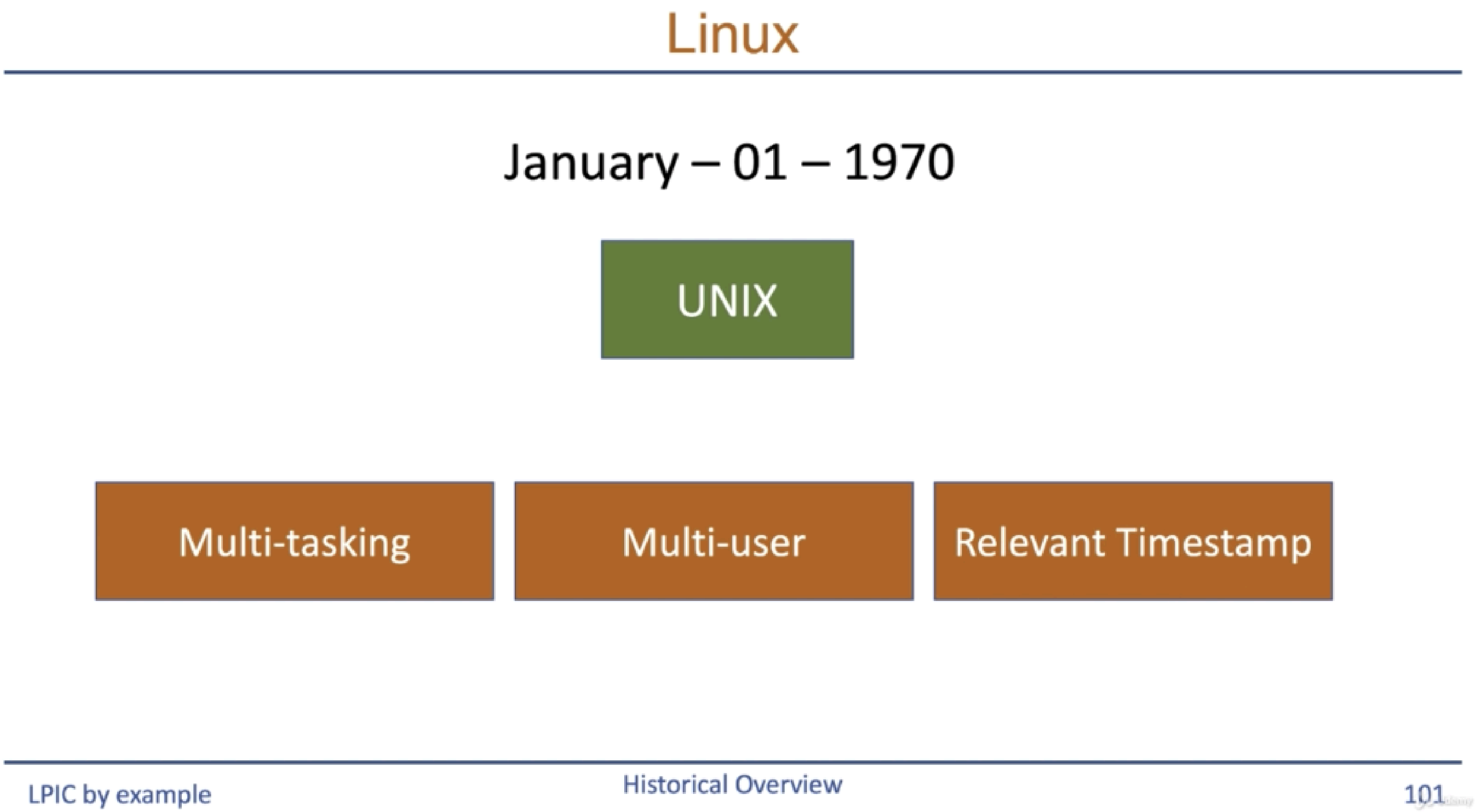
Linux History

LPIC by example



Linux

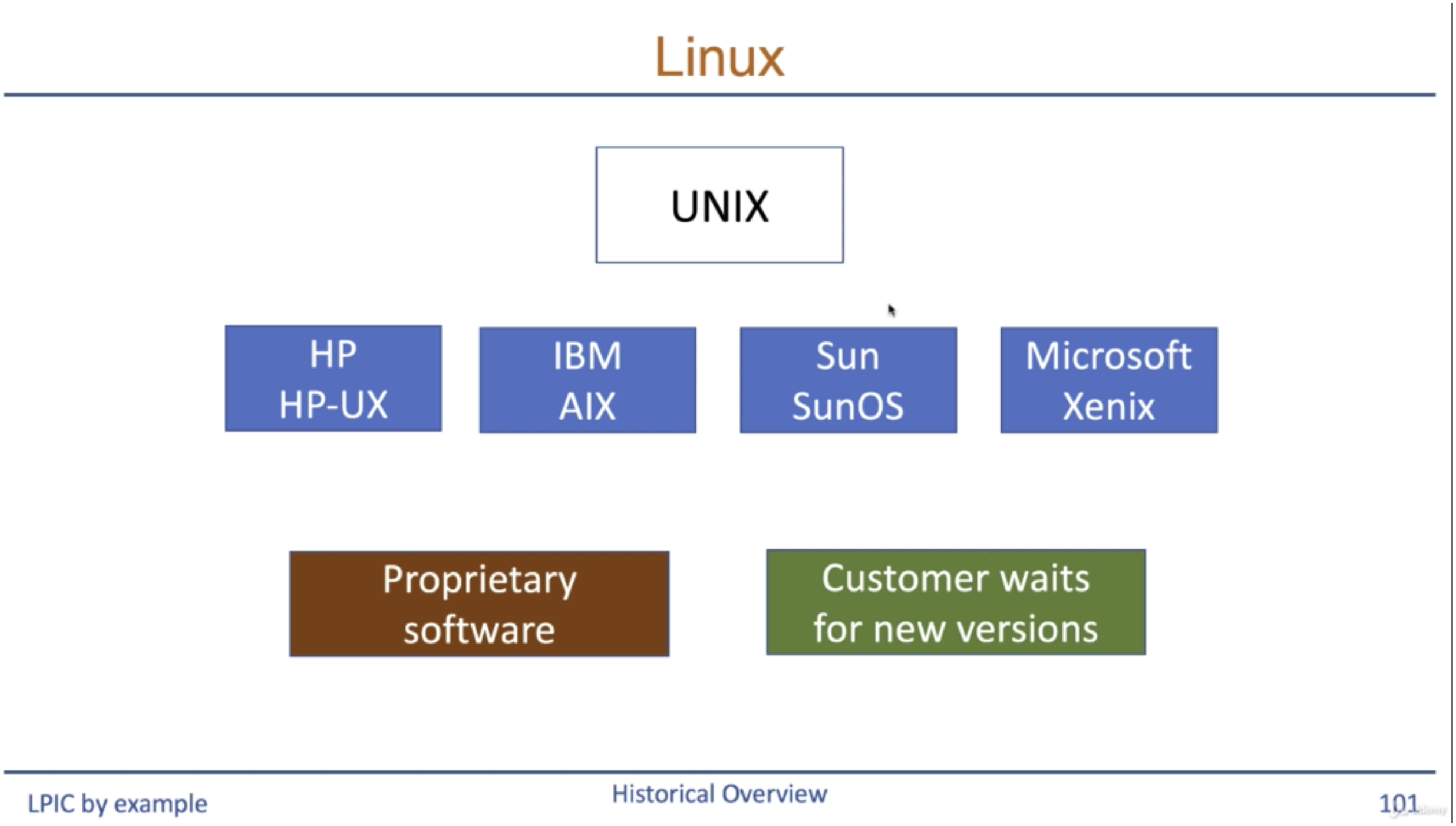
January – 01 – 1970

UNIX

Multi-tasking

Multi-user

Relevant Timestamp



Linux

UNIX

HP
HP-UX

IBM
AIX

Sun
SunOS

Microsoft
Xenix

Proprietary software

Customer waits for new versions

Linux

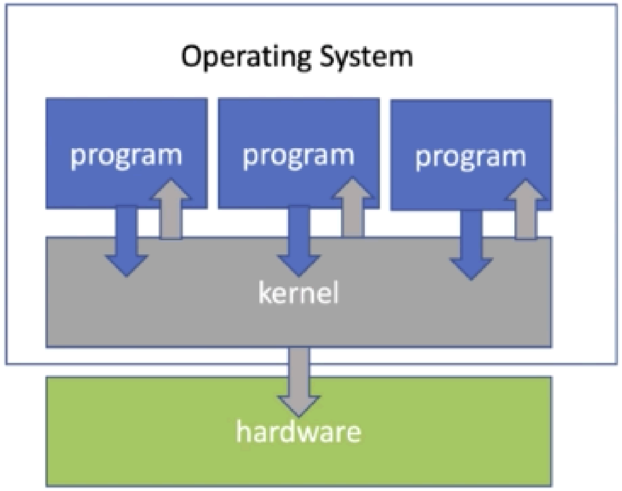
Richard Stallman

GNU Free software

Source code -> Compiler -> Machine code

Linus Torvalds

Linux



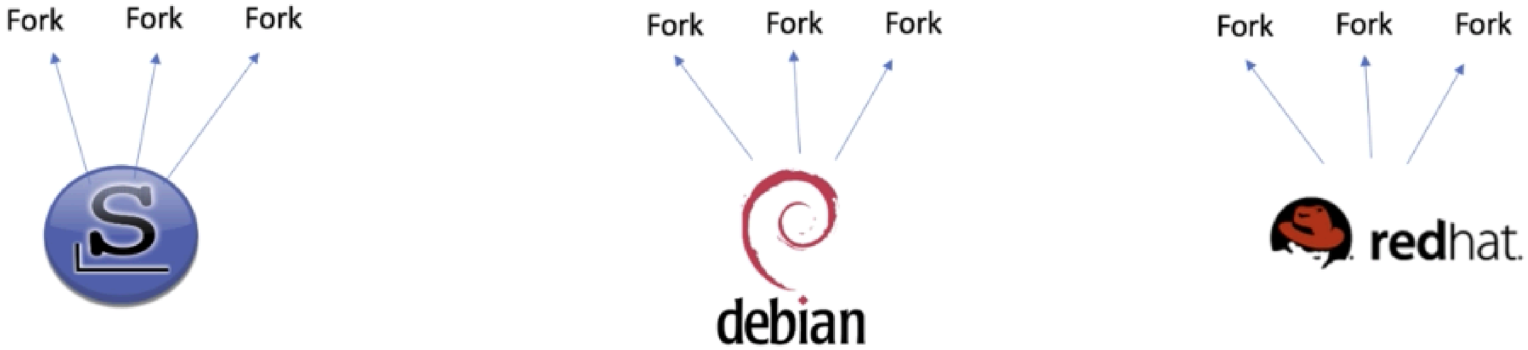
Download and compile all things separately

