Utilities**

(Refer Slide Time: 0:14)

## Important Utilities



Tools that can augment your productivity



## Utilities



- **find** – locating files and processing them
- **tar**, **gzip** etc – packaging collections of files
- **make** – conditional actions



So, here are some important utilities that could help you in improving your productivity. And these are quite extensive. So, we will actually have some demo where we can see more details find is one utility, which helps you in locating files and processing those files which are satisfying the conditions that you gave for the searching. Tar and the gzip are tools which are useful for packaging collections of files.

And make is a utility to have conditional actions performed. Actually, make is meant for conditional compilation of source codes. But it is a very generic tool. So, it need not be limited to compilation of source codes like C or C plus plus or Fortran, but actually to do any other action also. So, you can think of it as yet another way of writing a script that would work on the command prompt.

(Refer Slide Time: 1:16)

**find**

`find [pathnames] [conditions]`

-name	pattern to match filenames
-type	File type code eg., <b>c</b> for character file, <b>d</b> for directory, <b>l</b> for symbolic link etc.,
-atime	Files accessed <b>+n</b> (more than n), <b>-n</b> (less than n) days ago
-ctime	File changed <b>+n</b> (more than n), <b>-n</b> (less than n) days ago
-regex	Regular expression for pattern of filenames Combine with -regextype posix-basic, posix-egrep etc.,
-exec	Command to run using <b>{}</b> as place holder for filename
-print	Print the full path name of matching files

The syntax of find command is very simple after the find, you need to give a list of path names followed by the conditions. Typical conditions are listed here. So, minus name, followed by a pattern to match the filenames. So, the pattern can be like what the bash would expect. So, with a star or wild characters that are interpreted by bash, then minus type is a condition where you could actually tell what type of a file are you looking for?

And the code is C for character file, d directory, L for symbolic link et cetera. Minus a time is a condition to check. How many days back was the file accessed? So, was it like plus 30 that is more than 30 days back or minus 30 less than 30 days back? So, this is useful in finding out what are the recently accessed files, or what are the one of the oldest files touched in the system.


Minus c time is very similar to minus a 10 except that the files that are being searched are looked at when they were modified, rather than just merely accessed. Minus regex is also a condition where you could give a regular expression by also mentioning what type of a regular expression are you using posix basic or posix egrep et cetera. So, that the filenames using the complete path name can be looked at, in a very powerful manner.


Minus exec is an extremely useful condition where you can follow it up with a command and in the command, you could use the two braces, as the placeholder for the file name, which has been searched by the find command. Minus print is a condition where there is no value that is supplied after that. And what it does is just merely to print the file name, which has matched the pattern that you were searching for.

(Refer Slide Time: 3:24)

## file packaging

- Deep file hierarchies
- Large number of tiny files
- **tar** : collect a file hierarchy into a single file
- **gzip** : compress a file
- **Applications**: backup, file sharing, reduce disc utilization





IIT Madras  
BSc Degree

We need to package files for multiple reasons. So, if you have many sub directories and directories under them et cetera. So, actions like copying the directory across or looking through those directories for some specific file et cetera would be very slow. So, very deep file hierarchies are not recommended. And in situations where such hierarchies are present, but you do not need them immediately, it is a good idea to convert the entire file hierarchy into a single file, and then store it as a package.

So, tar file tar is a command paper archive, which can be used to convert a file hierarchy into a single tar file. Sometimes when you have a very large number of tiny files, each file would occupy the minimum block size that is recommended or required for the file system. So, there is a wastage of space. So, in such situations, also simply taring a whole bunch of tiny files would actually save some disk space for you.

And gzip is a compressing the utility to reduce the file occupancy of the tar file or any other large file in your file system. And where do we use these tools? We have the applications of these tools like tar and gzip in situations where we take backup, when we want to share a file across the Internet or just simply to reduce the disk utilization of the files in our directory.

(Refer Slide Time: 5:04)

## Possibilities



- `tar`, `zip`
- `compress` (`ncompress`), `gzip` (`ncompress`), `bzip2` (`bzip2`), `xz` (`xz-utils`), `7z` (`p7zip-full`)
- Tarballs like `bundle.tgz` for package + compress
- Time & memory required to shrink / expand versus size ratio
- Portability
- Unique names using timestamp, process ID etc., for backup tarballs



Now there are multiple possibilities that are available for you to just simply club all files into a single file. And the tar is one of the options a parquet but zip is also another option, zip would actually also compress the file while it is actually pulling them together, and if you are having a file, a tape archive file or any other file that is large in size, you could reduce its size by many utilities, compress gzip, dzip to xz with 7z are the tools that are useful in reducing the size of a file.

Particularly if you have ascii files or plain text files, they can be compressed to a very good ratio almost like 1 is to 10 if the file contains many repeating patterns. Now in the brackets, I have given the package names, which would let you have those tools installed on your system. And very often, when you have a huge set of files, bundled together as a tar file, and then after that you gzip to reduce the size, and then give it to someone the other user when they unzip it, the tar file would be still very big for them before they expand it to the respective file hierarchy.

So, you can actually have a possibility of zipping the files while adding into the tar file. And so dot tz said file format that is combining tar as well as gzip together is also a very good option. And which of the utilities, 7z or dzip2 or gzip, which of these utilities should I use. So, the decision is made by looking at how much time does it take to compress the file and how much of memory is required to do that particular compression, and also how much time and memory are required to expand it.

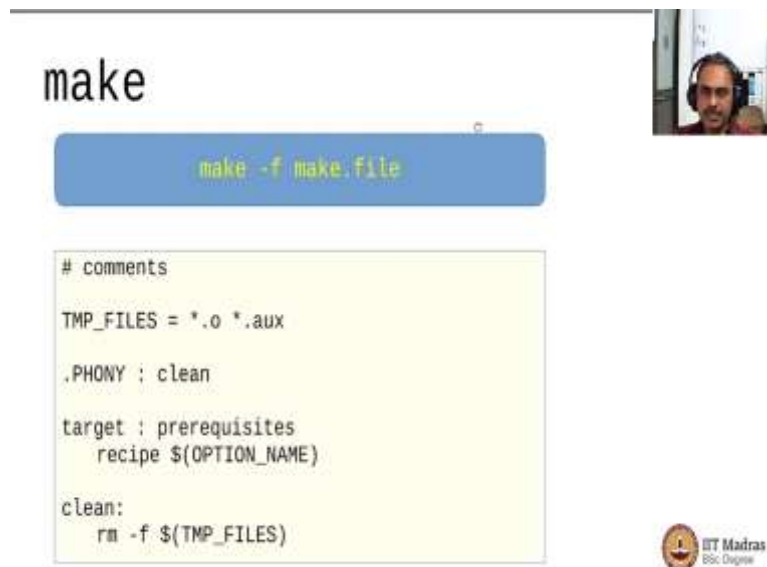
So, very often compress does the job very fast, but the compression ratio is not great. gzip is a very sweet intermediary. Bzip2 do a very good job of compressing by giving a very good

ratio, but then it would occupy more space and take little more time to do the job. And there are also other utilities to do the same thing of bundling the files together, like rar format, which is popular in Windows ecosystem.

But it is a good idea to use those tools which are having some portability. So, the tools I am mentioning here are open source tools available for all platforms. So, if you use these tools for the bundling of files and compressing them, then you can uncompress and unbundle those files quite easily because the tools are open source and freely available.

And it is also useful to learn how to stamp a date and time as well as the process ID along with a tar file for backups. so that unique file names are available when you are backing up some important pieces of information from your system at regular intervals.

(Refer Slide Time: 8:18)



make

```
make -f make.file
```

```
# comments
TMP_FILES = *.o *.aux
.PHONY : clean
target : prerequisites
    recipe $(OPTION_NAME)
clean:
    rm -f $(TMP_FILES)
```

IIT Madras  
BSc Degree

Make is a utility originally meant for compilation and you can actually create a script and pass it on as an option to the make using the minus f option. And it would actually do certain actions for you apart from compiling the source codes it can also do some actions for you. So, the most important syntax that you must know is that there is a target colon prerequisites and below that to get the indent the recipe of action that has to be taken.

So, that is the most important piece of information that has to be in the make file. And here is an example script which we will actually try out on the command line as a demonstration. Make files can be used to do conditional running of a script which is slightly powerful feature compared to just a plain bash script. So, let us explore these on the command prompt once.