Great complexity

Linux

You and I had a hard time enjoying all of this

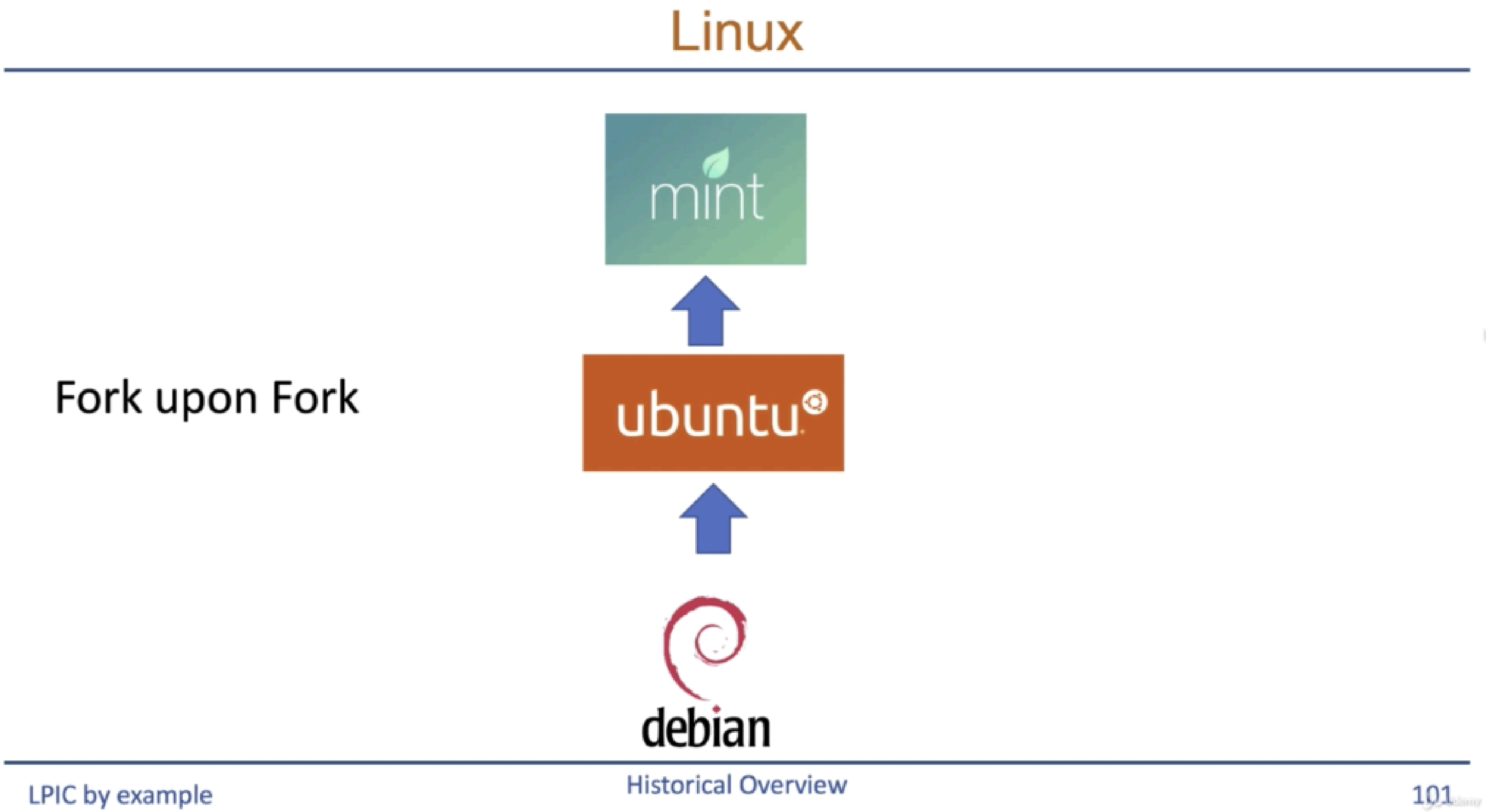
The solution: distributions



Fork Fork Fork

Fork Fork Fork

Fork Fork Fork



Linux

Fork upon Fork

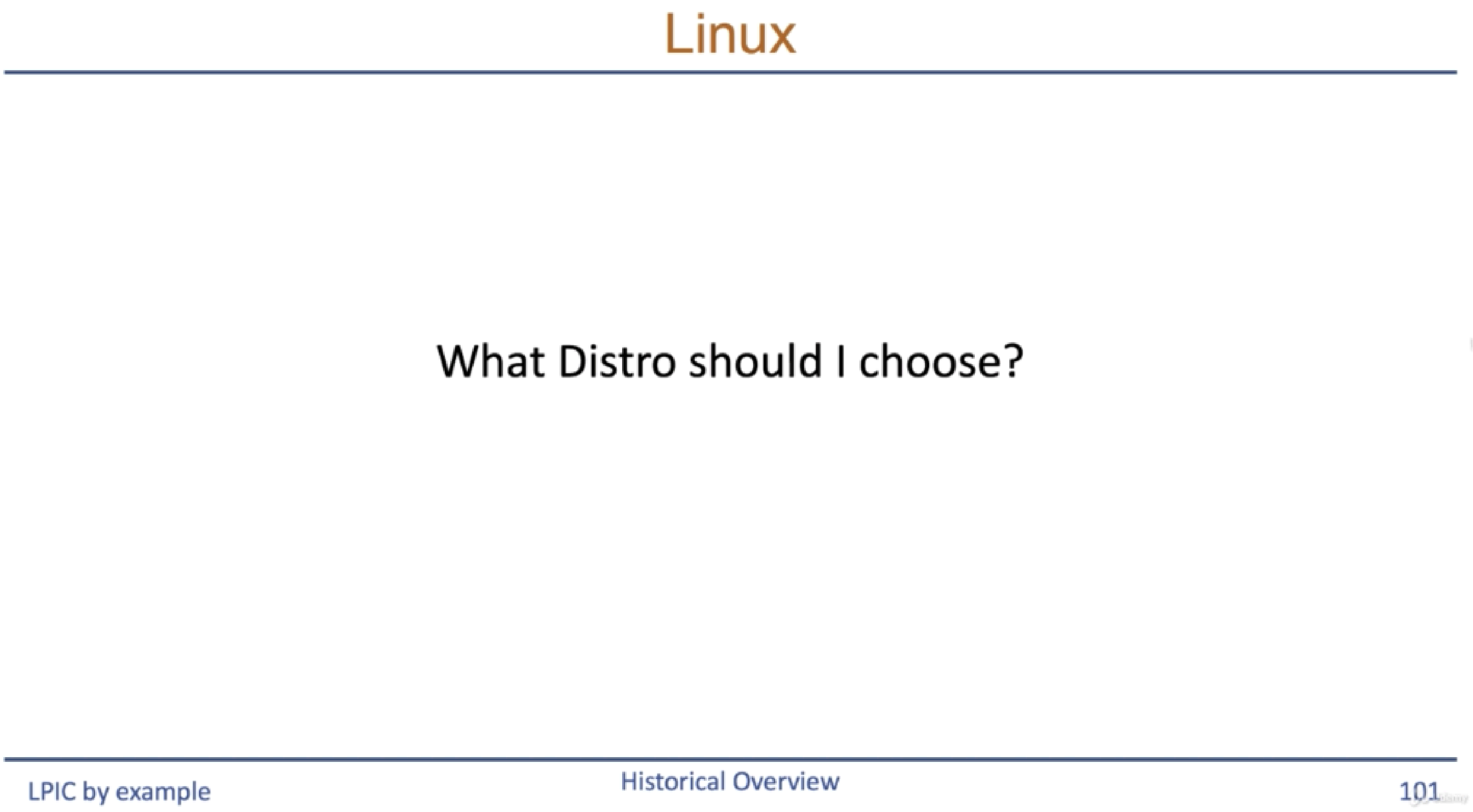
Linux

Open Source

I share a program

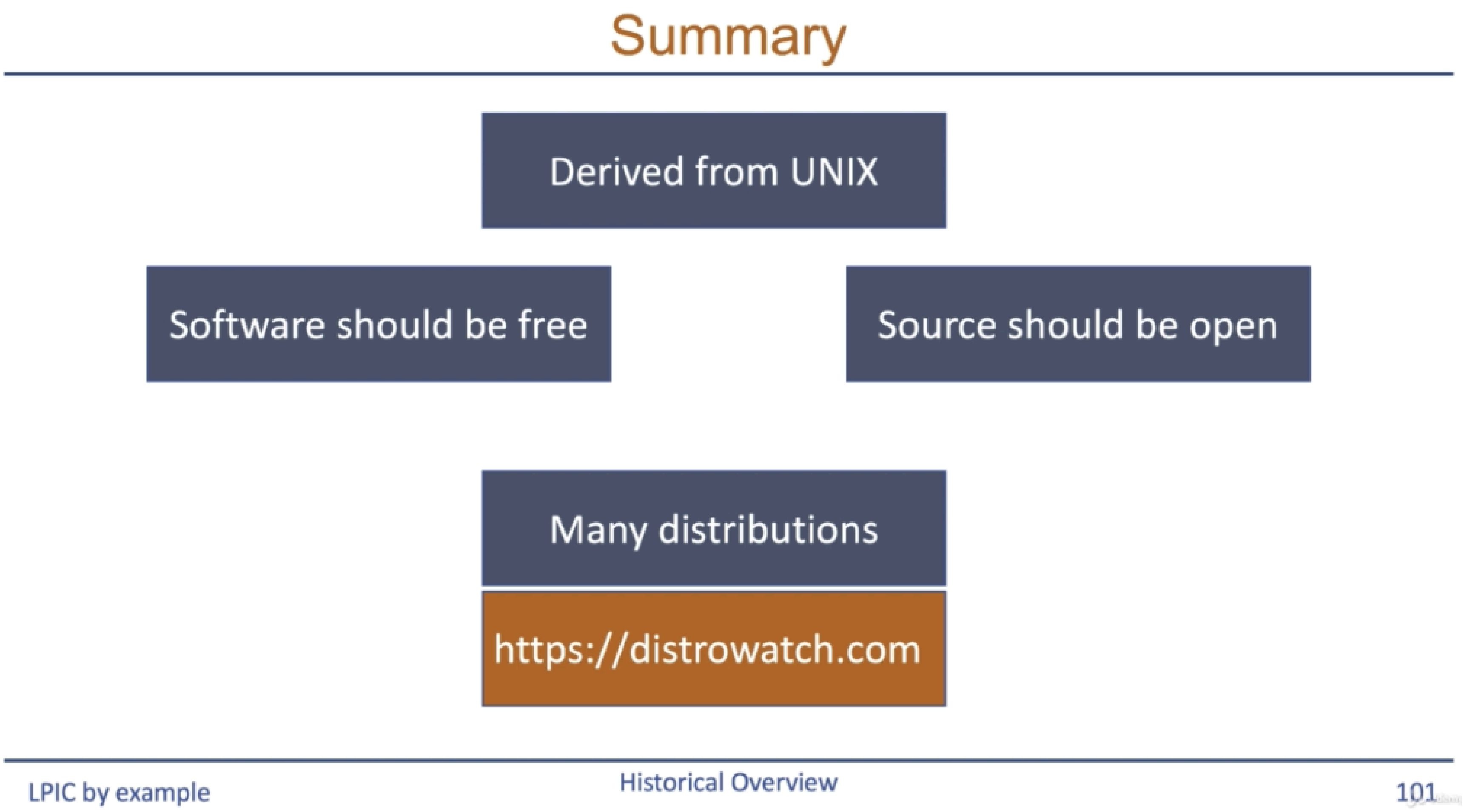
You can modify and share

We all use it



Linux

What Distro should I choose?



Summary

Derived from UNIX

Software should be free

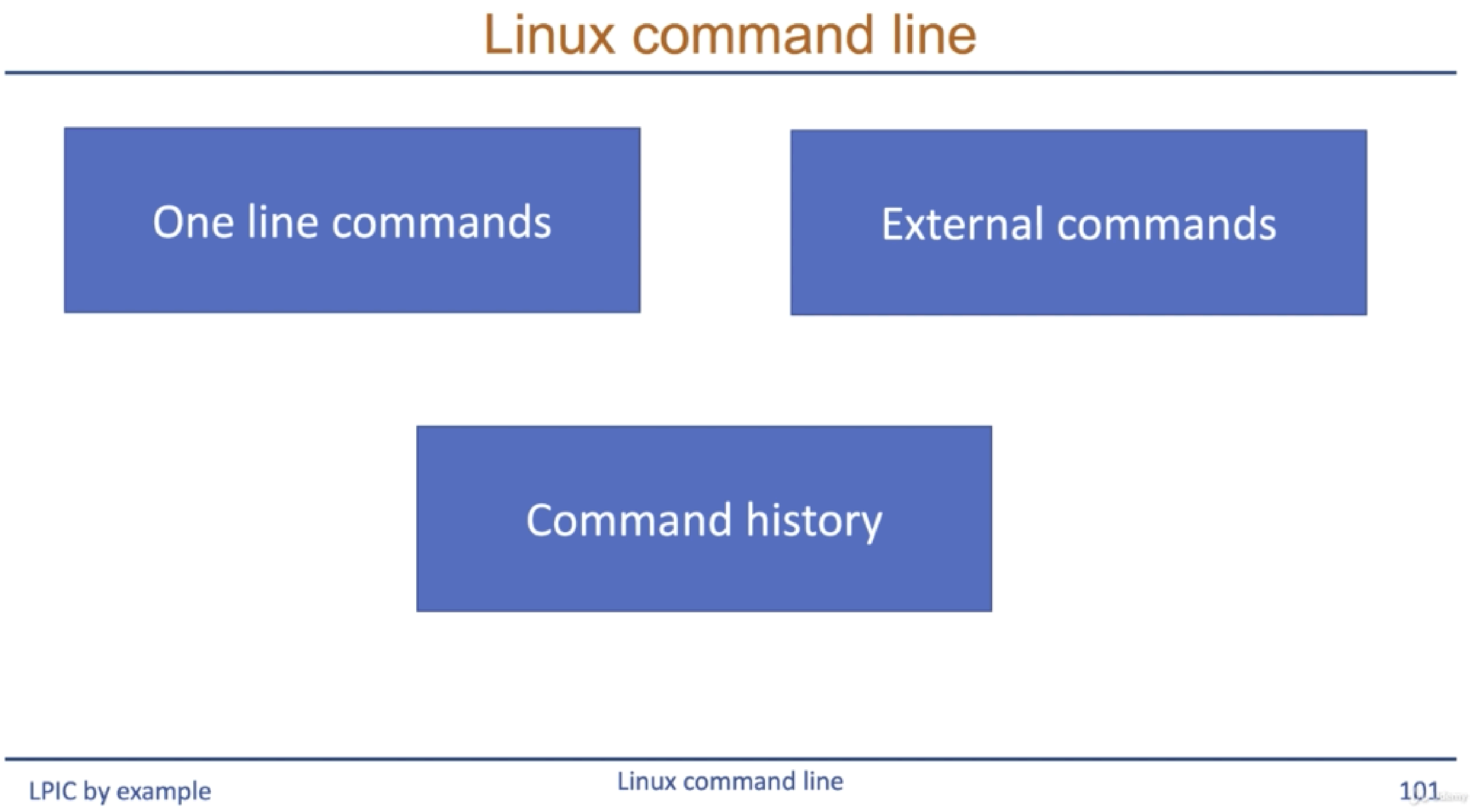
Source should be open

Many distributions

https://distrowatch.com

Linux command line (part 1)

LPIC by example

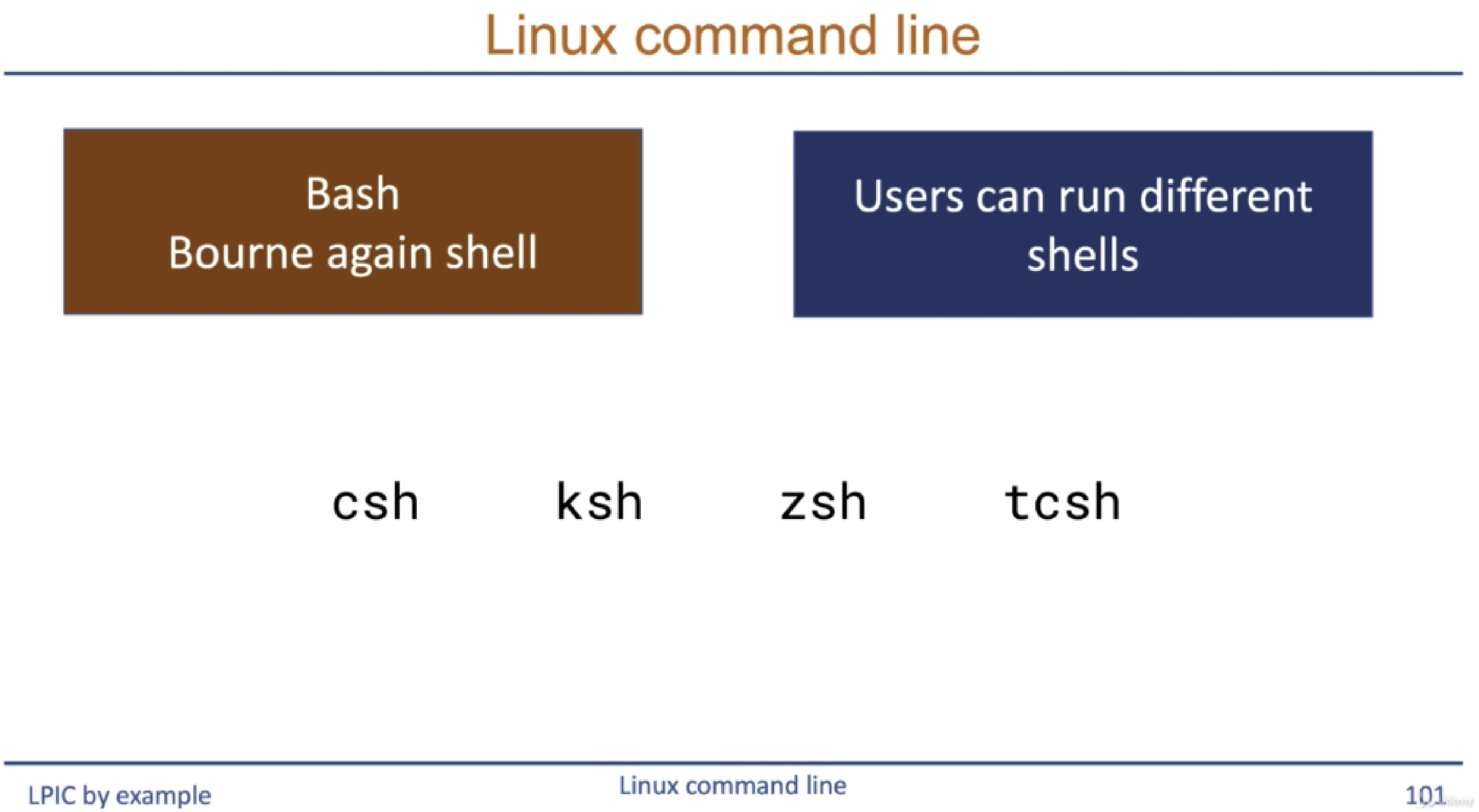


Linux command line

One line commands

External commands

Command history

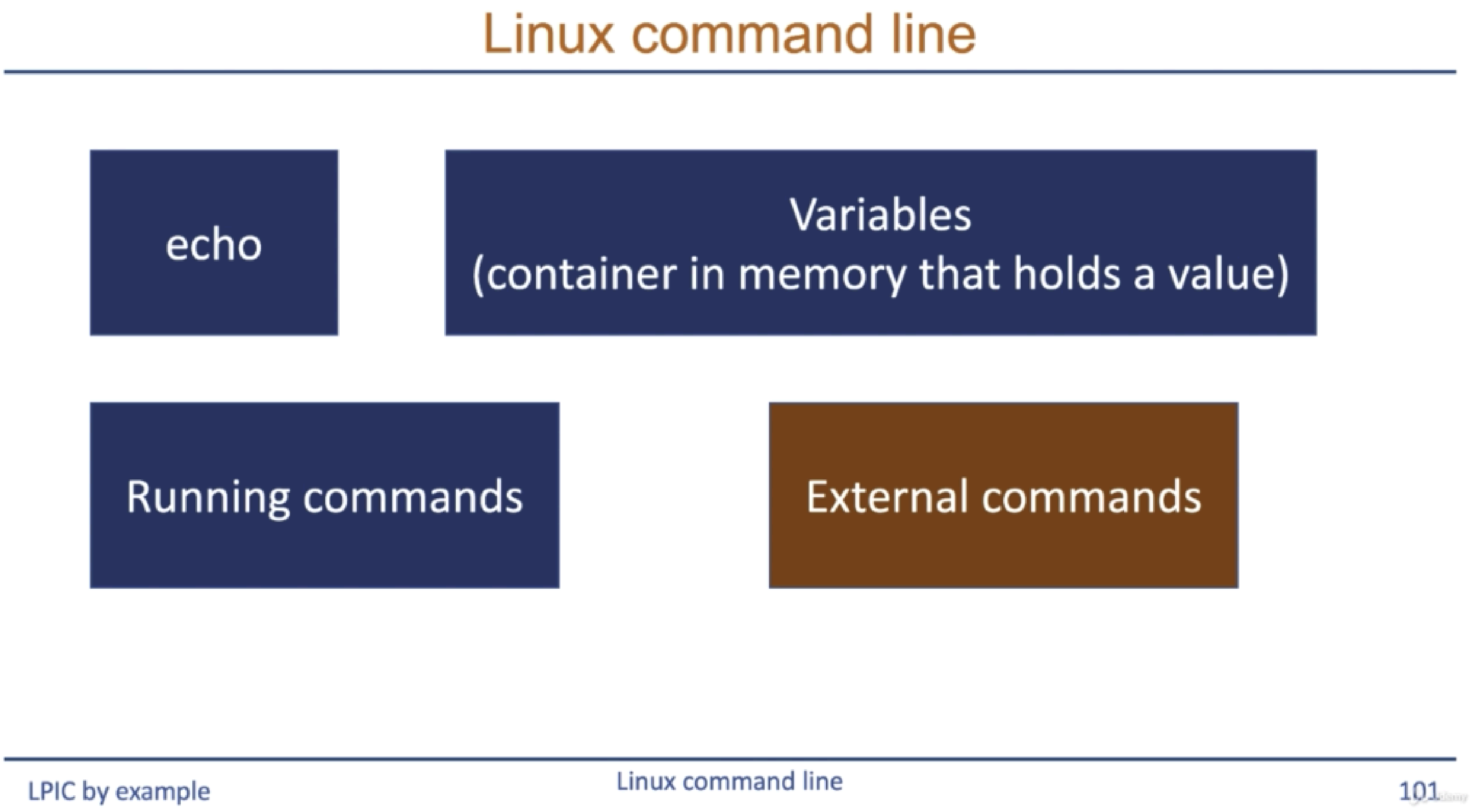


Linux command line

Bash
Bourne again shell

Users can run different shells

csh ksh zsh tcsh



Linux command line

echo

Variables
(container in memory that holds a value)

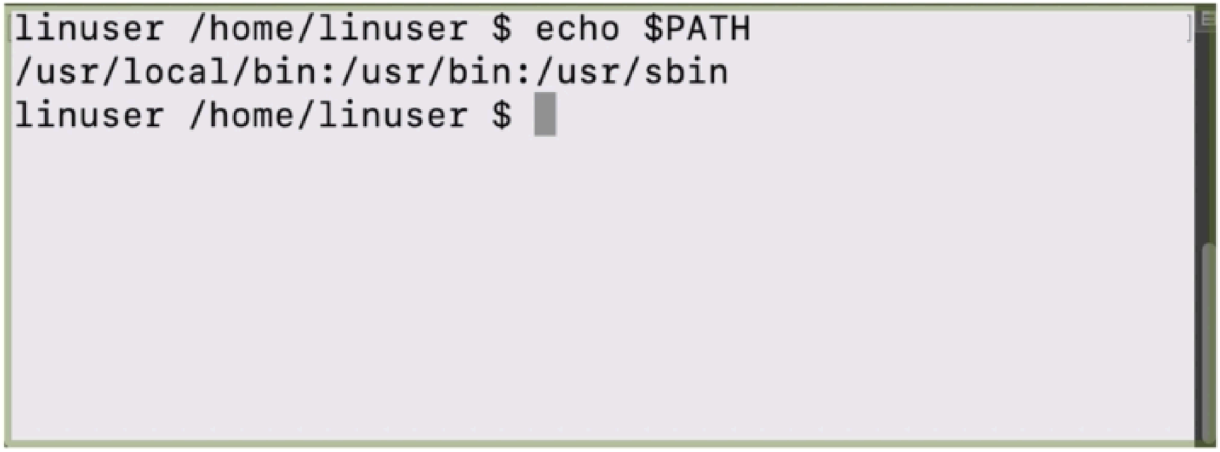
Running commands

External commands

Linux command line

PATH

`directory:directory:directory`

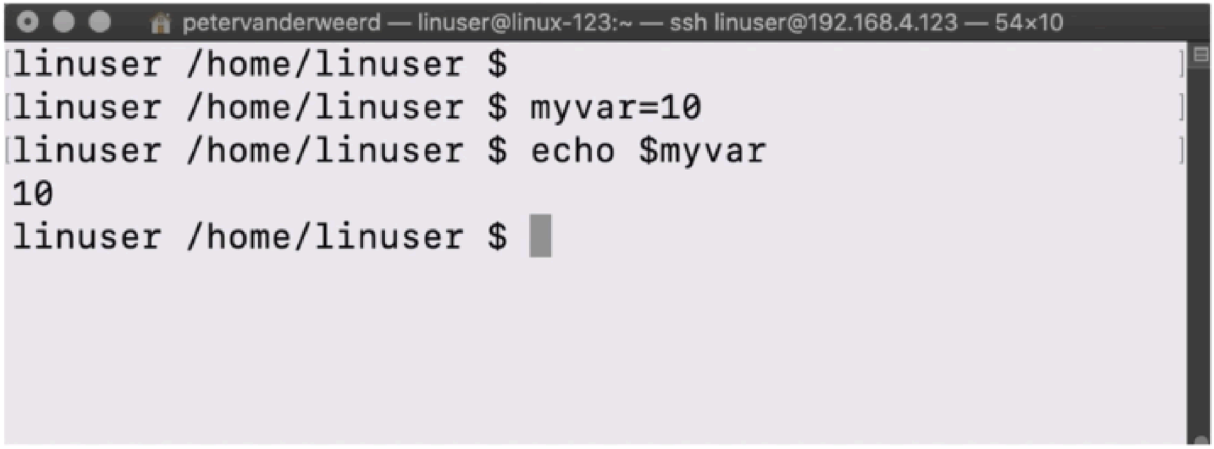


```
linuser /home/linuser $ echo $PATH
/usr/local/bin:/usr/bin:/usr/sbin
linuser /home/linuser $ 
```

Linux command line

Creating a variable

```
var=value
```



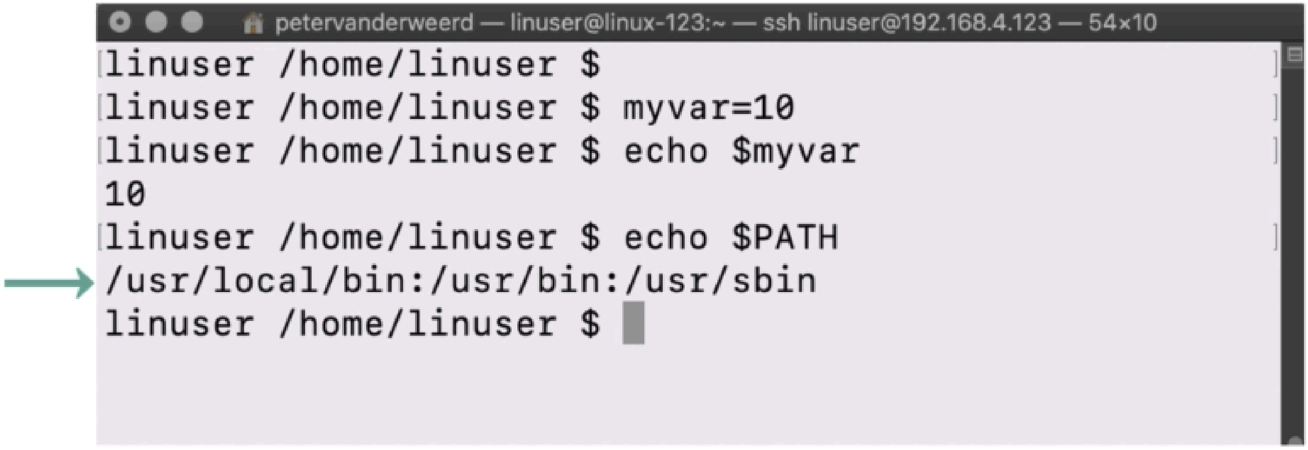
petervanderweerd — linuser@linux-123:~ — ssh linuser@192.168.4.123 — 54×10

```
linuser /home/linuser $
linuser /home/linuser $ myvar=10
linuser /home/linuser $ echo $myvar
10
linuser /home/linuser $
```

Linux command line

Creating a variable

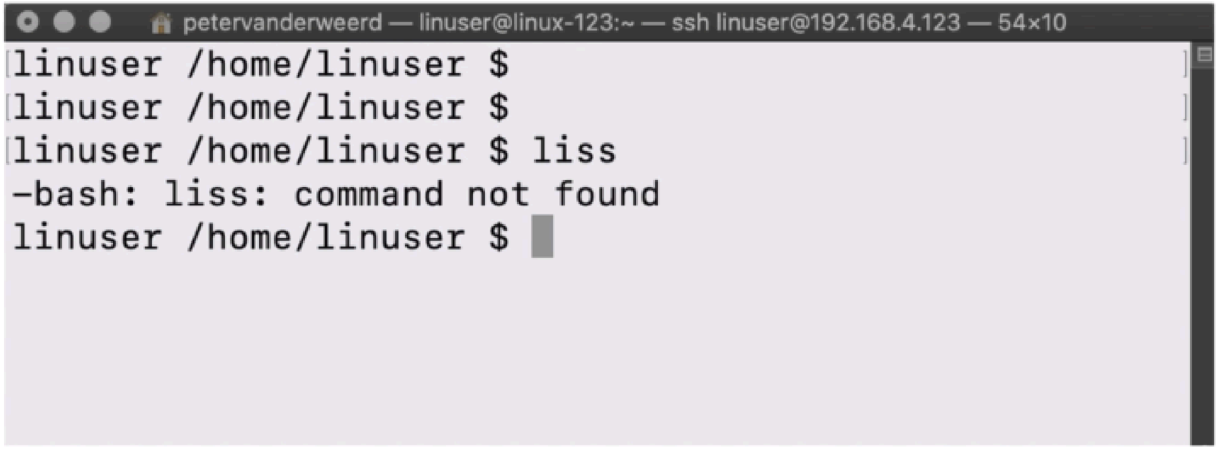
```
var=value
```


petervanderweerd — linuser@linux-123:~ — ssh linuser@192.168.4.123 — 54×10

```
linuser /home/linuser $
linuser /home/linuser $ myvar=10
linuser /home/linuser $ echo $myvar
10
linuser /home/linuser $ echo $PATH
/usr/local/bin:/usr/bin:/usr/sbin
linuser /home/linuser $
```

Linux command line

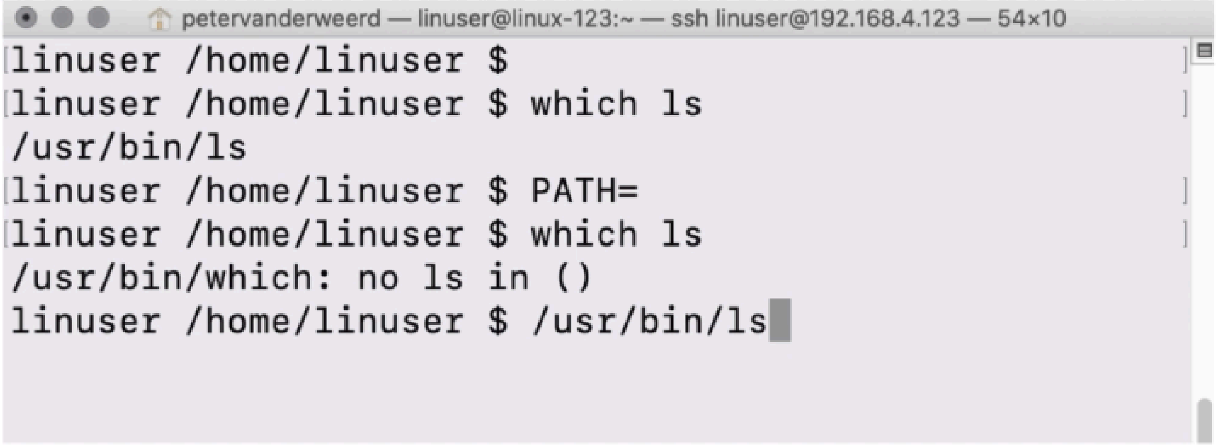
Command not found


petervanderweerd — linuser@linux-123:~ — ssh linuser@192.168.4.123 — 54×10

```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ liss
-bash: liss: command not found
linuser /home/linuser $
```

Linux command line

which



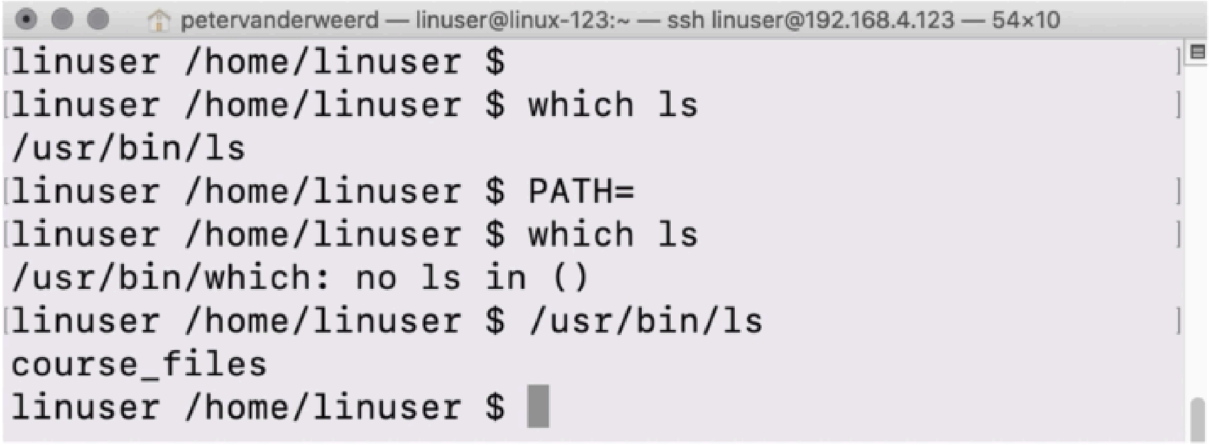
petervanderweerd — linuser@linux-123:~ — ssh linuser@192.168.4.123 — 54×10

```
linuser /home/linuser $
linuser /home/linuser $ which ls
/usr/bin/ls
linuser /home/linuser $ PATH=
linuser /home/linuser $ which ls
/usr/bin/which: no ls in ()
linuser /home/linuser $ /usr/bin/ls
```

Linux command line

type

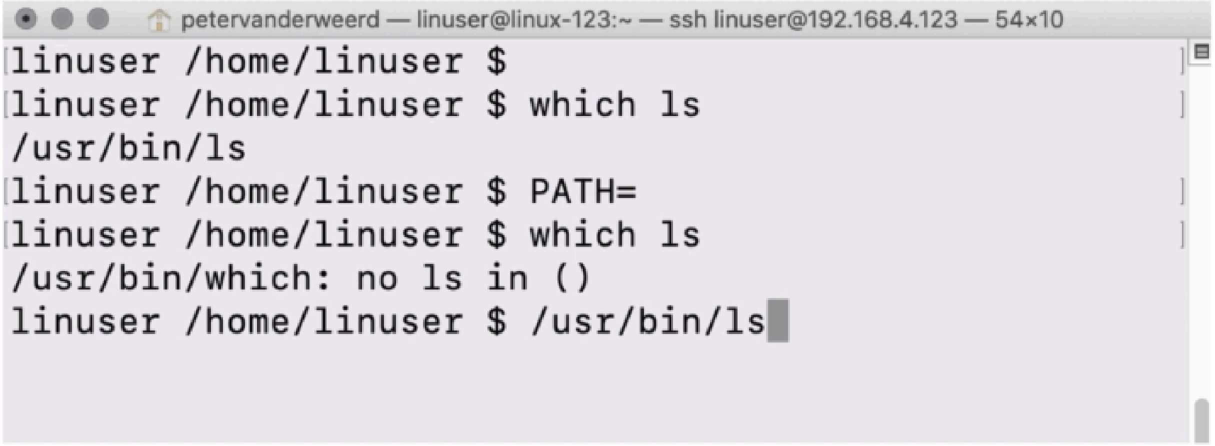
Internal command


petervanderweerd — linuser@linux-123:~ — ssh linuser@192.168.4.123 — 54×10

```
linuser /home/linuser $
linuser /home/linuser $ which ls
/usr/bin/ls
linuser /home/linuser $ PATH=
linuser /home/linuser $ which ls
/usr/bin/which: no ls in ()
linuser /home/linuser $ /usr/bin/ls
course_files
linuser /home/linuser $
```

Linux command line

which



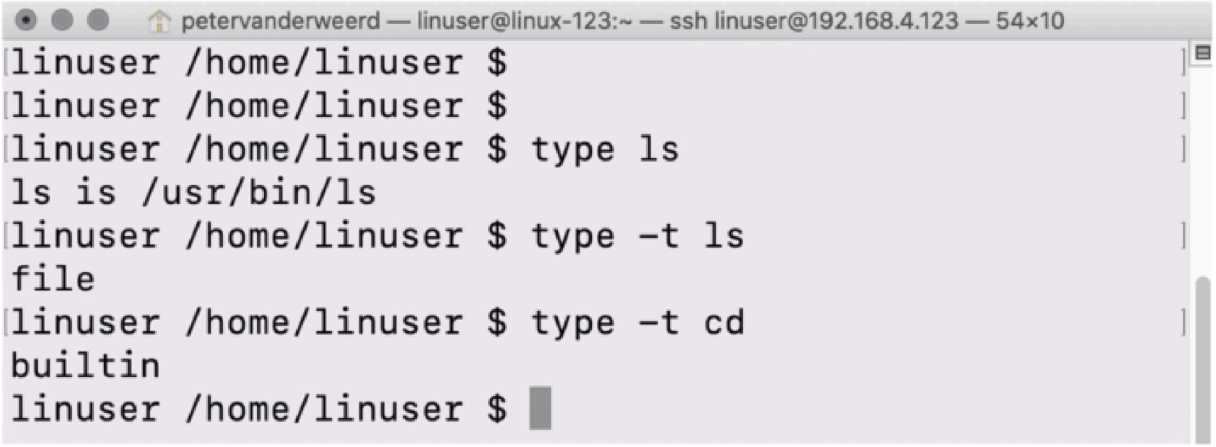
petervanderweerd — linuser@linux-123:~ — ssh linuser@192.168.4.123 — 54×10

```
linuser /home/linuser $
linuser /home/linuser $ which ls
/usr/bin/ls
linuser /home/linuser $ PATH=
linuser /home/linuser $ which ls
/usr/bin/which: no ls in ()
linuser /home/linuser $ /usr/bin/ls
```