(Refer Slide Time: 9:17)

```

FIND(1)                                General Commands Manual                                FIND(1)

NAME
    find - search for files in a directory hierarchy

SYNOPSIS
    find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]

DESCRIPTION
    This manual page documents the GNU version of find. GNU find searches the directory
    tree rooted at each given starting-point by evaluating the given expression
    from left to right, according to the rules of precedence (see section OPERATORS),
    until the outcome is known (the left hand side is false for and operations, true
    for or, at which point find moves on to the next file name. If no starting-
    point is specified, '.' is assumed.

    If you are using find in an environment where security is important (for example
    if you are using it to search directories that are writable by other users), you
    should read the 'Security Considerations' chapter of the findutils documentation,
    which is called Finding Files and comes with findutils. That document also in-
    cludes a lot more detail and discussion than this manual page, so you may find it
    a more useful source of information.

OPTIONS
    The -H, -L and -P options control the treatment of sybolic links. Command-line
    arguments following these are taken to be names of files or directories to be ex-
    amined, up to the first argument that begins with '-', or the argument '{' or
    Manual page find(1) line 1 (press h for help or q to quit)

```

```

find /tmp -name core -type f -print | xargs /bin/rm -f

Find files named core in or below the directory /tmp and delete them. Note that
this will work incorrectly if there are any filenames containing newlines, single
or double quotes, or spaces.

find /tmp -name core -type f -print0 | xargs -0 /bin/rm -f

Find files named core in or below the directory /tmp and delete them, processing
filenames in such a way that file or directory names containing single or double
quotes, spaces or newlines are correctly handled. The -name test comes before
the -type test in order to avoid having to call stat(2) on every file.

find . -type f -exec file '{}' \;

Runs 'file' on every file in or below the current directory. Notice that the
braces are enclosed in single quote marks to protect them from interpretation as
shell script punctuation. The semicolon is similarly protected by the use of a
backslash, though single quotes could have been used in that case also.

find / \( -perm -4000 -fprintf /root/suid.txt '%u %u %u\n' \) , \(
  \( -size +100M -fprintf /root/big.txt '%-10s %u\n' \) \)

Traverse the filesystem just once, listing setuid files and directories into
/root/suid.txt and large files into /root/big.txt.

find $HOME -etime 5
Manual page find(1) line 1243 (press h for help or q to quit)

```

```

gshanic@kali:~$ man find
Now running command: man find
gshanic@kali:~$ find $HOME

```



```
/home/gphani/.cache/google-chrome/System Profile/Code Cache
/home/gphani/.cache/google-chrome/System Profile/Code Cache/wasm
/home/gphani/.cache/google-chrome/System Profile/Code Cache/wasm/index-dir
/home/gphani/.cache/google-chrome/System Profile/Code Cache/wasm/index-dir/the-real-index
/home/gphani/.cache/google-chrome/System Profile/Code Cache/wasm/index
/home/gphani/.cache/google-chrome/System Profile/Code Cache/js
/home/gphani/.cache/google-chrome/System Profile/Code Cache/js/index-dir
/home/gphani/.cache/google-chrome/System Profile/Code Cache/js/index-dir/the-real-index
/home/gphani/.cache/google-chrome/System Profile/Code Cache/js/index
/home/gphani/.cache/google-chrome/Guest Profile/Cache
/home/gphani/.cache/google-chrome/Guest Profile/Cache/Cache_Data
/home/gphani/.cache/google-chrome/Guest Profile/Cache/Cache_Data/index-dir
/home/gphani/.cache/google-chrome/Guest Profile/Cache/Cache_Data/index-dir/the-real-index
/home/gphani/.cache/google-chrome/Guest Profile/Cache/Cache_Data/index
/home/gphani/.cache/google-chrome/Guest Profile/Code Cache
/home/gphani/.cache/google-chrome/Guest Profile/Code Cache/wasm
/home/gphani/.cache/google-chrome/Guest Profile/Code Cache/wasm/index-dir
/home/gphani/.cache/google-chrome/Guest Profile/Code Cache/wasm/index-dir/the-real-index
/home/gphani/.cache/google-chrome/Guest Profile/Code Cache/wasm/index
/home/gphani/.cache/google-chrome/Guest Profile/Code Cache/js
/home/gphani/.cache/google-chrome/Guest Profile/Code Cache/js/index-dir
/home/gphani/.cache/google-chrome/Guest Profile/Code Cache/js/index-dir/the-real-index
/home/gphani/.cache/google-chrome/Guest Profile/Code Cache/js/index
/home/gphani/.cache/google-chrome/Profile 4
/home/gphani/.cache/google-chrome/Profile 4/Cache
/home/gphani/.cache/google-chrome/Profile 4/Cache/Cache_Data
```

```
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sll.d
do/2718609325eaxeprn.sqlite
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sll.d
do/2718609325eaxeprn.files
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sll.d
s
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sll.d/l
s/data.sqlite
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sll.d/l
s/usage
/home/gphani/.mozilla/firefox/lo8ttfce.default-
/home/gphani/.mozilla/firefox/lo8ttfce.default/times.json
/home/gphani/.mozilla/firefox/profiles.ini
/home/gphani/.mozilla/firefox/Pending Pings
/home/gphani/.mozilla/firefox/installs.ini
/home/gphani/.mozilla/systemextensionsdev
/home/gphani/.ssh
/home/gphani/.ssh/known hosts
/home/gphani/.bashrc
/home/gphani/.vidyo
/home/gphani/.vidyo/VidyoDesktop
/home/gphani/.vidyo/VidyoDesktop/ca-certificates.crt
/home/gphani/.personal
/home/gphani/.personal/sign.png
/home/gphani/.personal/seal.jpg
/home/gphani/.bash history
gphani@gphani:~$
```

So, now let us look at the find command. So, the find help man page available. So, it is quite long. And there are some examples also given here. So, you could look at them. So, the simplest way of running is find and then the path which is the home directory and an action which is like print the file names. So, what it does is just looks at the entire home directory and every file that is found it will just print out the file.

So, you can actually find out how many files are there in the home directory simply by using this command. So, in my home directory, I have 70,000 files, of course, you may be wondering how come I have so many paths, most of them are actually cache files from the browser and other utilities that are running.

(Refer Slide Time: 10:13)

```
gphanigirne:~$ find $HOME -mtime -2 -print
```

```
/idb
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/moz-extension++9
540c-e507-445f-98a0-d8795b4de2e1/userContextId=4294967295/idb
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++accounts.
.do/idb
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.youtube-
nocookie.com/.metadata-v2
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.youtube-
nocookie.com/ls
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.youtube-
nocookie.com/ls/data.sqlite
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.youtube-
nocookie.com/ls/usage
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++app.sli.do/i
db
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sli.do/.
metadata-v2
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sli.do/i
db
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sli.do/l
s
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sli.do/l
s/data.sqlite
/home/gphani/.mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sli.do/l
s/usage
/home/gphani/.ssh/known_hosts
/home/gphani/.bash_history
gphanigirne:~$
```

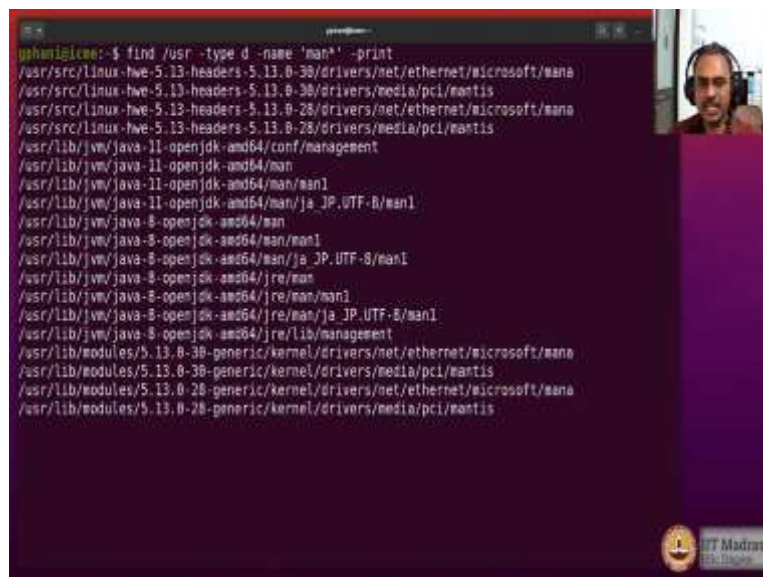
```
./mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.youtube-nocookie
./mozilla/firefox/3tbbelus.default-release/storage/default/https+++app.sli.do
./mozilla/firefox/3tbbelus.default-release/storage/default/https+++app.sli.do/idb/27106
25eaxepto.sqlite
./mozilla/firefox/3tbbelus.default-release/storage/default/https+++app.sli.do/idb/27106093
25eaxepto.files
./mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sli.do
./mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sli.do/idb/27106093
25eaxepto.sqlite
./mozilla/firefox/3tbbelus.default-release/storage/default/https+++www.sli.do/idb/27106093
25eaxepto.files
./mozilla/firefox/lo8tftce.default
./mozilla/firefox/lo8tftce.default/times.json
./mozilla/firefox/profiles.ini
./mozilla/firefox/Pending Pings
./mozilla/firefox/installs.ini
./mozilla/systemextensionsdev
./mozilla/extensions
./ssh
./bashrc
./vidyo
./vidyo/VidyoDesktop
./vidyo/VidyoDesktop/ca-certificates.crt
./personal
./personal/sign.png
./personal/seal.jpg
gphanigirne:~$
```



Now, let us say, what are the files that were modified in the last two days, which require me to basically take it back for example, so, I can so, here is a command where in the home directory files that were modified in the last two days are being printed out. So, obviously, the cache folder will also be there. But you also see that there are a bunch of other files that are being reported.