Linux command line

type

Internal command

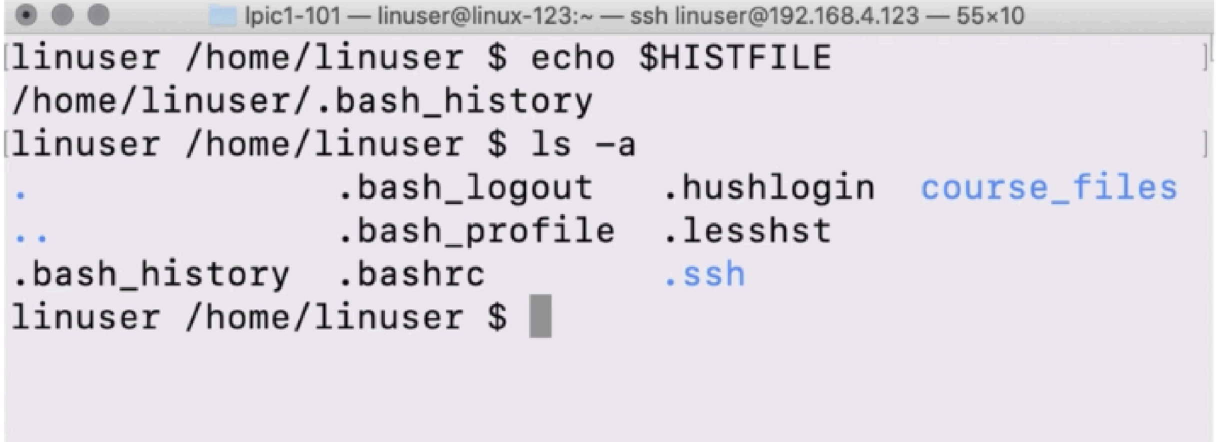

petervanderweerd — linuser@linux-123:~ — ssh linuser@192.168.4.123 — 54×10

```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ type ls
ls is /usr/bin/ls
linuser /home/linuser $ type -t ls
file
linuser /home/linuser $ type -t cd
builtin
linuser /home/linuser $ 
```

Linux command line

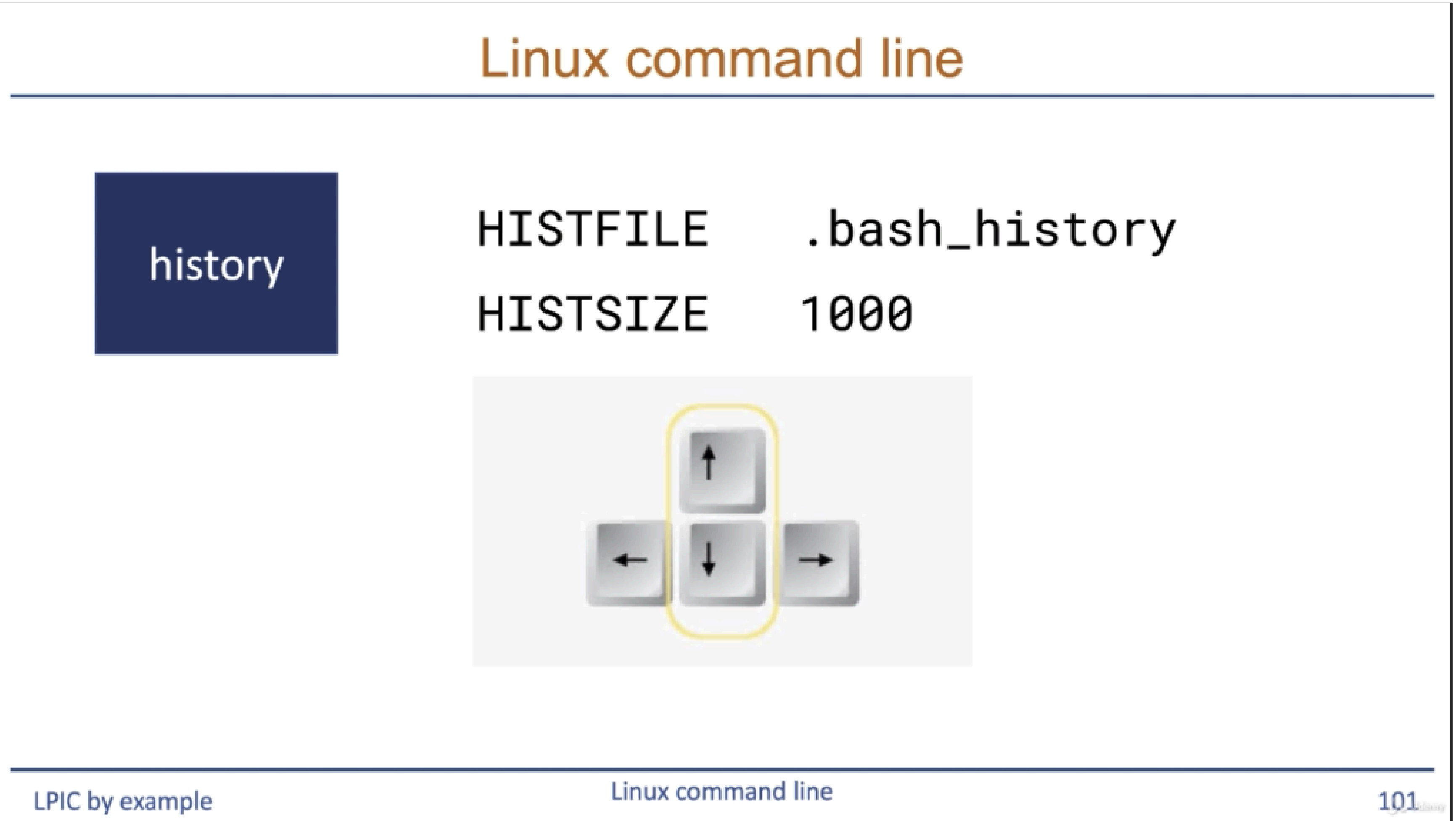
history

HISTFILE .bash_history



lpic1-101 — linuser@linux-123:~ — ssh linuser@192.168.4.123 — 55×10

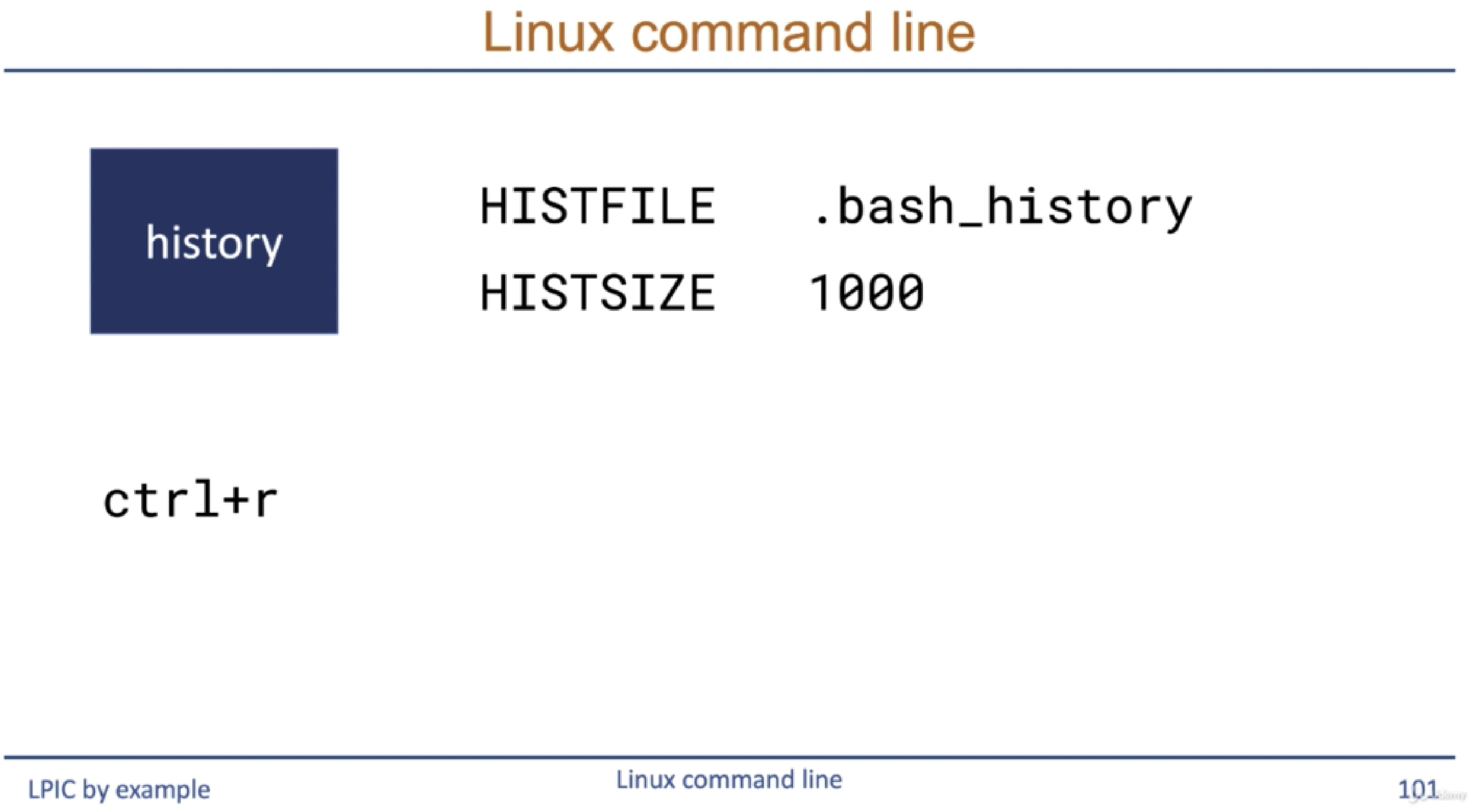
```
linuser /home/linuser $ echo $HISTFILE
/home/linuser/.bash_history
linuser /home/linuser $ ls -a
.               .bash_logout   .hushlogin  course_files
..              .bash_profile  .lesshst
.bash_history   .bashrc        .ssh
linuser /home/linuser $ 
```



Linux command line

history

```
HISTFILE   .bash_history
HISTSIZE   1000
```

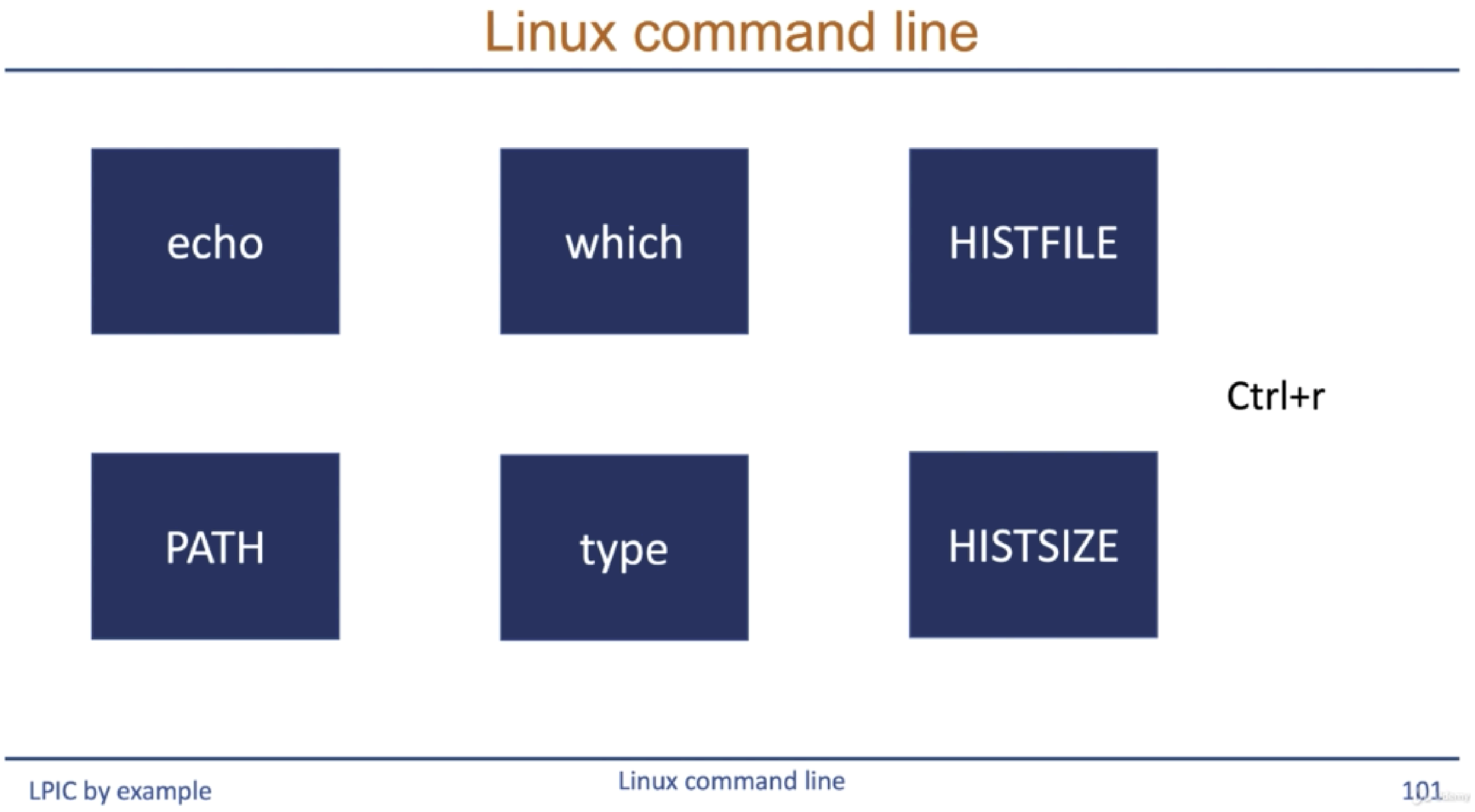


Linux command line

history

```
HISTFILE   .bash_history
HISTSIZE   1000
```

```
ctrl+r
```



Linux command line

echo

which

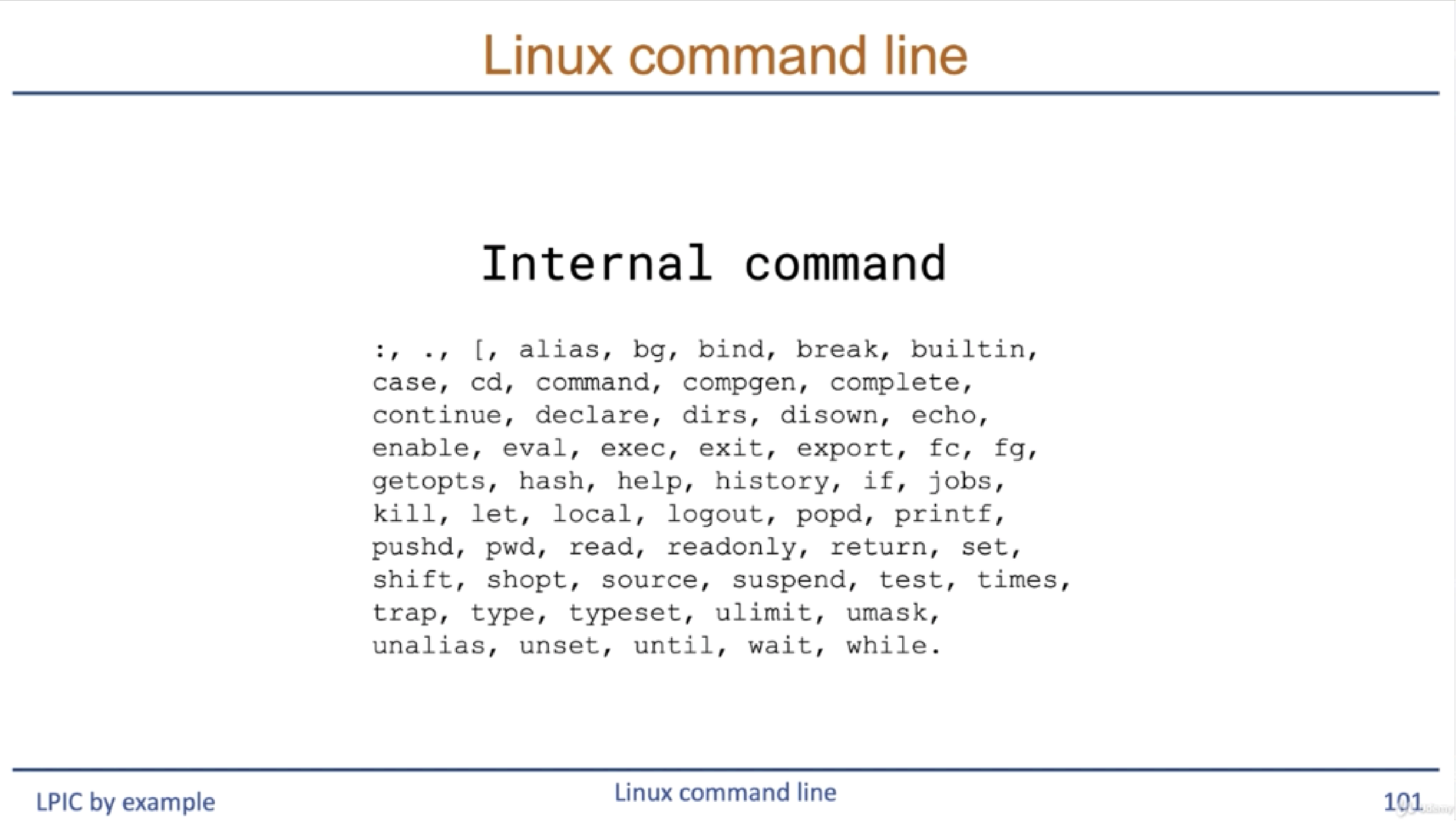
HISTFILE

Ctrl+r

PATH

type

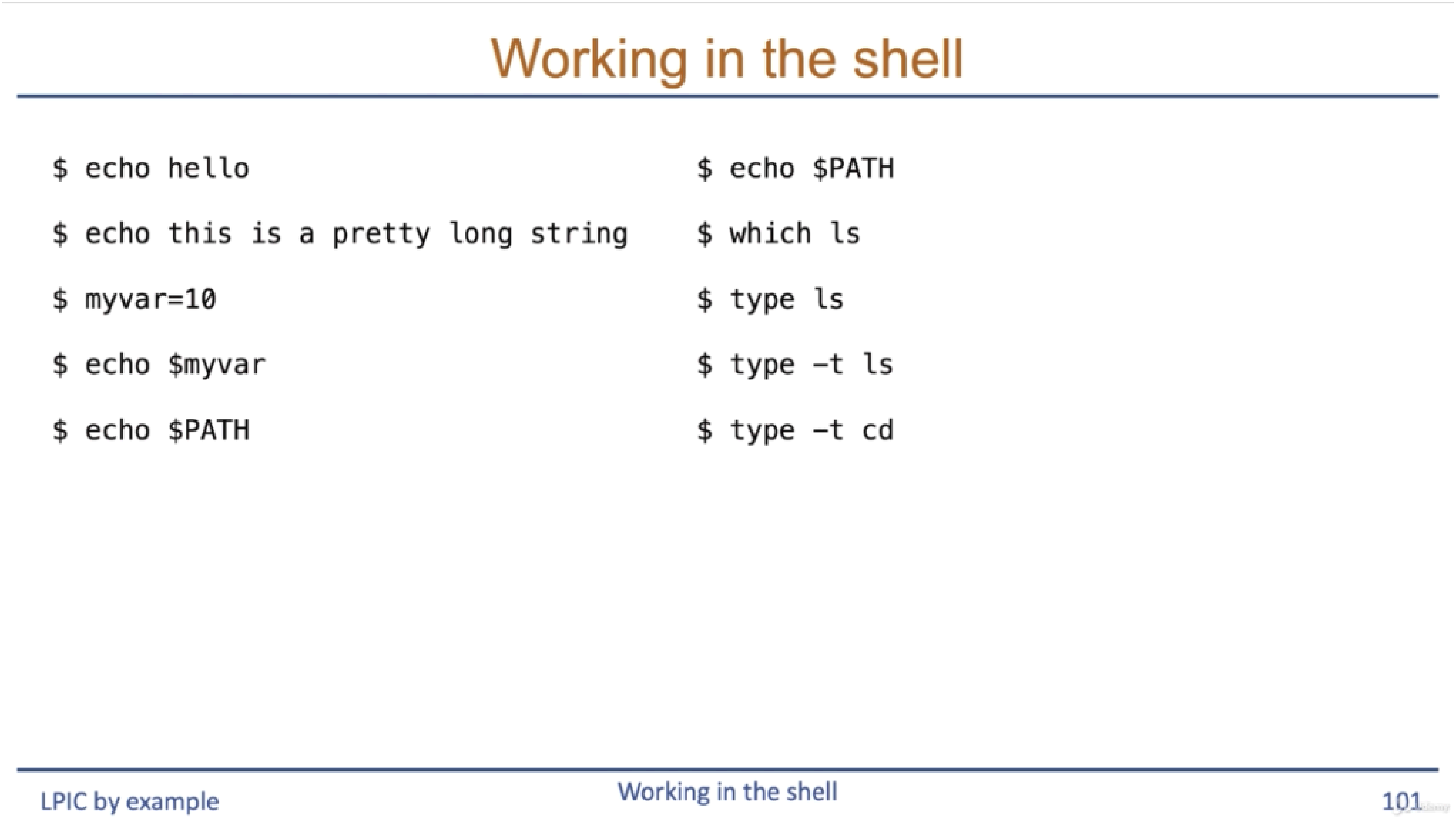
HISTSIZE



Linux command line

Internal command

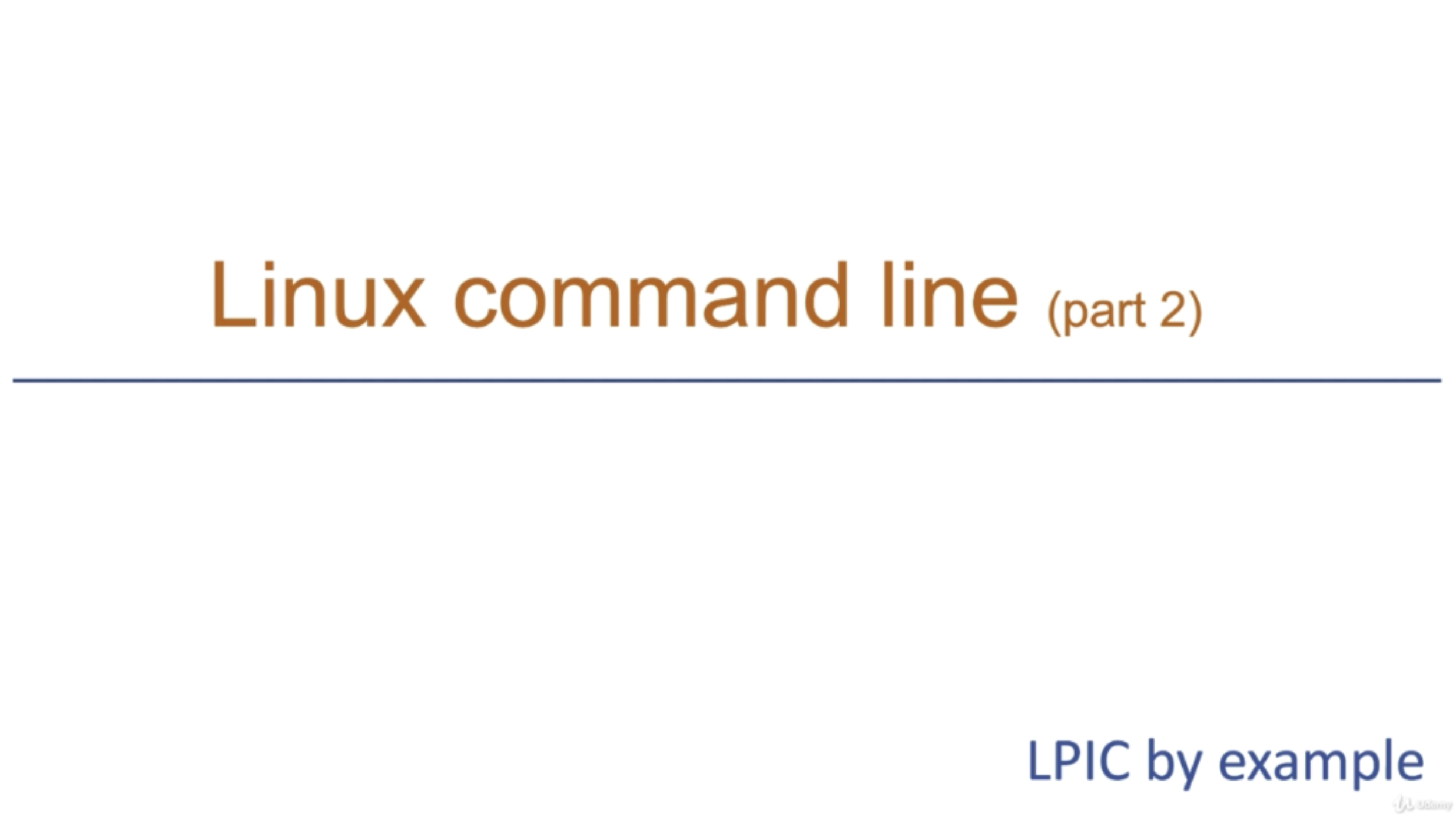
```
:, ., [, alias, bg, bind, break, builtin,
case, cd, command, compgen, complete,
continue, declare, dirs, disown, echo,
enable, eval, exec, exit, export, fc, fg,
getopts, hash, help, history, if, jobs,
kill, let, local, logout, popd, printf,
pushd, pwd, read, readonly, return, set,
shift, shopt, source, suspend, test, times,
trap, type, typeset, ulimit, umask,
unalias, unset, until, wait, while.
```



Working in the shell

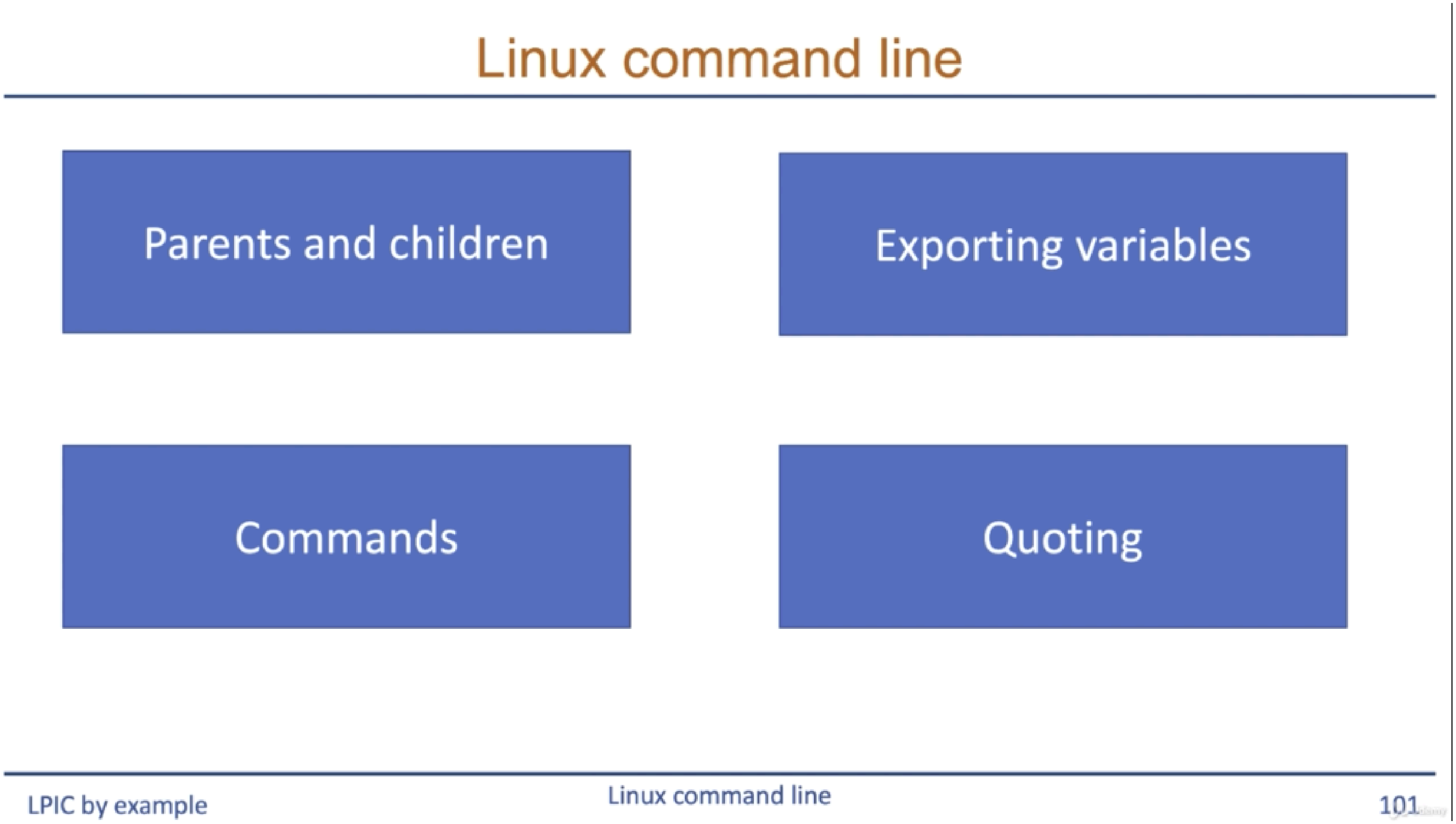
```
$ echo hello

$ echo this is a pretty long string

$ myvar=10

$ echo $myvar

$ echo $PATH
```

```
$ echo $PATH

$ which ls

$ type ls

$ type -t ls

$ type -t cd
```



Linux command line (part 2)

LPIC by example



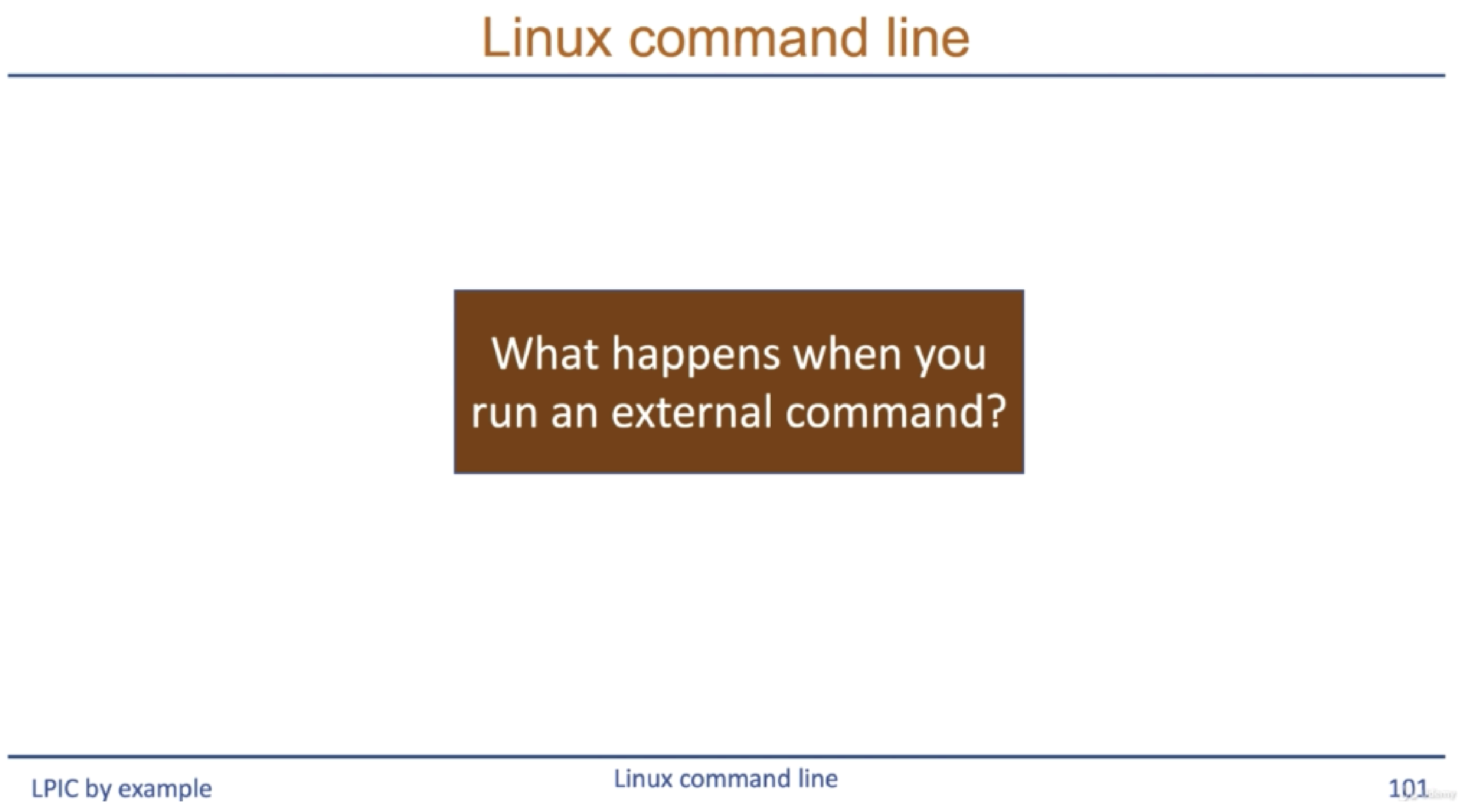
Linux command line

Parents and children

Exporting variables

Commands

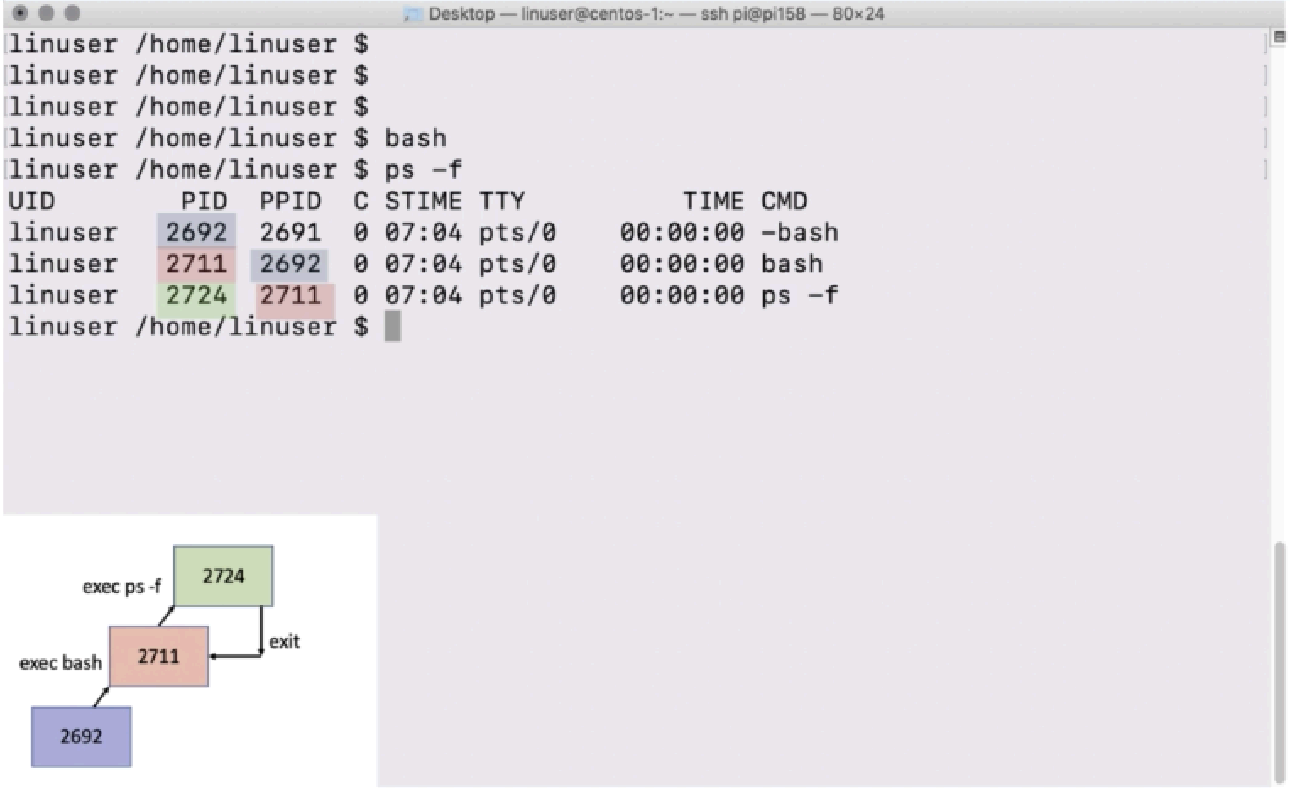
Quoting



Linux command line

What happens when you run an external command?