So, I go to the documents folder and run the command to find out what are the files that are modified in the last two days, and this is a small folder. So, I have a limited number of files that will be listed and here are the facts that are there. And I go to my home directory, and I can say how many files are modified more than a month back and you can see that obviously, some cache files also listed, but there are a bunch of other files that are also there, which actually tells me that there are some files that I have not looked at the last one month.

(Refer Slide Time: 11:25)



```
gpham@icm:~$ find /usr -type d -name 'man*' -print
/usr/src/linux-hwe-5.13.0-38/drivers/net/ethernet/microsoft/mana
/usr/src/linux-hwe-5.13.0-38/drivers/media/pcl/mantis
/usr/src/linux-hwe-5.13.0-28/drivers/net/ethernet/microsoft/mana
/usr/src/linux-hwe-5.13.0-28/drivers/media/pcl/mantis
/usr/lib/jvm/java-11-openjdk-amd64/conf/management
/usr/lib/jvm/java-11-openjdk-amd64/man
/usr/lib/jvm/java-11-openjdk-amd64/man/man1
/usr/lib/jvm/java-11-openjdk-amd64/man/ja_JP.UTF-8/man1
/usr/lib/jvm/java-8-openjdk-amd64/man
/usr/lib/jvm/java-8-openjdk-amd64/man/man1
/usr/lib/jvm/java-8-openjdk-amd64/man/ja_JP.UTF-8/man1
/usr/lib/jvm/java-8-openjdk-amd64/jre/man
/usr/lib/jvm/java-8-openjdk-amd64/jre/man/man1
/usr/lib/jvm/java-8-openjdk-amd64/jre/man/ja_JP.UTF-8/man1
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/management
/usr/lib/modules/5.13.0-38-generic/kernel/drivers/net/ethernet/microsoft/mana
/usr/lib/modules/5.13.0-38-generic/kernel/drivers/media/pcl/mantis
/usr/lib/modules/5.13.0-28-generic/kernel/drivers/net/ethernet/microsoft/mana
/usr/lib/modules/5.13.0-28-generic/kernel/drivers/media/pcl/mantis
```

```
gpham@linux:~$ find /usr -type d -name 'man?' -print
/usr/share/man/es/man8
/usr/share/man/es/man7
/usr/share/man/pt/man1
/usr/share/man/pt/man8
/usr/share/man/pt/man8
/usr/share/man/cs/man7
/usr/share/man/cs/man1
/usr/share/man/cs/man8
/usr/share/man/cs/man8
/usr/share/man/hr/man1
/usr/share/man/et/man1
/usr/share/texlive/texmf-dist/tex4ht/ht-fonts/ascii/manfnt
/usr/share/texlive/texmf-dist/fonts/map/dvips/manfnt-font
/usr/share/texlive/texmf-dist/fonts/afm/boekwater/manfnt-font
/usr/share/texlive/texmf-dist/fonts/type1/boekwater/manfnt-font
/usr/share/texlive/texmf-dist/tex/generic/pgf/luapgf/manual
/usr/share/texlive/texmf-dist/tex/latex/man1
/usr/share/texlive/texmf-dist/tex/latex/manuscript
/usr/share/texlive/texmf-dist/tex/latex/manfnt
/usr/share/texlive/texmf-dist/tex/latex/man1ind
/usr/share/doc/texlive-doc/pdftex/manual
/usr/share/doc/texlive-doc/fonts/fontinst/manual
/usr/share/doc/flex/examples/manual
/usr/share/doc/manpages
/usr/share/doc/python3.11/man
/usr/share/doc/po-debconf/manpages.vi
```

```
gpham@linux:~$ find /usr -type d -name 'man?' -print
```

```
gpham@linux:~$ find /usr -type d -name 'man?' -print
/usr/src/linux-hwe-5.13-headers-5.13.0-38/drivers/net/ethernet/microsoft/mana
/usr/src/linux-hwe-5.13-headers-5.13.0-28/drivers/net/ethernet/microsoft/mana
/usr/lib/jvm/java-11-openjdk-amd64/man/man1
/usr/lib/jvm/java-11-openjdk-amd64/ja_JP.UTF-8/man1
/usr/lib/jvm/java-8-openjdk-amd64/man/man1
/usr/lib/jvm/java-8-openjdk-amd64/ja_JP.UTF-8/man1
/usr/lib/jvm/java-8-openjdk-amd64/jre/man/man1
/usr/lib/jvm/java-8-openjdk-amd64/jre/man/ja_JP.UTF-8/man1
/usr/lib/modules/5.13.0-38-generic/kernel/drivers/net/ethernet/microsoft/mana
/usr/lib/modules/5.13.0-28-generic/kernel/drivers/net/ethernet/microsoft/mana
```

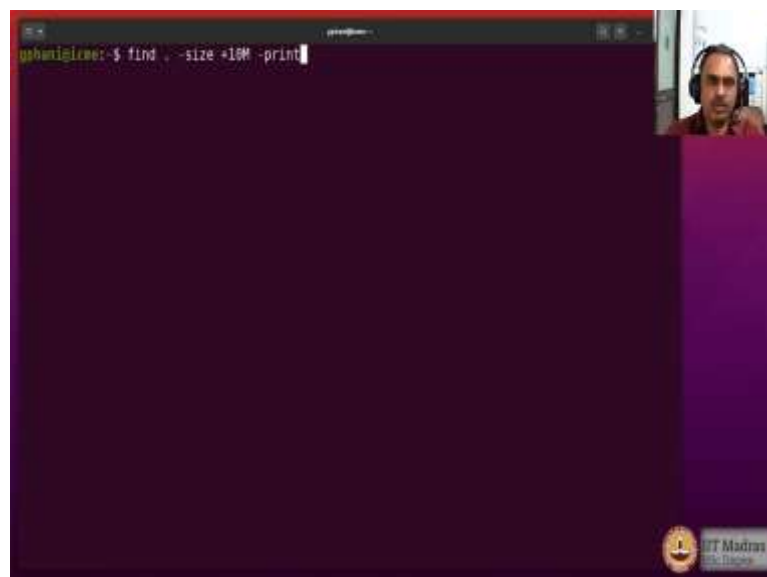
Some more examples, let us say there are some manpages that are available in the user directory and I want to know where are those directories kept? So, let us try that out. Find slash user. So, go to the slash user directory and minus type d that is find out those files of the type directory and the name should be man star. So, the name should contain the pattern man after that anything else can be followed.

And what should I do I should just print out those directory names onto the screen. So, this is a command. So, find command actually goes through the entire directory hierarchy and then looks at the options that we have given to see what are the matching or suitable files of the type that we are asking and the name that we are asking and then print them out. So, as it is going through it may take some time because of the depth of the hierarchy that is available.

And you see the reality structures to give the output. So, here do you see that the man manual page directories are all here user share man and then man 1 to man 8 there the different sections of the manpage that are being listed Of course, there are also other directories being listed which are having the word man as a part of them. So, what I could also now do is instead of star I can put a question mark so, that only one character has to be there man.

And you now see that it is a slightly lesser list, because it has only one character after man that we are trying to match. So, you can look for directories as well as for files.

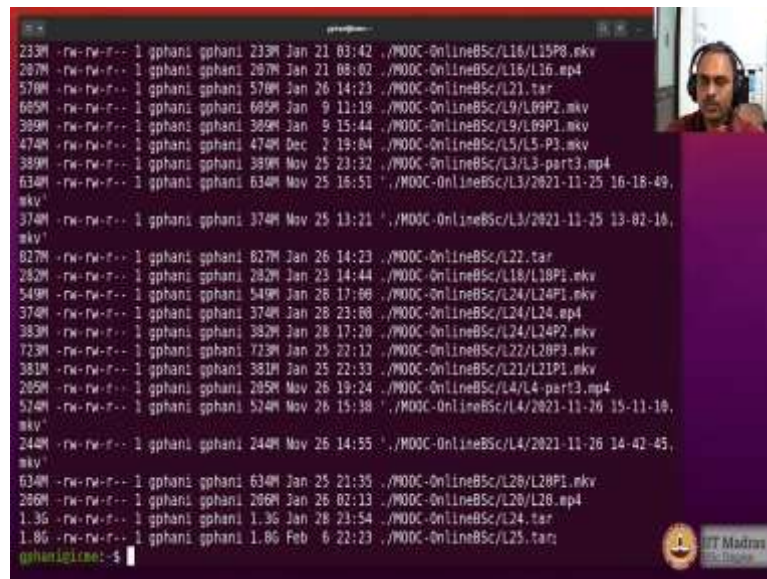
(Refer Slide Time: 13:19)



```
gphanigirae:~$ find . -size +10M -exec ls -lsh {} \;
```

```
29M -rw-rw-r-- 1 gphani gphani 29M Feb 20 2014 './MMS488/Videos/SMS group Corporate Vid  
EW (Low).mp4'  
13M -rwxrwx--- 1 gphani gphani 13M Aug 9 2014 './MMS488/Videos/lateralgrowth1.avi  
28M -rwxrwx--- 1 gphani gphani 28M Aug 9 2014 './MMS488/Videos/field.avi  
14M -rw-rw-r-- 1 gphani gphani 14M Feb 20 2014 './MMS488/Videos/continue casting steel  
w13.mp4'  
39M -rwxrwx--- 1 gphani gphani 39M Aug 9 2014 './MMS488/Videos/coarsening2.avi  
33M -rwxrwx--- 1 gphani gphani 33M Aug 9 2014 './MMS488/Videos/Superalloys.mpg  
163M -rwxrwx--- 1 gphani gphani 163M Aug 9 2014 './MMS488/Videos/cd_solid.mpeg  
36M -rwxrwx--- 1 gphani gphani 36M Aug 9 2014 './MMS488/Videos/rsp.mov  
28M -rwxrwx--- 1 gphani gphani 28M Aug 9 2014 './MMS488/Videos/CellularGrowth.avi  
17M -rwxrwx--- 1 gphani gphani 17M Aug 9 2014 './MMS488/Videos/Jaipur_Gastaldi/eq_growth 6  
_2_mini.avi  
17M -rwxrwx--- 1 gphani gphani 17M Aug 9 2014 './MMS488/Videos/Jaipur_Gastaldi/Pres Gast  
ICSSP3.ppt'  
14M -rw-rw-r-- 1 gphani gphani 14M Feb 20 2014 './MMS488/Videos/Continuous casting of steel  
1 (Low)2.mp4'  
13M -rwxrwx--- 1 gphani gphani 13M Aug 9 2014 './MMS488/Videos/in190_sm.avi  
12M -rwxrwx--- 1 gphani gphani 12M Aug 9 2014 './MMS488/Videos/olm.avi  
15M -rwxrwx--- 1 gphani gphani 15M Aug 9 2014 './MMS488/Videos/rsp.rvf  
57M -rw-rw-r-- 1 gphani gphani 57M Feb 4 14:28 './MMS488/L11.mp4  
48M -rw-rw-r-- 1 gphani gphani 48M Feb 8 09:59 './MMS488/L12.mp4  
52M -rw-rw-r-- 1 gphani gphani 52M Feb 15 10:26 './MMS488/D01.mp4  
58M -rw-rw-r-- 1 gphani gphani 58M Jan 31 11:40 './MMS488/L08.mp4  
55M -rw-rw-r-- 1 gphani gphani 55M Feb 4 12:58 './MMS488/L10.mp4  
11M -rw-rw-r-- 1 gphani gphani 11M Mar 5 14:32 './mozilla/firefox/3tbbelus.default-release  
/storage/permanent/chrome/lib/3870112724rsegmoittet-es.sqlite  
gphanigirae:~$ find . -size +10M -exec ls -lsh {} \;
```

```
374M -rw-rw-r-- 1 gphani gphani 374M Nov 25 13:21 './MOOC-OnlineBSc/L3/2021-11-25 13-02-  
mkv'  
027M -rw-rw-r-- 1 gphani gphani 027M Jan 26 14:23 './MOOC-OnlineBSc/L22.tar  
195M -rw-rw-r-- 1 gphani gphani 195M Jan 23 14:55 './MOOC-OnlineBSc/L18/L18P2.mkv  
145M -rw-rw-r-- 1 gphani gphani 145M Jan 23 18:11 './MOOC-OnlineBSc/L18/L18.mp4  
282M -rw-rw-r-- 1 gphani gphani 282M Jan 23 14:44 './MOOC-OnlineBSc/L18/L18P1.mkv  
549M -rw-rw-r-- 1 gphani gphani 549M Jan 28 17:00 './MOOC-OnlineBSc/L24/L24P1.mkv  
374M -rw-rw-r-- 1 gphani gphani 374M Jan 28 23:08 './MOOC-OnlineBSc/L24/L24.mp4  
383M -rw-rw-r-- 1 gphani gphani 383M Jan 28 17:20 './MOOC-OnlineBSc/L24/L24P2.mkv  
723M -rw-rw-r-- 1 gphani gphani 723M Jan 25 22:12 './MOOC-OnlineBSc/L22/L20P3.mkv  
184M -rw-rw-r-- 1 gphani gphani 184M Jan 26 01:37 './MOOC-OnlineBSc/L22/L22.mp4  
381M -rw-rw-r-- 1 gphani gphani 381M Jan 25 22:33 './MOOC-OnlineBSc/L21/L21P1.mkv  
189M -rw-rw-r-- 1 gphani gphani 189M Jan 26 02:13 './MOOC-OnlineBSc/L21/L21.mp4  
196M -rw-rw-r-- 1 gphani gphani 196M Nov 26 15:10 './MOOC-OnlineBSc/L4/2021-11-26 14-59-52.  
mkv'  
205M -rw-rw-r-- 1 gphani gphani 205M Nov 26 19:24 './MOOC-OnlineBSc/L4/L4-part1.mp4  
514M -rw-rw-r-- 1 gphani gphani 514M Nov 26 15:38 './MOOC-OnlineBSc/L4/2021-11-26 15-11-10.  
mkv'  
244M -rw-rw-r-- 1 gphani gphani 244M Nov 26 14:55 './MOOC-OnlineBSc/L4/2021-11-26 14-42-45.  
mkv'  
634M -rw-rw-r-- 1 gphani gphani 634M Jan 25 21:35 './MOOC-OnlineBSc/L20/L20P1.mkv  
266M -rw-rw-r-- 1 gphani gphani 266M Jan 26 02:13 './MOOC-OnlineBSc/L20/L20.mp4  
1.3G -rw-rw-r-- 1 gphani gphani 1.3G Jan 28 23:54 './MOOC-OnlineBSc/L24.tar  
1.0G -rw-rw-r-- 1 gphani gphani 1.0G Feb 6 22:23 './MOOC-OnlineBSc/L25.tar  
131M -rw-rw-r-- 1 gphani gphani 131M Feb 20 2014 './MMS488/Videos/The production process o  
f a steel casting (HD).mp4'  
163M -rwxrwx--- 1 gphani gphani 163M Aug 9 2014 './MMS488/Videos/cd_solid.mpeg  
gphanigirae:~$
```



```
233M -rw-rw-r-- 1 gphani gphani 233M Jan 21 03:42 ./M00C-OnlineBSc/L16/L15P8.mkv
207M -rw-rw-r-- 1 gphani gphani 207M Jan 21 08:02 ./M00C-OnlineBSc/L16/L16.mp4
570M -rw-rw-r-- 1 gphani gphani 570M Jan 26 14:23 ./M00C-OnlineBSc/L21.tar
605M -rw-rw-r-- 1 gphani gphani 605M Jan 9 11:19 ./M00C-OnlineBSc/L9/L09P2.mkv
309M -rw-rw-r-- 1 gphani gphani 309M Jan 9 15:44 ./M00C-OnlineBSc/L9/L09P1.mkv
474M -rw-rw-r-- 1 gphani gphani 474M Dec 2 19:04 ./M00C-OnlineBSc/L5/L5-P3.mkv
389M -rw-rw-r-- 1 gphani gphani 389M Nov 25 23:32 ./M00C-OnlineBSc/L3/L3-part3.mp4
634M -rw-rw-r-- 1 gphani gphani 634M Nov 25 16:51 ./M00C-OnlineBSc/L3/2021-11-25 16-18-40,
mkv
374M -rw-rw-r-- 1 gphani gphani 374M Nov 25 13:21 ./M00C-OnlineBSc/L3/2021-11-25 13-02-16,
mkv
827M -rw-rw-r-- 1 gphani gphani 827M Jan 26 14:23 ./M00C-OnlineBSc/L22.tar
282M -rw-rw-r-- 1 gphani gphani 282M Jan 23 14:44 ./M00C-OnlineBSc/L18/L18P1.mkv
549M -rw-rw-r-- 1 gphani gphani 549M Jan 26 17:06 ./M00C-OnlineBSc/L24/L24P1.mkv
374M -rw-rw-r-- 1 gphani gphani 374M Jan 28 23:08 ./M00C-OnlineBSc/L24/L24.mp4
383M -rw-rw-r-- 1 gphani gphani 383M Jan 28 17:20 ./M00C-OnlineBSc/L24/L24P2.mkv
723M -rw-rw-r-- 1 gphani gphani 723M Jan 25 22:12 ./M00C-OnlineBSc/L22/L20P3.mkv
381M -rw-rw-r-- 1 gphani gphani 381M Jan 25 22:33 ./M00C-OnlineBSc/L21/L21P1.mkv
205M -rw-rw-r-- 1 gphani gphani 205M Nov 26 19:24 ./M00C-OnlineBSc/L4/L4-part3.mp4
574M -rw-rw-r-- 1 gphani gphani 574M Nov 26 15:38 ./M00C-OnlineBSc/L4/2021-11-26 15-11-10,
mkv
244M -rw-rw-r-- 1 gphani gphani 244M Nov 26 14:55 ./M00C-OnlineBSc/L4/2021-11-26 14-42-45,
mkv
634M -rw-rw-r-- 1 gphani gphani 634M Jan 25 21:35 ./M00C-OnlineBSc/L28/L28P1.mkv
206M -rw-rw-r-- 1 gphani gphani 206M Jan 26 02:13 ./M00C-OnlineBSc/L20/L20.mp4
1.3G -rw-rw-r-- 1 gphani gphani 1.3G Jan 28 21:54 ./M00C-OnlineBSc/L24.tar
1.8G -rw-rw-r-- 1 gphani gphani 1.8G Feb 6 22:23 ./M00C-OnlineBSc/L25.tar
gphani@mac:~$
```