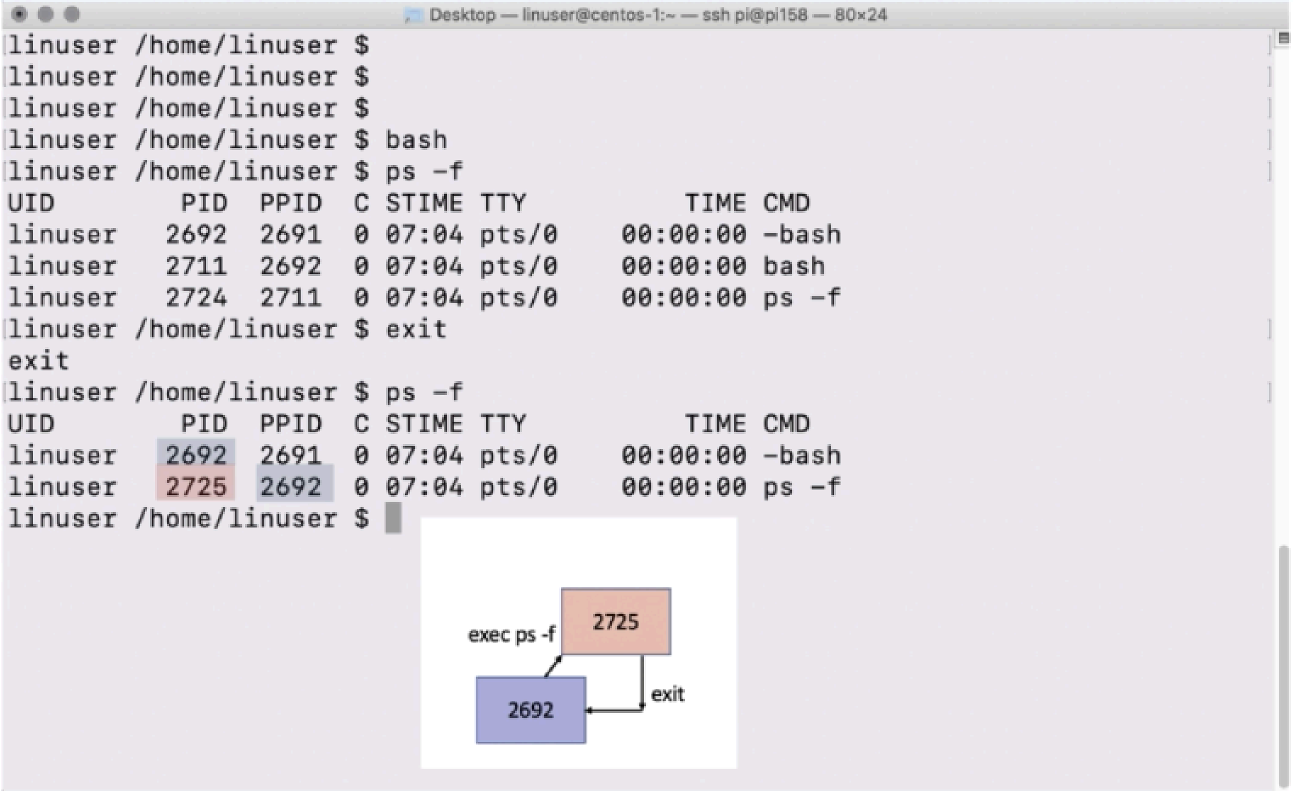
Linux command line



Desktop — linuser@centos-1:~ — ssh pi@pi158 — 80×24

```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ bash
linuser /home/linuser $ ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
linuser     2692  2691  0 07:04 pts/0    00:00:00 -bash
linuser     2711  2692  0 07:04 pts/0    00:00:00 bash
linuser     2724  2711  0 07:04 pts/0    00:00:00 ps -f
linuser /home/linuser $
```

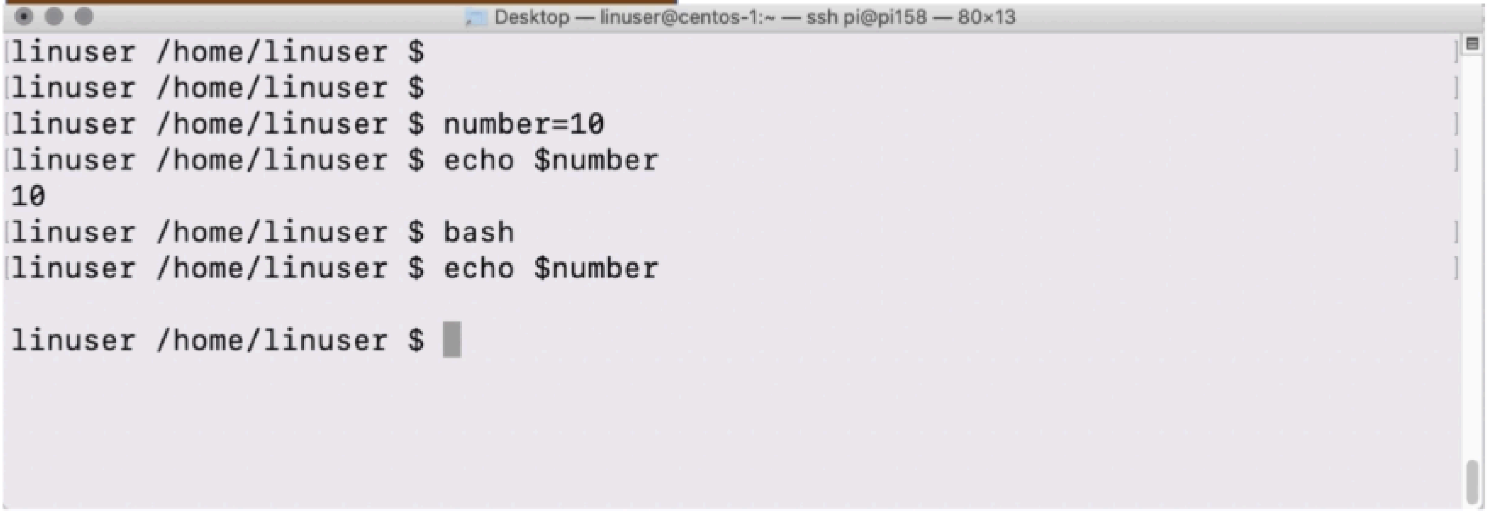
Linux command line



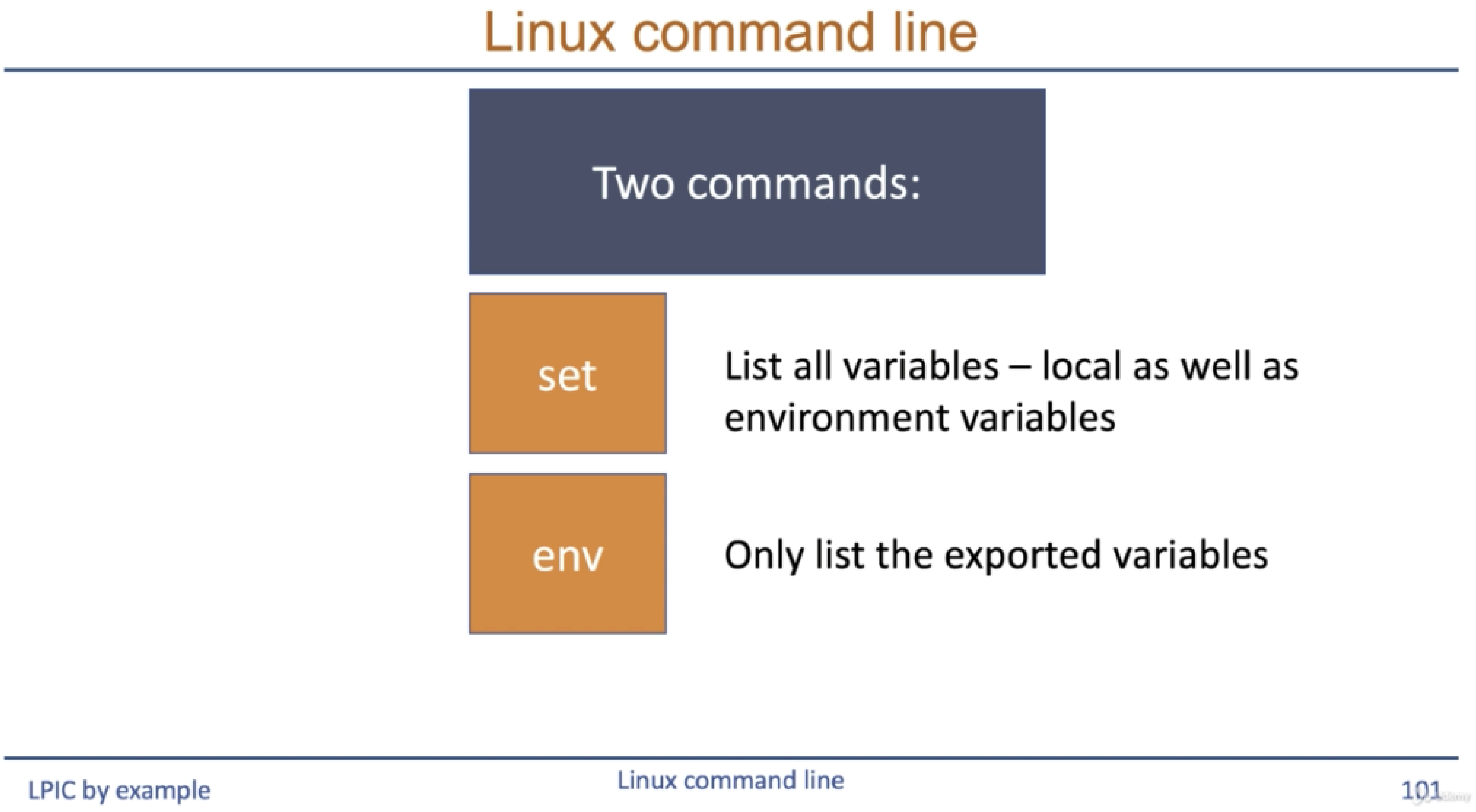
```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ bash
linuser /home/linuser $ ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
linuser     2692  2691  0 07:04 pts/0    00:00:00 -bash
linuser     2711  2692  0 07:04 pts/0    00:00:00 bash
linuser     2724  2711  0 07:04 pts/0    00:00:00 ps -f
linuser /home/linuser $ exit
exit
linuser /home/linuser $ ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
linuser     2692  2691  0 07:04 pts/0    00:00:00 -bash
linuser     2725  2692  0 07:04 pts/0    00:00:00 ps -f
linuser /home/linuser $ 
```

Linux command line

Variables are local to the shell by default



```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ number=10
linuser /home/linuser $ echo $number
10
linuser /home/linuser $ bash
linuser /home/linuser $ echo $number

linuser /home/linuser $
```



Linux command line

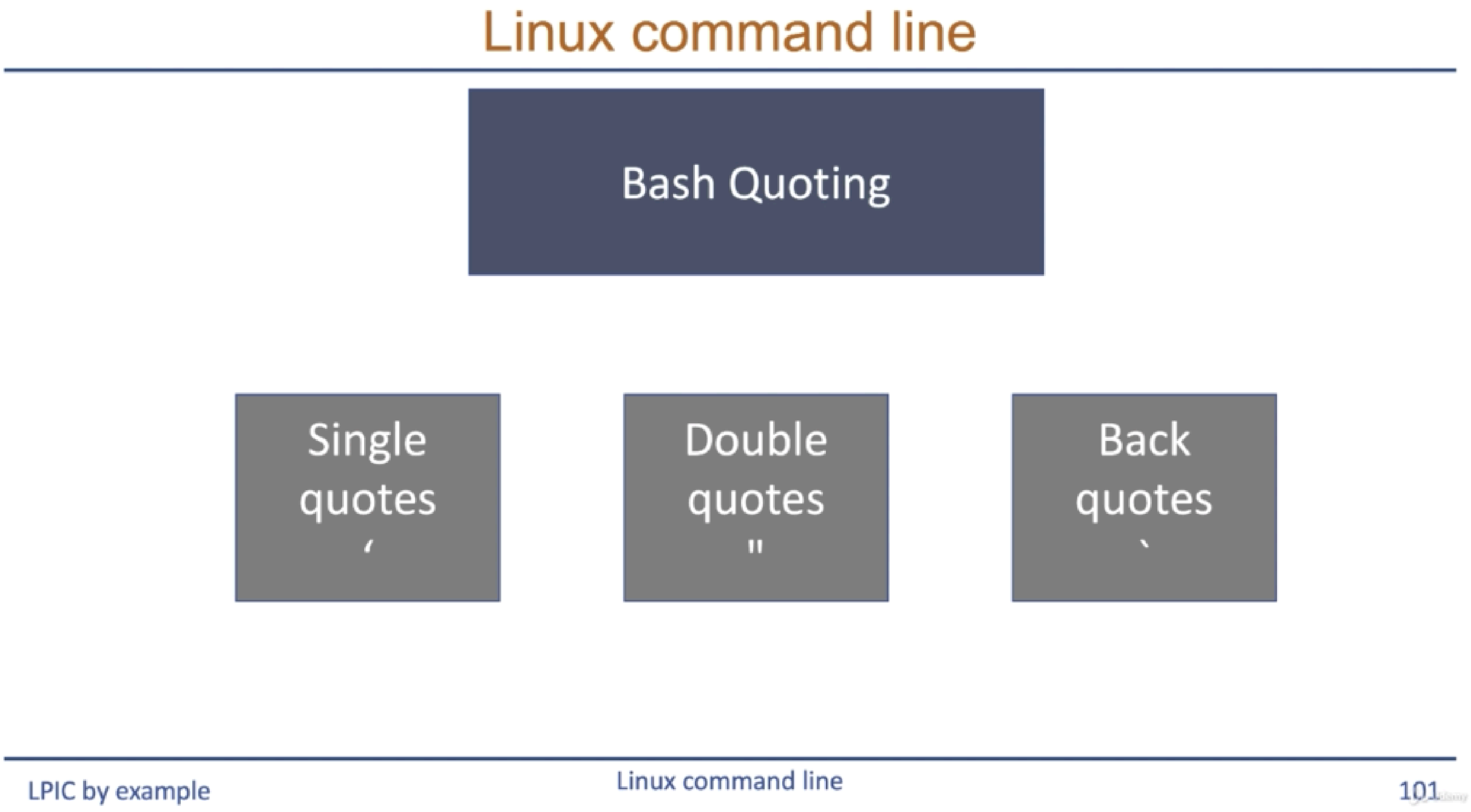
Two commands:

set

List all variables – local as well as environment variables

env

Only list the exported variables



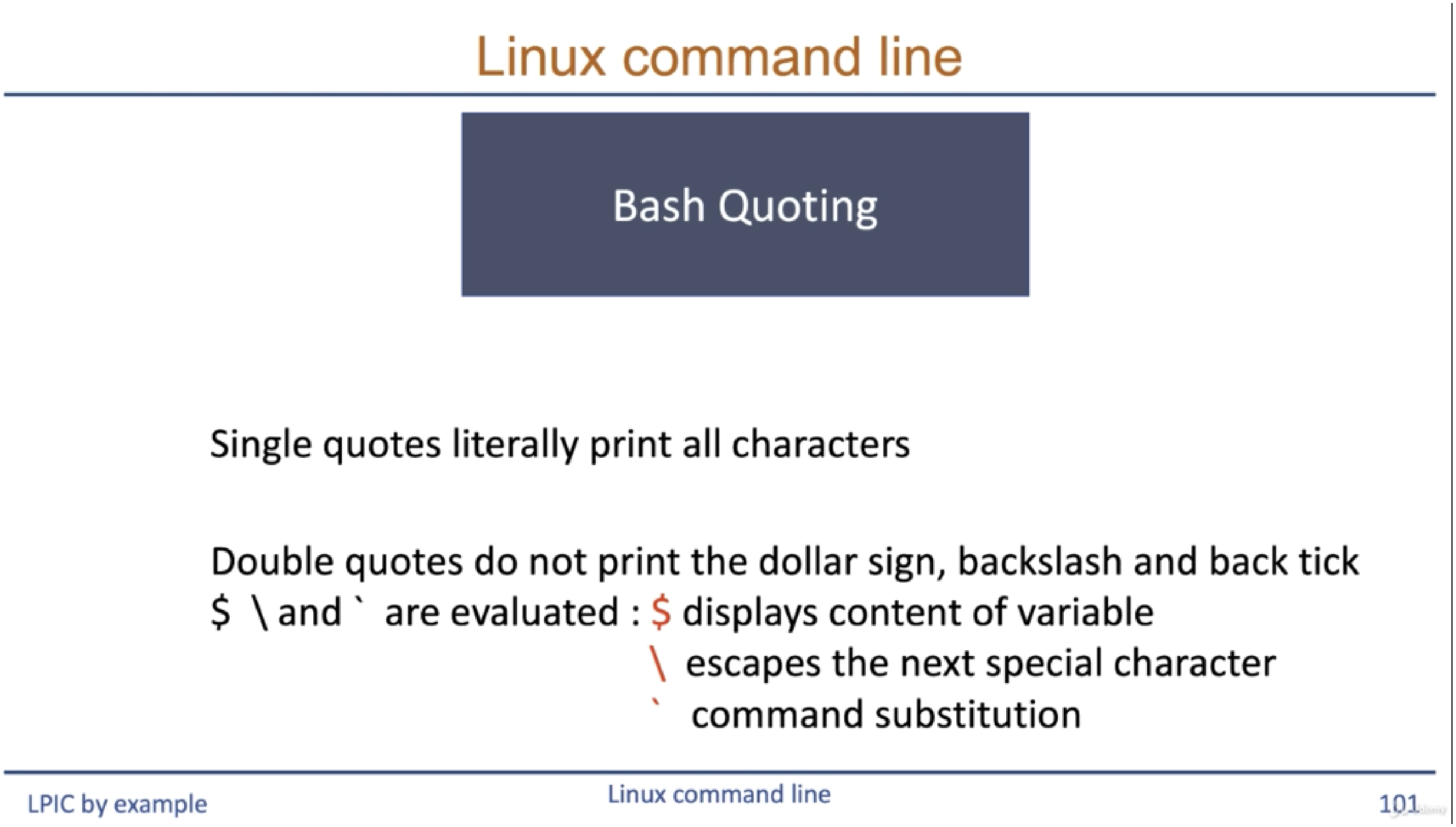
Linux command line

Bash Quoting

Single quotes
‘

Double quotes
"

Back quotes
`

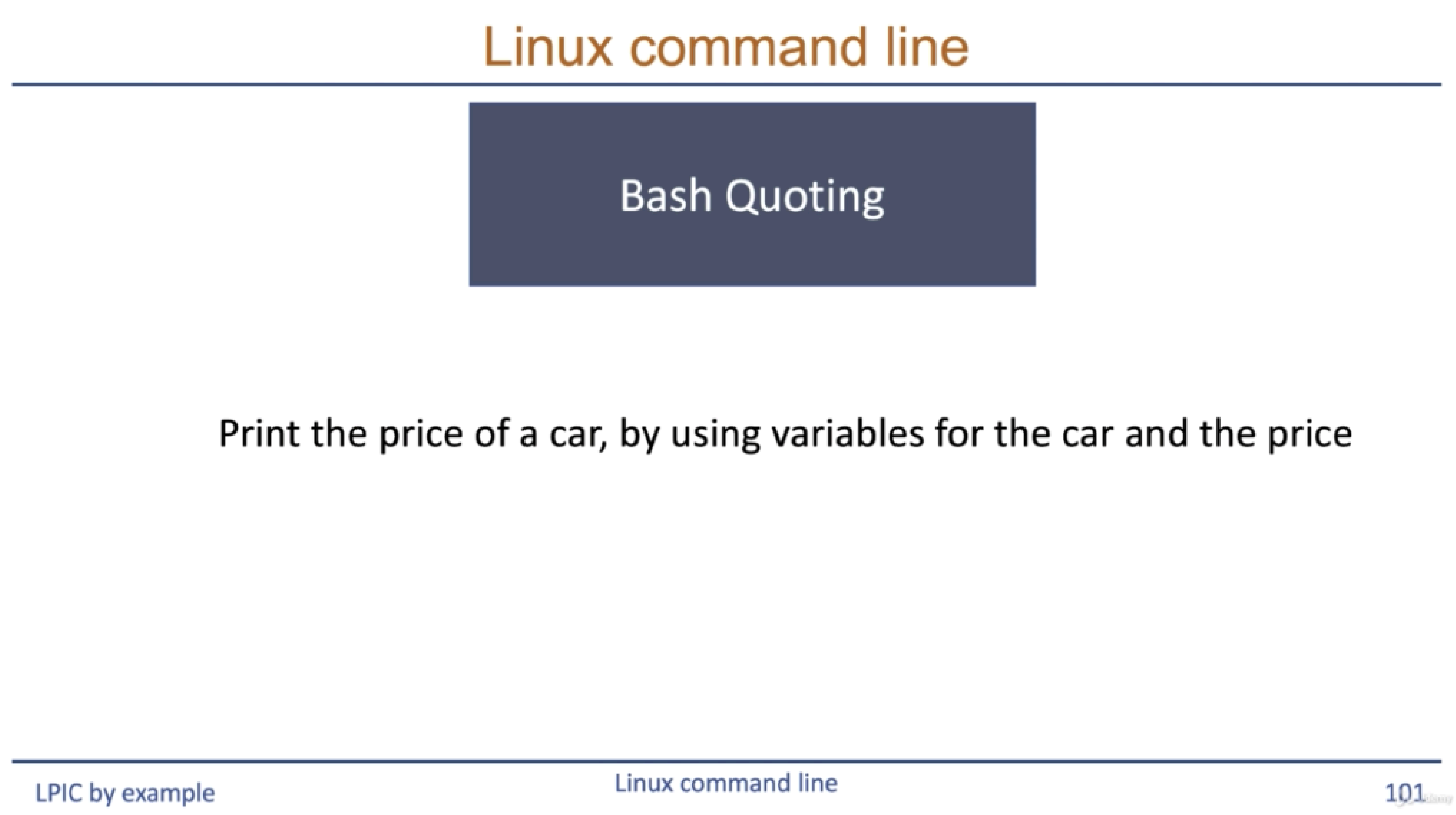


Linux command line

Bash Quoting

Single quotes literally print all characters

Double quotes do not print the dollar sign, backslash and back tick
$ \ and ` are evaluated : $ displays content of variable
\ escapes the next special character
` command substitution

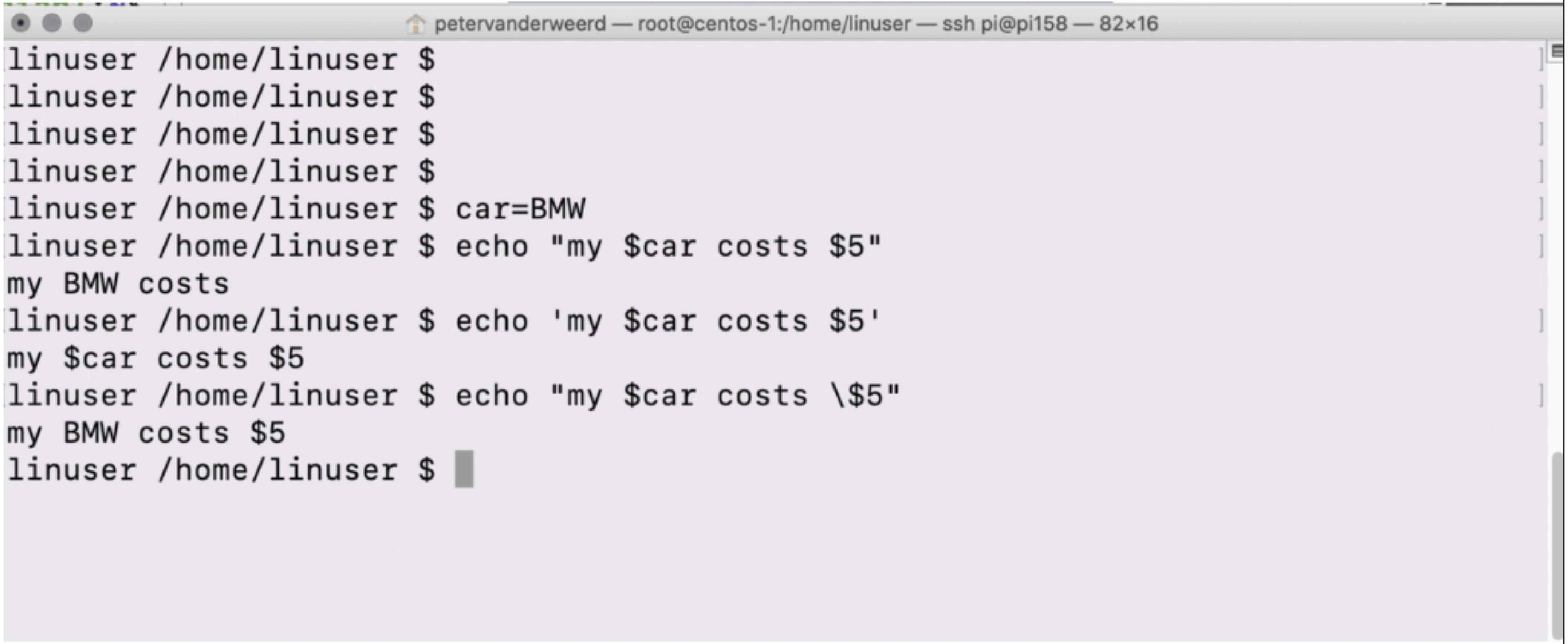


Linux command line

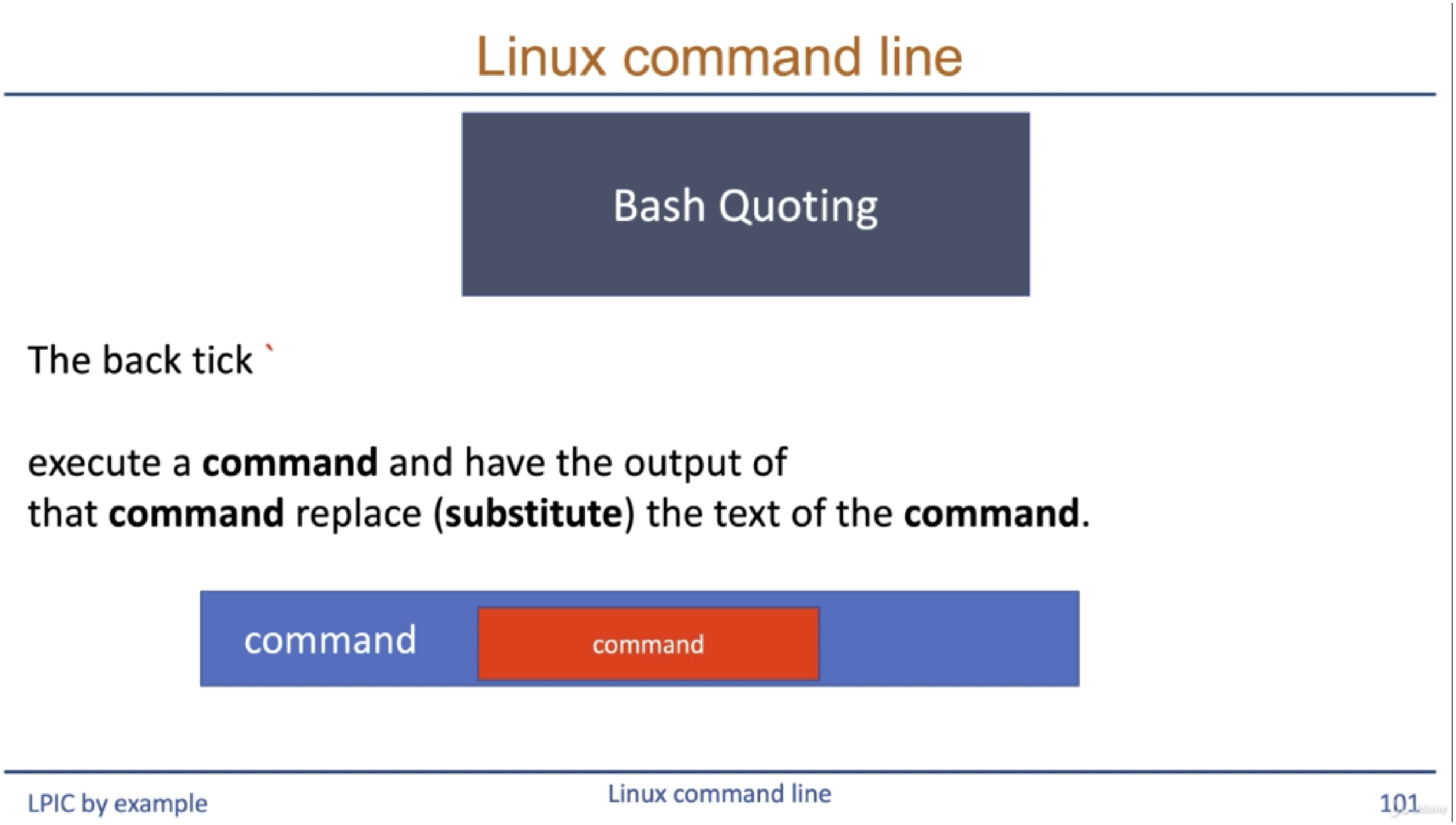
Bash Quoting

Print the price of a car, by using variables for the car and the price

Linux command line


petervanderweerd — root@centos-1:/home/linuser — ssh pi@pi158 — 82×16

```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ car=BMW
linuser /home/linuser $ echo "my $car costs $5"
my BMW costs
linuser /home/linuser $ echo 'my $car costs $5'
my $car costs $5
linuser /home/linuser $ echo "my $car costs \$5"
my BMW costs $5
linuser /home/linuser $
```



Linux command line

Bash Quoting

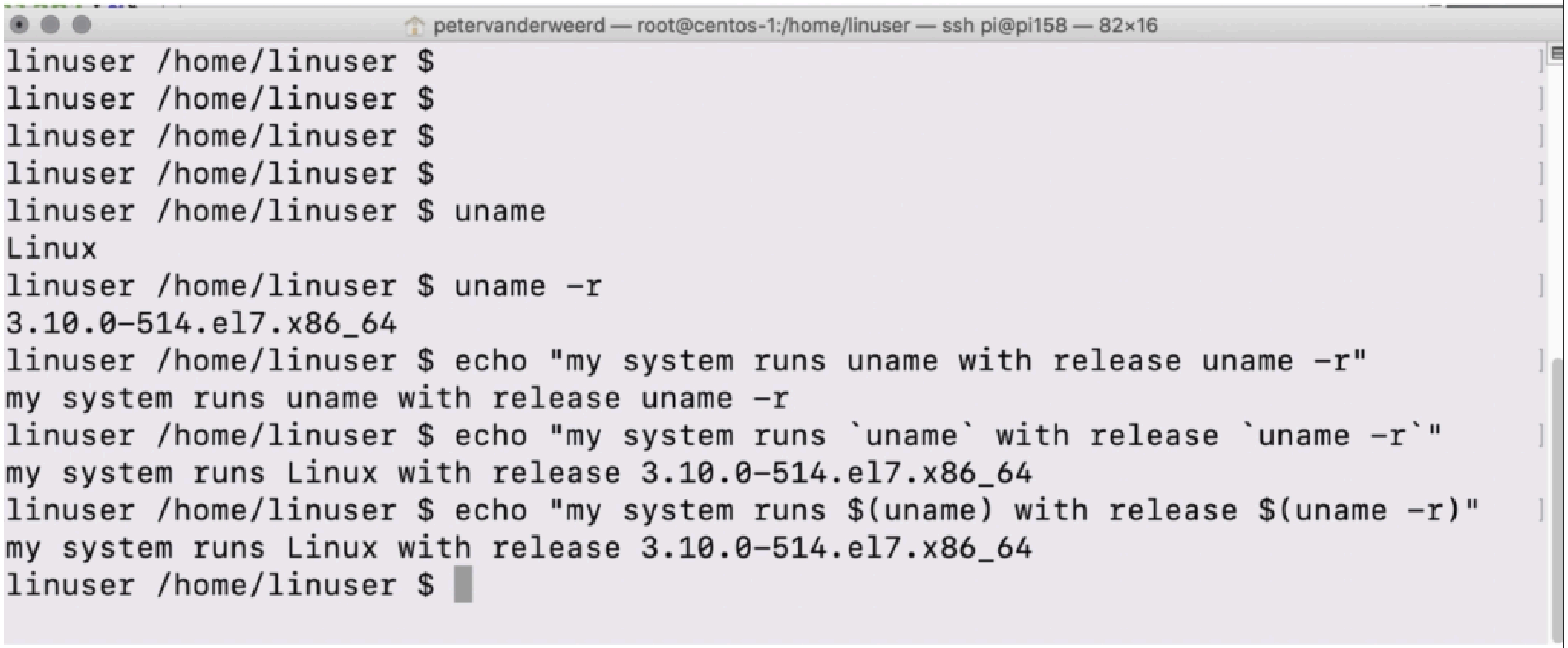
The back tick `

execute a **command** and have the output of
that **command** replace (**substitute**) the text of the **command**.