Now, let us say, we want to look at what are the files that are larger than a particular limit. So, I want to see what files are more than 100 MB, for example. And I want to print them out to see whether I need to move them out or compress them to save some disk space and so on. So, here, what we are doing is, in the current directory, my home directory, I am looking at files which are greater than 10 megabytes, I want to just print them out.

So, I have a lot of files. Now, just merely printing them does not help me because I do not know the individual sizes. So, what I would do is I would like to execute a code on them. And the execution of the code should be `ls -lh` So, that the human readable size of the file should be listed along with the file name and it is a long listing and what is the file name I am using the paste placeholder which is the braces side by side and I need to close the line execution line.

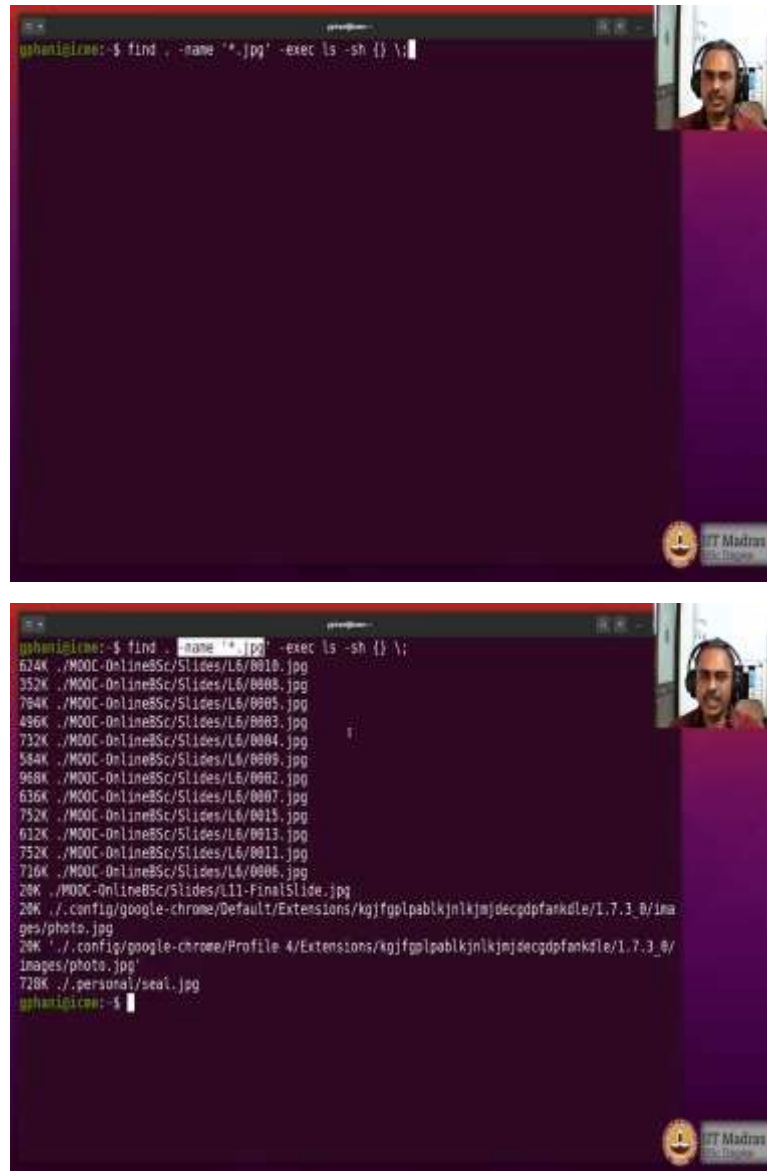
So, I escaped the semicolon because that is interpreted by Shell in different manner. So, I just pass it on only to the find command that way. So, what will happen is that every file that is matching the condition of being greater than 10 megabytes, that file name will be inserted at the place that I am indicating here. And the command that is listed here will be executed which is basically a `ls -lh file name`.

So, you know what the output would look like. So, the output would have the file size in a human readable form followed by the long listing. So, you now see that we are able to look at the file sizes of all those which are larger than 10 mi 10 megabytes. So, we can then go to the 100 megabytes, and look at the large files. So, it seems that I have very large number of files that are also very big.



So, I go for 500, maybe 200. So, I still have large number of files that are so, huge. So, these are all basically the videos I have prepared for this free course. And I can now see if some of them need to be compressed. So, I can go on to compress and see whether I can save some space.

(Refer Slide Time: 15:51)



The image shows two screenshots of a terminal window. The top screenshot shows the command `gphanigirne: $ find . -name '*.jpg' -exec ls -sh {} \;` being entered. The bottom screenshot shows the output of the command, listing various .jpg files and their sizes in a human-readable format (K for kilobytes).

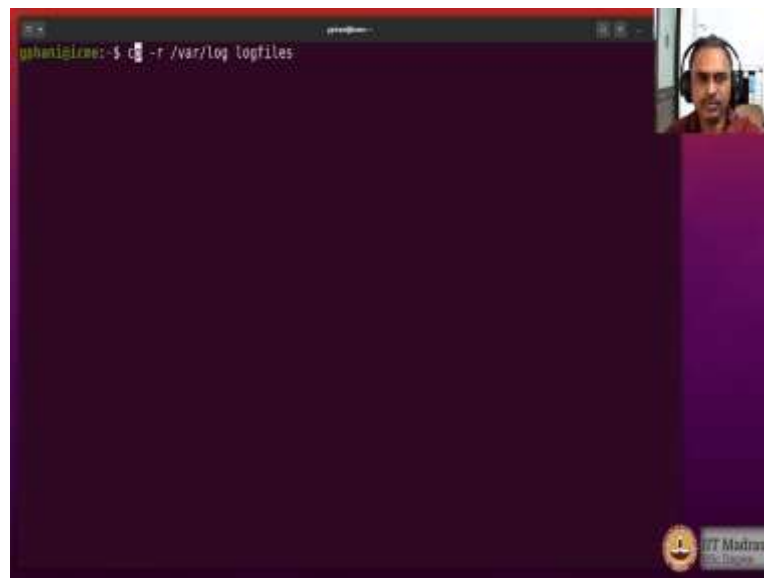
```
gphanigirne: $ find . -name '*.jpg' -exec ls -sh {} \;
674K ./MOOC-OnlineBSc/Slides/L6/0010.jpg
352K ./MOOC-OnlineBSc/Slides/L6/0008.jpg
794K ./MOOC-OnlineBSc/Slides/L6/0005.jpg
496K ./MOOC-OnlineBSc/Slides/L6/0003.jpg
732K ./MOOC-OnlineBSc/Slides/L6/0004.jpg
584K ./MOOC-OnlineBSc/Slides/L6/0009.jpg
988K ./MOOC-OnlineBSc/Slides/L6/0002.jpg
636K ./MOOC-OnlineBSc/Slides/L6/0007.jpg
752K ./MOOC-OnlineBSc/Slides/L6/0015.jpg
612K ./MOOC-OnlineBSc/Slides/L6/0013.jpg
752K ./MOOC-OnlineBSc/Slides/L6/0011.jpg
716K ./MOOC-OnlineBSc/Slides/L6/0006.jpg
28K ./MOOC-OnlineBSc/Slides/L11-FinalSlide.jpg
28K ./config/google-chrome/Default/Extensions/kgjfgplpablkjnlkjmjdecgdpfankdle/1.7.3_0/images/photo.jpg
28K ./config/google-chrome/Profile 4/Extensions/kgjfgplpablkjnlkjmjdecgdpfankdle/1.7.3_0/images/photo.jpg
728K ./personal/seal.jpg
gphanigirne: $
```

I could also look for a particular type of file and then do an action. So, every jpg file, for example, I want to just see whether any of those are large in size. So, what this command does is in the current directory, it will go on to look for all the files which have a pattern of the name, star dot jpg. And then on them, it will execute the command ls minus sh and then the pattern.

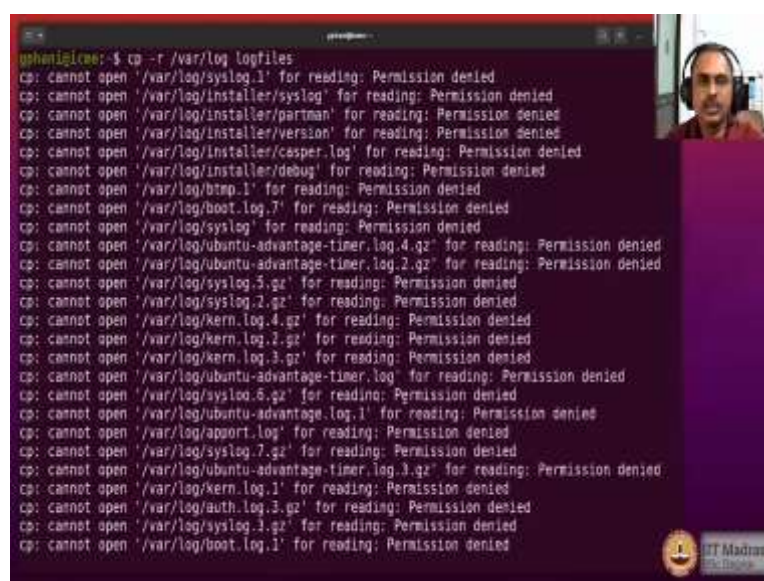
So, you can see all the jpg files are listed. And the size is given in a human readable fashion here. And so, if I see that some of them are too big, and I want to do something to them, like reduce the number of pixels, et cetera I can do that. So, you could actually use the find utility to identify the type of files that you want to work on them. So, exec is a very powerful option on them. And the syntax here is very specific. So, you have to escape the semicolon.

And remember that this will be executed on every file that matches the particular pattern that we gave. So, you could also move those files to a different place using the exec command. But you know that those actions are quite serious. And so, use them as u dim necessary for your directory.

(Refer Slide Time: 17:29)



```
gphnigizme:~$ cp -r /var/log logfiles
```



```
gphnigizme:~$ cp -r /var/log logfiles
cp: cannot open '/var/log/syslog.1' for reading: Permission denied
cp: cannot open '/var/log/installer/syslog' for reading: Permission denied
cp: cannot open '/var/log/installer/partman' for reading: Permission denied
cp: cannot open '/var/log/installer/version' for reading: Permission denied
cp: cannot open '/var/log/installer/cesper.log' for reading: Permission denied
cp: cannot open '/var/log/installer/debug' for reading: Permission denied
cp: cannot open '/var/log/btmp.1' for reading: Permission denied
cp: cannot open '/var/log/boot.log.7' for reading: Permission denied
cp: cannot open '/var/log/syslog' for reading: Permission denied
cp: cannot open '/var/log/ubuntu-advantage-timer.log.4.gz' for reading: Permission denied
cp: cannot open '/var/log/ubuntu-advantage-timer.log.2.gz' for reading: Permission denied
cp: cannot open '/var/log/syslog.5.gz' for reading: Permission denied
cp: cannot open '/var/log/syslog.2.gz' for reading: Permission denied
cp: cannot open '/var/log/kern.log.4.gz' for reading: Permission denied
cp: cannot open '/var/log/kern.log.2.gz' for reading: Permission denied
cp: cannot open '/var/log/kern.log.3.gz' for reading: Permission denied
cp: cannot open '/var/log/ubuntu-advantage-timer.log' for reading: Permission denied
cp: cannot open '/var/log/syslog.6.gz' for reading: Permission denied
cp: cannot open '/var/log/ubuntu-advantage.log.1' for reading: Permission denied
cp: cannot open '/var/log/apport.log' for reading: Permission denied
cp: cannot open '/var/log/syslog.7.gz' for reading: Permission denied
cp: cannot open '/var/log/ubuntu-advantage-timer.log.3.gz' for reading: Permission denied
cp: cannot open '/var/log/kern.log.1' for reading: Permission denied
cp: cannot open '/var/log/auth.log.3.gz' for reading: Permission denied
cp: cannot open '/var/log/syslog.3.gz' for reading: Permission denied
cp: cannot open '/var/log/boot.log.1' for reading: Permission denied
```

```

log: Permission denied
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649
6c65741e14be5c8-0000000037b7987-0005d8a7f6b71994.journal' for reading: Permission denie
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649
6c65741e14be5c8-0000000031df35b-0005d8a7f6b71994.journal' for reading: Permission denie
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649008
6c65741e14be5c8-0000000030dee73-0005d8a7f6b71994.journal' for reading: Permission denied
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649008
6c65741e14be5c8-0000000036b4363-0005d8a7f6b71994.journal' for reading: Permission denied
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649008
6c65741e14be5c8-000000003780edf-0005d8a7f6b71994.journal' for reading: Permission denied
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649008
6c65741e14be5c8-00000000345d136-0005d8a7f6b71994.journal' for reading: Permission denied
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649008
6c65741e14be5c8-0000000037e3a4f-0005d8a7f6b71994.journal' for reading: Permission denied
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649008
6c65741e14be5c8-0000000033794e5-0005d8a7f6b71994.journal' for reading: Permission denied
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649008
6c65741e14be5c8-0000000037d5ebc-0005d8a7f6b71994.journal' for reading: Permission denied
cp: cannot open '/var/log/journal/ef75e947e5e14591b862612777e5fd78/system@81997add5d9649008
6c65741e14be5c8-0000000037f13c8-0005d8a7f6b71994.journal' for reading: Permission denied
cp: cannot access '/var/log/private': Permission denied
cp: cannot access '/var/log/speech-dispatcher': Permission denied
cp: cannot access '/var/log/unattended-upgrades': Permission denied
cp: cannot open '/var/log/boot.log.2' for reading: Permission denied
cp: cannot open '/var/log/apport.log.3.gz' for reading: Permission denied
cp: cannot open '/var/log/apport.log.2.gz' for reading: Permission denied
gghnigicme:~$

```

```

gghnigicme:~/logfiles$ ls
alternatives.log      dmccg
alternatives.log.1    dmccg.0
alternatives.log.10.gz dmccg.1.gz
alternatives.log.11.gz dmccg.2.gz
alternatives.log.12.gz dmccg.3.gz
alternatives.log.2.gz  dmccg.4.gz
alternatives.log.3.gz  dpkg.log
alternatives.log.4.gz  dpkg.log.1
alternatives.log.5.gz  dpkg.log.10.gz
alternatives.log.6.gz  dpkg.log.11.gz
alternatives.log.7.gz  dpkg.log.12.gz
alternatives.log.8.gz  dpkg.log.1.gz-2021060916.backup
alternatives.log.9.gz  dpkg.log.2.gz
apt                  dpkg.log.3.gz
bootstrap.log        dpkg.log.4.gz
cups                 dpkg.log.5.gz
dist-upgrade         dpkg.log.6.gz
gghnigicme:~/logfiles$

```

Now to illustrate the compression capability et cetera. Let me get a copy of the log folders because that is something that I can try out. So, copy minus r var log to a folder called log files. So, there are some files I am not supposed to be having permission to read. So, nevertheless, the rest of them are available. So, you see that a bunch of files have come. So, these are all the log files that have come.

And I can find out how much size it occupies 450 megabytes of size. And minus lr will give me recursively and I can see that a lot of depth is there here.

(Refer Slide Time: 18:26)

```
gphanigirne:~/logfiles$ cd ..
gphanigirne:~$ du -sh logfiles/
450M    logfiles/
gphanigirne:~$ tar -cvf logfiles.tar logfiles/
```

```
gphanigirne:~$ ls -l
total 470592
drwx----- 2 gphani gphani 4096 Dec 6 09:52 bin
drwx----- 2 gphani gphani 4096 Nov 25 16:58 Desktop
drwx----- 3 gphani gphani 4096 Mar 6 21:04 Documents
drwx----- 3 gphani gphani 4096 Mar 5 14:54 Downloads
-rw-rw-r-- 1 gphani gphani 503090 Mar 6 21:15 hwinfo.txt
drwxrwxr-x 14 gphani gphani 4096 Mar 6 22:17 logfiles
-rw-rw-r-- 1 gphani gphani 401310720 Mar 6 22:18 logfiles.tar
-rw-rw-r-- 1 gphani gphani 11122 Mar 6 21:17 lshw.txt
-rw-rw-r-- 1 gphani gphani 256 Mar 6 15:23 makefile
drwx----- 3 gphani gphani 4096 Mar 4 12:58 MMS480
drwx----- 27 gphani gphani 4096 Feb 6 22:25 MISC-OnlineMSc
drwx----- 2 gphani gphani 4096 Jul 28 2020 Music
drwx----- 2 gphani gphani 4096 Mar 5 10:52 Pictures
drwx----- 2 gphani gphani 4096 Jul 28 2020 Public
drwx----- 3 gphani gphani 4096 Jul 30 2020 snap
drwx----- 3 gphani gphani 4096 Nov 9 20:59 Templates
drwx----- 4 gphani gphani 4096 Mar 6 21:57 Videos
gphanigirne:~$
```

```
drwx----- 3 gphani gphani 4.0K Mar 4 12:58 MMS480
drwx----- 27 gphani gphani 4.0K Feb 6 22:25 MISC-OnlineMSc
drwx----- 2 gphani gphani 4.0K Jul 28 2020 Music
drwx----- 2 gphani gphani 4.0K Mar 5 10:52 Pictures
drwx----- 2 gphani gphani 4.0K Jul 28 2020 Public
drwx----- 3 gphani gphani 4.0K Jul 30 2020 snap
drwx----- 3 gphani gphani 4.0K Nov 9 20:59 Templates
drwx----- 4 gphani gphani 4.0K Mar 6 21:57 Videos
gphanigirne:~$ gzip logfiles.tar
ls -lh logfiles.tar.gzgphanigirne:~$ ls -lh logfiles.tar.gz
-rw-rw-r-- 1 gphani gphani 23M Mar 6 22:18 logfiles.tar.gz
gphanigirne:~$ gunzip logfiles.tar.gz
ls -lh logfiles.tar
450M -rw-rw-r-- 1 gphani gphani 450M Mar 6 22:18 logfiles.tar
gphanigirne:~$
gphanigirne:~$
gphanigirne:~$
gphanigirne:~$ bzip2 logfiles.tar
ls -l logfiles.tar.bz2gphanigirne:~$ ls -l logfiles.tar.bz2
-rw-rw-r-- 1 gphani gphani 16581711 Mar 6 22:18 logfiles.tar.bz2
gphanigirne:~$ ls -lh logfiles.tar.bz2
-rw-rw-r-- 1 gphani gphani 16M Mar 6 22:18 logfiles.tar.bz2
gphanigirne:~$ bzip2 -d logfiles.tar.bz2
gphanigirne:~$ compress logfiles.tar
gphanigirne:~$ ls -lh logfiles.tar.Z
-rw-rw-r-- 1 gphani gphani 45M Mar 6 22:18 logfiles.tar.Z
gphanigirne:~$
```

So, what I can do is tar. So, first, let me just see. So, it is occupying 450 megabytes. So, what I will do is tar minus cvf log file dot tar. So, here, what I am doing is I am creating a bundle of all the files, which are in the directory called log files. And I am naming the bundle is logfiles dot tar. So, it is no bundling. And that bundle if you see here, it has roughly the same size, you know, it is almost the same size.

So, 460 megabytes about 10 megabytes more because it has to have the information about those files, in which sequence it is keeping in the particular paper archive file. Now I can compress this and see how much it can come down. So, you see now the file has been shrunk from 460 megabytes to 23 megabytes, that is a ratio of almost 1 is to 20 and that is because the log files are mainly text files.

And they have very repetitive kind of pattern inside them. And therefore, I can actually keep the entire log files directory with just about 5 percent of the size that would occupy otherwise. Now, let us look at the efficiency of comparison with the other utilities also. So, I would unzip that so, it is 460 megabytes.

So, now I will zip it using another utility bzip2 as you can make out it takes a little more time than G zip, and it has shrunk it to 16 megabytes instead of 23 megabytes which means that the bzip2 is slightly more efficient in shrinking, but it takes more time to shirk it. Now, that is a balance that you need to pick. So, if you are desperate and you have enough ram that this compression process can finish in reasonable time, you can actually shrink it to the maximum possible using bzip2.

But if you are having limited resources, then gzip is pretty good. Now there is some other utility which also takes a little less time. So, that is called compress I am unzipping the file so, that I can try another algorithm. So, compress so, you see that the compress has done quite fast, but it has shrunk it only 45 megabytes. So, you can see that compresses very fast, but it shrinks not so much.

Bzip2 is a bit slow, but it shrinks very well. And gzip is something in between. So, you have to choose what you like to use for a particular application. But in any case, for a directory which has a lot of sub directories and not a small files, it is always a good idea to package those as a tar file if you do not need those files regularly. So, whenever you need them, you can always un tar the file and start using them again.



(Refer Slide Time: 22:32)

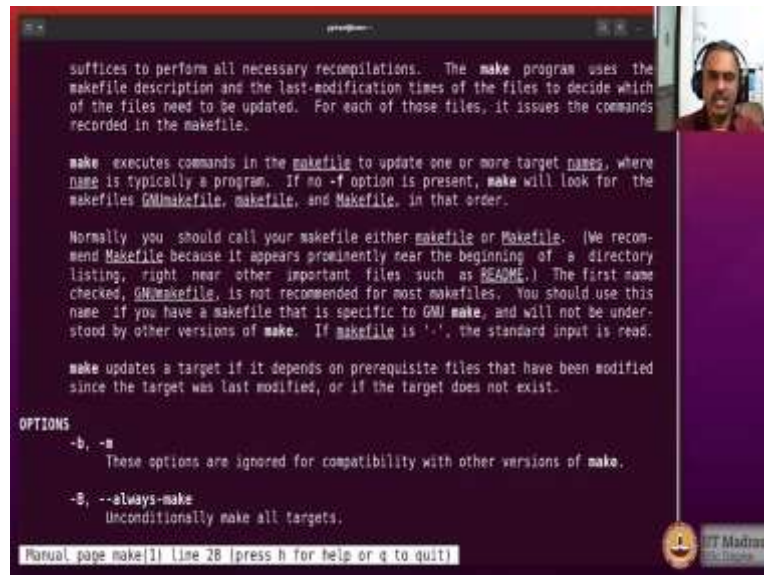
```
gphanigirne:~$ ls
bin      Downloads  logfiles.tar.Z  MMS400  Pictures  Templates
Desktop  hwinfo.txt  lshw.txt       MDOC-OnlineMSc  Public   Videos
Documents logfiles    make.file      Music    snap
gphanigirne:~$ rm -rf log
```

```
gphanigirne:~$ ls
bin      Documents  hwinfo.txt  lshw.txt  MMS400  Music  Public  Templat
Desktop  Downloads  logfiles.tar.Z  make.file  MDOC-OnlineMSc  Pictures  snap  Videos
gphanigirne:~$ uncompress logfiles.tar.Z
gphanigirne:~$ tar -xvf lz
```

```
logfiles/dpkg.log.11.gz
logfiles/dmesg.3.gz
logfiles/alternatives.log.7.gz
logfiles/dmesg.2.gz
logfiles/alternatives.log.6.gz
logfiles/dmesg
gphanigirne:~$ ls -l
total 476592
drwx----- 2 gphani gphani 4096 Dec 6 09:52 bin
drwx----- 2 gphani gphani 4096 Nov 25 16:50 Desktop
drwx----- 3 gphani gphani 4096 Mar 6 21:04 Documents
drwx----- 3 gphani gphani 4096 Mar 5 14:54 Downloads
-rw-rw-r-- 1 gphani gphani 503090 Mar 6 21:15 hwinfo.txt
drwxrwxr-x 14 gphani gphani 4096 Mar 6 22:17 logfiles
-rw-rw-r-- 1 gphani gphani 401310720 Mar 6 22:18 logfiles.tar
-rw-rw-r-- 1 gphani gphani 11122 Mar 6 21:17 lshw.txt
-rw-rw-r-- 1 gphani gphani 256 Mar 6 15:23 make.file
drwx----- 3 gphani gphani 4096 Mar 4 12:58 MMS400
drwx----- 27 gphani gphani 4096 Feb 6 22:25 MDOC-OnlineMSc
drwx----- 2 gphani gphani 4096 Jul 28 2020 Music
drwx----- 2 gphani gphani 4096 Mar 5 10:52 Pictures
drwx----- 2 gphani gphani 4096 Jul 28 2020 Public
drwx----- 3 gphani gphani 4096 Jul 30 2020 snap
drwx----- 3 gphani gphani 4096 Nov 9 20:59 Templates
drwx----- 4 gphani gphani 4096 Mar 6 21:57 Videos
gphanigirne:~$ du -sh logfiles
460M logfiles
gphanigirne:~$
```

So, how do I do that. So, the directory is locked was already there. So, I remove that. So, now you see that the directory is not there. So, I would uncompress and tar minus xvf and now the directories back. So, you see that we are able to unzip and untar. The archive files quite quickly. So, it is a very good idea to invest some time in mastering this particular utility, which could save you a lot of space and also effort in transferring files across systems. If you have large number of them in a very deep hierarchy or file system.

(Refer Slide Time: 23:25)



```

suffices to perform all necessary recompilations.  The make program uses the
makefile description and the last-modification times of the files to decide which
of the files need to be updated.  For each of those files, it issues the commands
recorded in the makefile.

make executes commands in the makefile to update one or more target names, where
name is typically a program.  If no -f option is present, make will look for the
makefiles GNUmakefile, makefile, and Makefile, in that order.

Normally, you should call your makefile either makefile or Makefile.  (We recom-
mend Makefile because it appears prominently near the beginning of a directory
listing, right near other important files such as README.)  The first name
checked, GNUmakefile, is not recommended for most makefiles.  You should use this
name if you have a makefile that is specific to GNU make, and will not be under-
stood by other versions of make.  If makefile is '.', the standard input is read.

make updates a target if it depends on prerequisite files that have been modified
since the target was last modified, or if the target does not exist.

OPTIONS
-b, -s
    These options are ignored for compatibility with other versions of make.

-B, --always-make
    Unconditionally make all targets.

Manual page make[1] line 28 (press h for help or q to quit)
```



```

gphanigirne:~$ ls
bin      Downloads  logfiles.tar  MW3400      Pictures  Templates
Desktop  hwinfo.txt  lshw.txt     #B0C-0a1neB5:  Public    Videos
Documents logfiles    make.file    Music        snap

gphanigirne:~$ more make.file
BACKUPDIR=/home/gphani/backup
DATESTAMP=$(shell date +%Y-%m-%d_%H_%M)
TARBALL=backup_${DATESTAMP}.tar

backup: file1 file2
@echo "Backing up file1 file2 to ${DATESTAMP}"
tar -cvf ${TARBALL} file1 file2
@ls -l ${TARBALL}
mv ${TARBALL} ${BACKUPDIR}

gphanigirne:~$ mkdir backup
gphanigirne:~$ touch file1
gphanigirne:~$ touch file2
gphanigirne:~$
```

```
gphanigirne:~$ cp logfiles/
alternatives.log
alternatives.log.1
alternatives.log.10.gz
alternatives.log.11.gz
alternatives.log.12.gz
alternatives.log.2.gz
alternatives.log.3.gz
alternatives.log.4.gz
alternatives.log.5.gz
alternatives.log.6.gz
alternatives.log.7.gz
alternatives.log.8.gz
alternatives.log.9.gz
apt/
bootstrap.log
cups/
dist-upgrade/
dmesg
dmesg.8
dmesg.1.gz
dmesg.2.gz
dmesg.3.gz
dmesg.4.gz
dpkg.log
dpkg.log.1
dpkg.log.10.gz
gphanigirne:~$ cp logfiles/
dpkg.log.11.gz
dpkg.log.12.gz
dpkg.log.1.gz-2021060916.backup
dpkg.log.2.gz
dpkg.log.3.gz
dpkg.log.4.gz
dpkg.log.5.gz
dpkg.log.6.gz
dpkg.log.7.gz
dpkg.log.8.gz
dpkg.log.9.gz
faillog
fontconfig.log
gdm3/
gpu-manager.log
hp/
installer/
journal/
lastlog
openvpn/
private/
speech-dispatcher/
teamviewer15/
unattended-upgrades/
wtmp
```

```
gphanigirne:~$ ls
backup  documents  file2      logfiles.tar  MMS438  Pictures  Templates
bin     Downloads  hinfo.txt  lshw.txt      NMC-OnlineBsc  Public    Videos
Desktop file1  logfiles  make.file     Music      snap

gphanigirne:~$ make -f make.file backup
Backing up file1 file2 to 2022-03-06 22:26
tar -cvf backup_2022-03-06_22.26.tar file1 file2
file1
file2
-rw-rw-r-- 1 gphani gphani 174880 Mar  6 22:26 backup_2022-03-06_22.26.tar
mv backup_2022-03-06_22.26.tar /home/gphani/backup
gphanigirne:~$ ls -l backup
total 172
-rw-rw-r-- 1 gphani gphani 174880 Mar  6 22:26 backup_2022-03-06_22.26.tar
gphanigirne:~$ date
Sunday 06 March 2022 10:27:11 PM IST
gphanigirne:~$ make -f make.file backup
make: 'backup' is up to date.
gphanigirne:~$ ls -> file1
gphanigirne:~$ make -f make.file backup
Backing up file1 file2 to 2022-03-06 22:27
tar -cvf backup_2022-03-06_22.27.tar file1 file2
file1
file2
-rw-rw-r-- 1 gphani gphani 174880 Mar  6 22:27 backup_2022-03-06_22.27.tar
mv backup_2022-03-06_22.27.tar /home/gphani/backup
gphanigirne:~$
```

```
gphanigirne:~$ ls -l backup/
total 344
-rw-rw-r-- 1 gphani gphani 174880 Mar  6 22:26 backup_2022-03-06_22.26.tar
-rw-rw-r-- 1 gphani gphani 174880 Mar  6 22:27 backup_2022-03-06_22.27.tar
gphanigirne:~$
```

The make utility is also useful as a shell script extension for some maintenance activities. As I mentioned to you it is actually meant, meant for conditional compilation of source codes, but let us use it for also other purposes. So, the manpage is quite extensive, you can read through that. And an example of that is available. So, here is an example I have created. So, what it does is a variable has been created.

Which has the value of the directory, which is the destination for keeping the backup, and then am storing the date timestamp by using a shell command so, that it is actually stored in a variable. Now, this colon in front of the equal to is like you know, a one time substitution of this particular variable. Whereas if you have equal to, then that means that every time this particular variable is occurring, the substitution is performed.

It is a small difference between those two usages in make syntax. And the name of the tar file is being given here as backup underscore the stamp dot tar. And what am I backing up I am backing up two files, file 1 and file 2. And the actions are all listed here where I am displaying a text saying I am backing up the files and then I am making a tar file and then I am looking at the size of the tar file and then I am moving into the backup directory.

So, let me make up make a directory for backup so, the directory is available. And the files that I need to backup are also to be created. So, let me create them using the touch file one, touch file 2. And there is nothing in them yet. So, what I would do is copy in the log file so, there is some contents that I could copy across. So, dmesg. So, let me see the size of it. So, that is pretty large. So, copy that to file 1 and also file 2.

So, now I have got file 1 and file 1 that are there and make file. So, make backup. So, when I run like that, what it will do is, it will actually look at a file called make file that is available. And in that make file, what are the commands available, one of the targets is to be written as backup, and it will try to perform that particular action. What is given below it within indentation, but we do not have a file called make file.

So, we need to tell that file saying that it is the comments are actually make dot file, not make file with a capital M. So, when we run that, what it does is it will actually backup those two files to a directory called backup under the home directory and as a tar file. So, let us look at the contents of the backup directory. And you can see that it has a tar file, which contains a date timestamp, and it contains the two files that we have just wanted to backup.

Now this is a very nice utility, because every time you run, it would actually change the file name. So, what is the date so, you can see that now the time is different. So, let us go ahead and try that again. Make file would perform actions only when the target files have changed. So, I have already backed up the file 1 and file 1, they have not changed, so, the backup needs to be performed.

So, what I will do is I would append some text to the file 1 so, that the file name has changed file content has changed. So, now if I try to make the backup, it would actually do it because one of the files has changed. So, this is a powerful feature of make where one of the dependencies has changed only then the action is performed, which is actually something that we no need to worry in a y script because that is automatically performed by the make utility.

So, look at the directory. Now you have got the two backup folders. And these are with the two different names because we have run them at two different instances of time. So, you can actually use the make utility for such actions as cleaning up the directory of temporary files, or zipping things that you do not want to occupy the file system with and so, on. Any maintenance activity can also be taken up as a make file command.