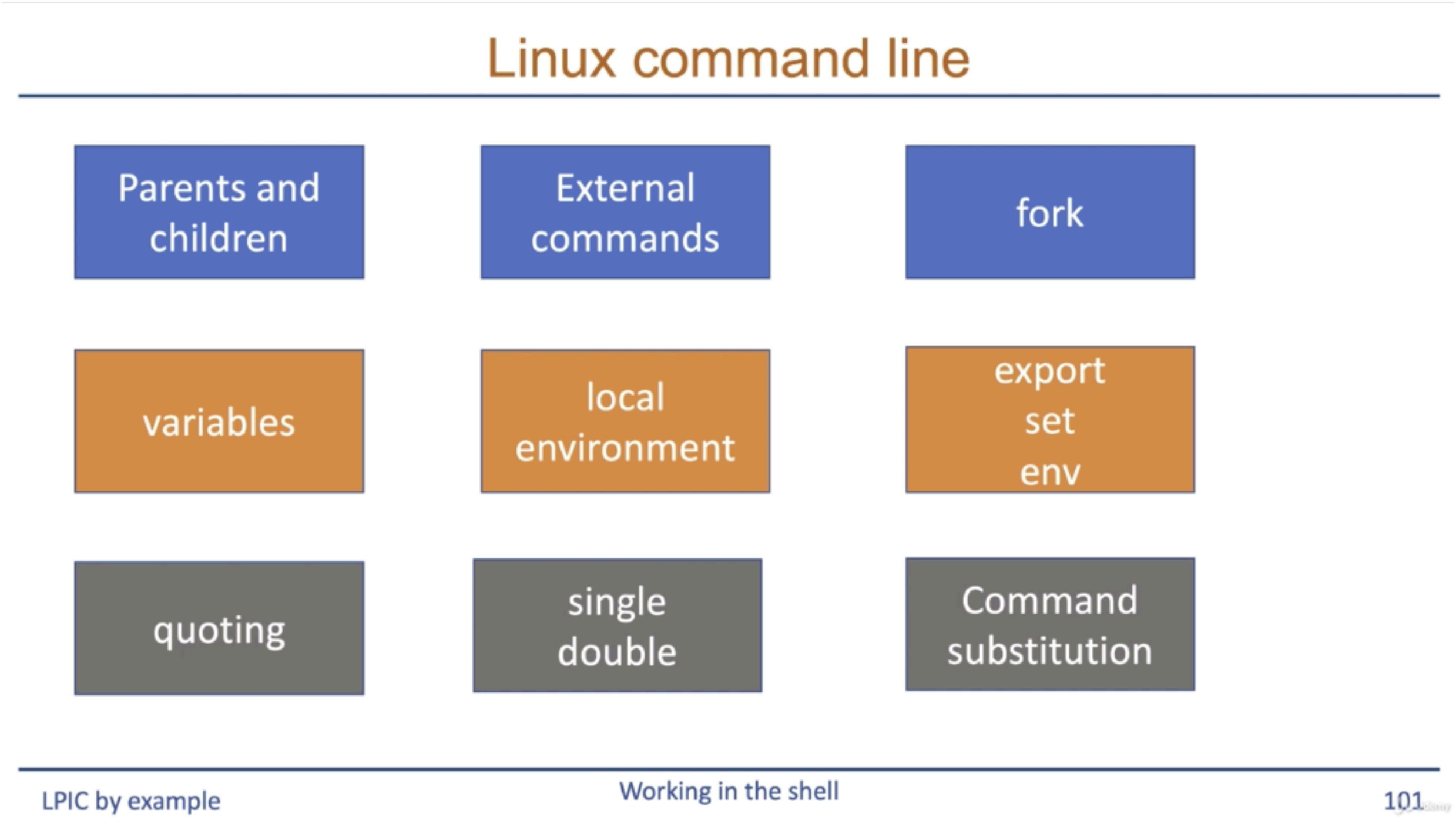
command

command

Linux command line


petervanderweerd — root@centos-1:/home/linuser — ssh pi@pi158 — 82×16

```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ uname
Linux
linuser /home/linuser $ uname -r
3.10.0-514.el7.x86_64
linuser /home/linuser $ echo "my system runs uname with release uname -r"
my system runs uname with release uname -r
linuser /home/linuser $ echo "my system runs `uname` with release `uname -r`"
my system runs Linux with release 3.10.0-514.el7.x86_64
linuser /home/linuser $ echo "my system runs $(uname) with release $(uname -r)"
my system runs Linux with release 3.10.0-514.el7.x86_64
linuser /home/linuser $
```



Linux command line

Parents and children

External commands

fork

variables

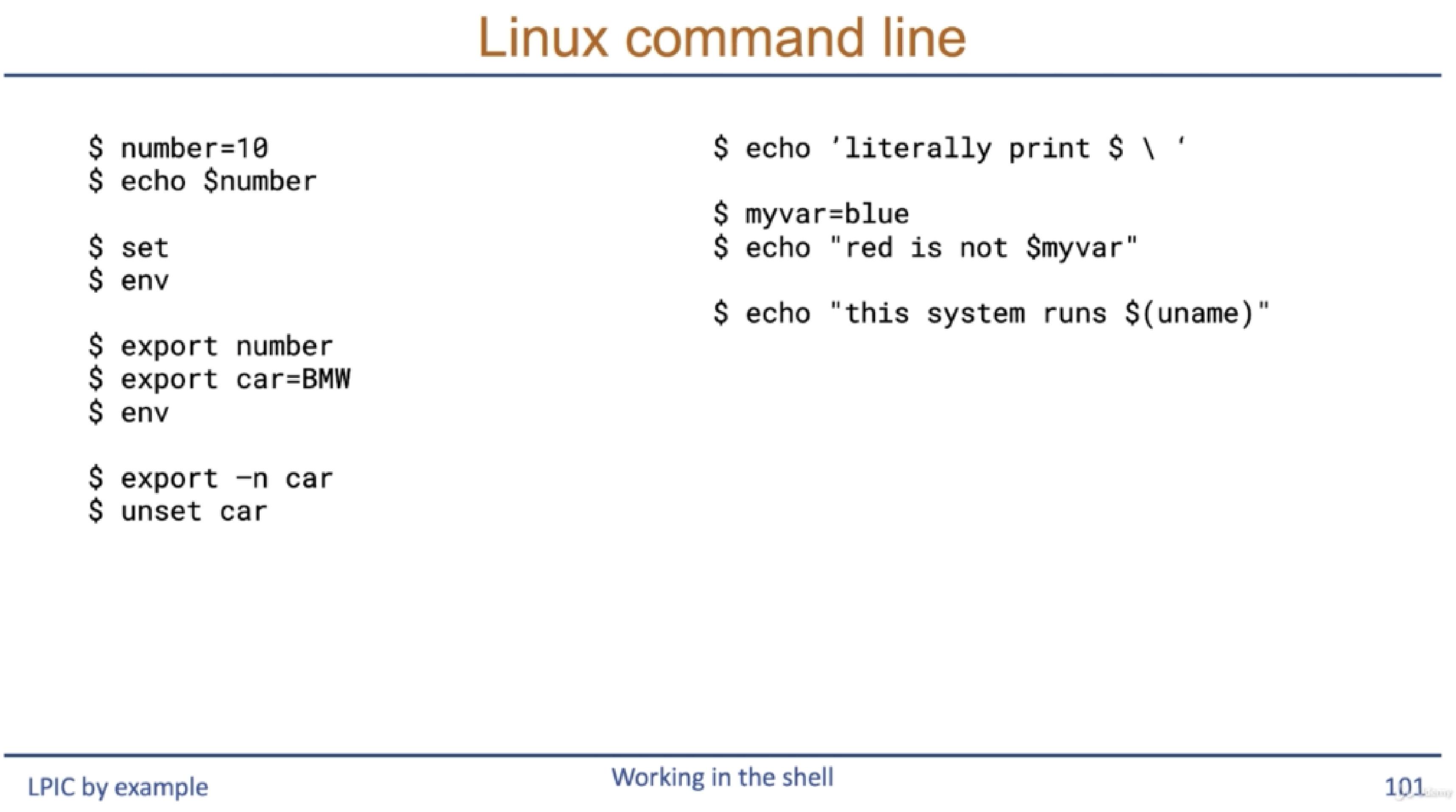
local environment

export
set
env

quoting

single
double

Command substitution



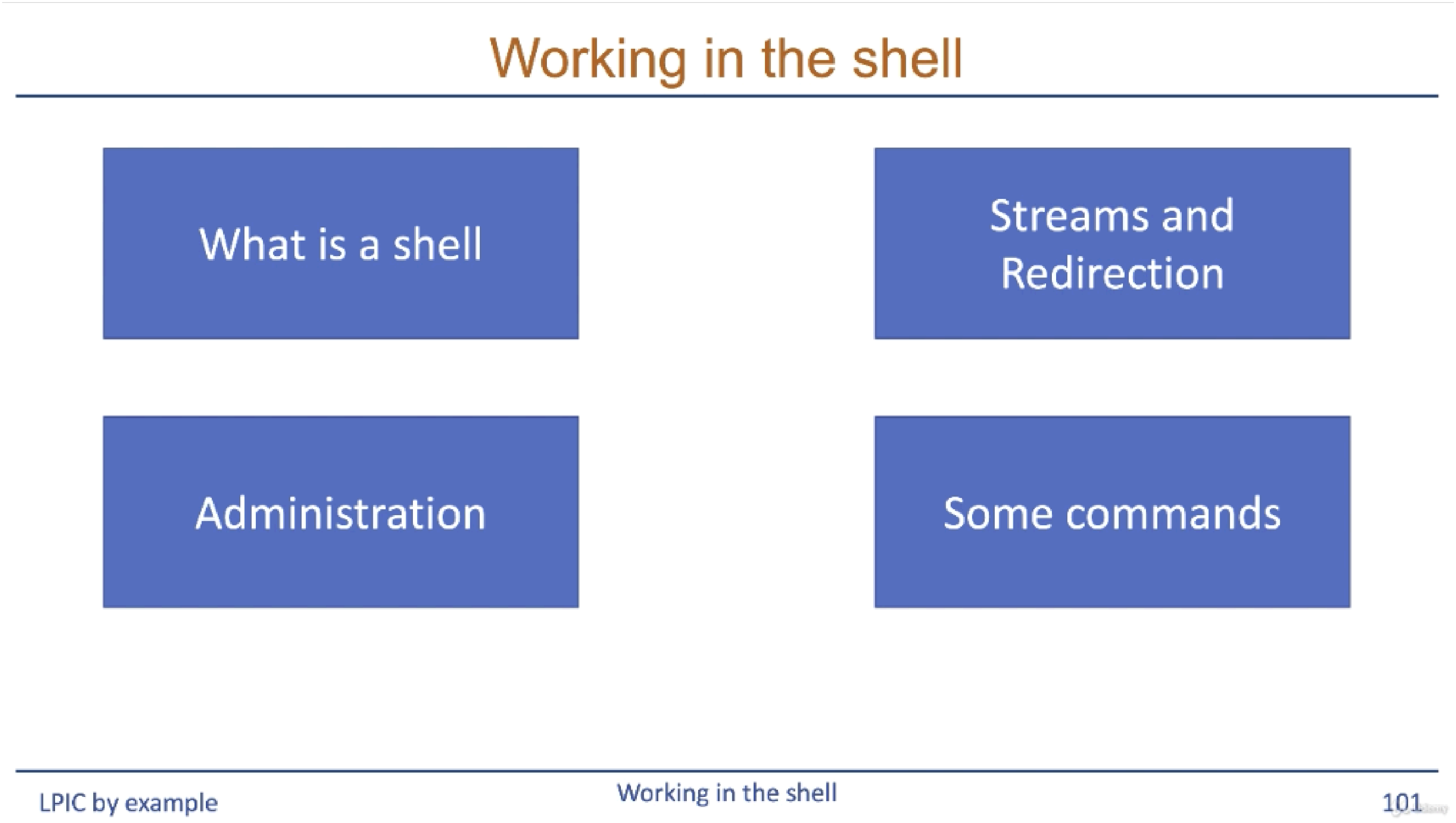
Linux command line

```
$ number=10
$ echo $number

$ set
$ env

$ export number
$ export car=BMW
$ env

$ export -n car
$ unset car
```

```
$ echo ’literally print $ \ ‘

$ myvar=blue
$ echo "red is not $myvar"

$ echo "this system runs $(uname)"
```

Working in the shell (part 1)

LPIC by example



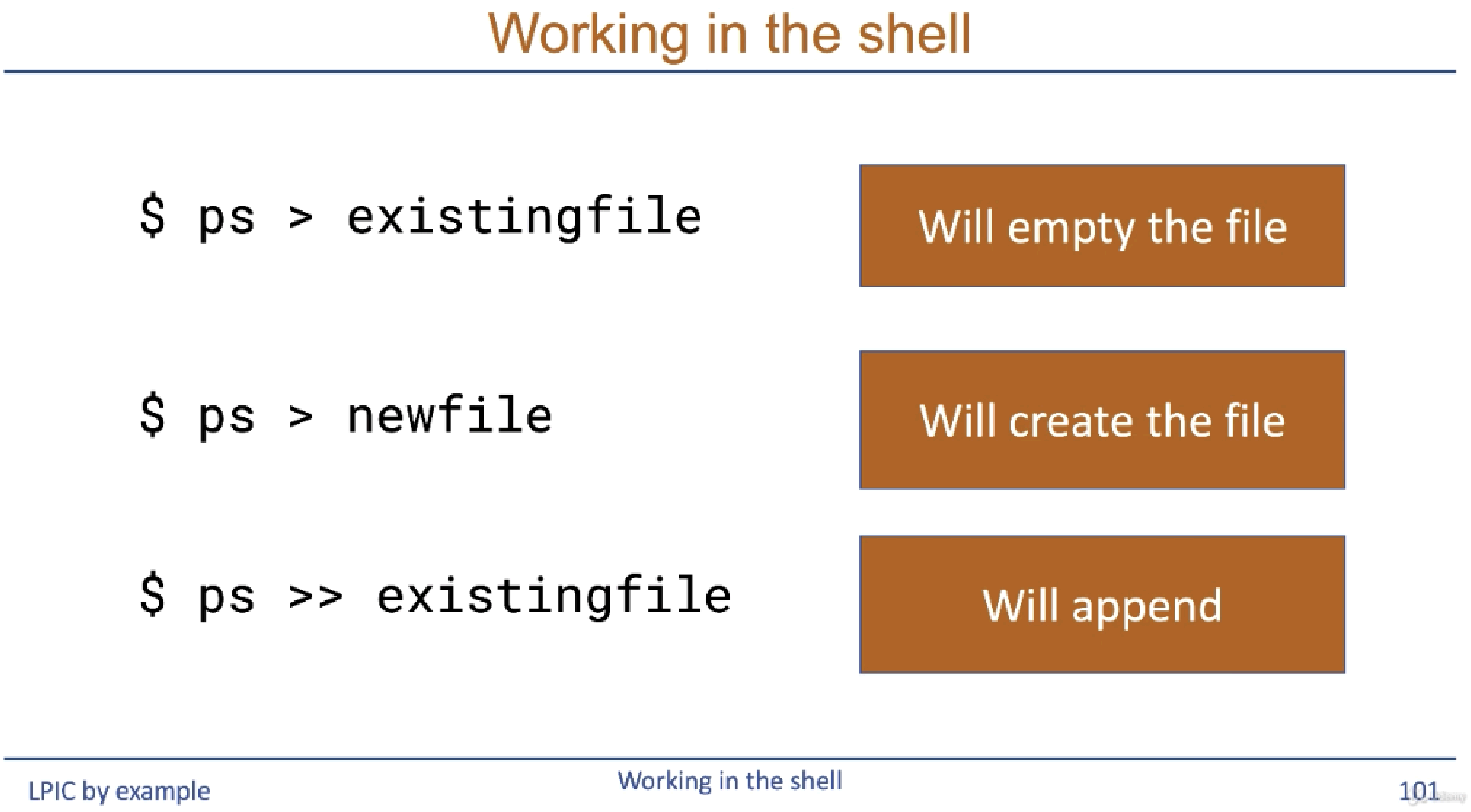
Working in the shell

What is a shell

Streams and Redirection

Administration

Some commands



Working in the shell

```
$ ps > existingfile
```

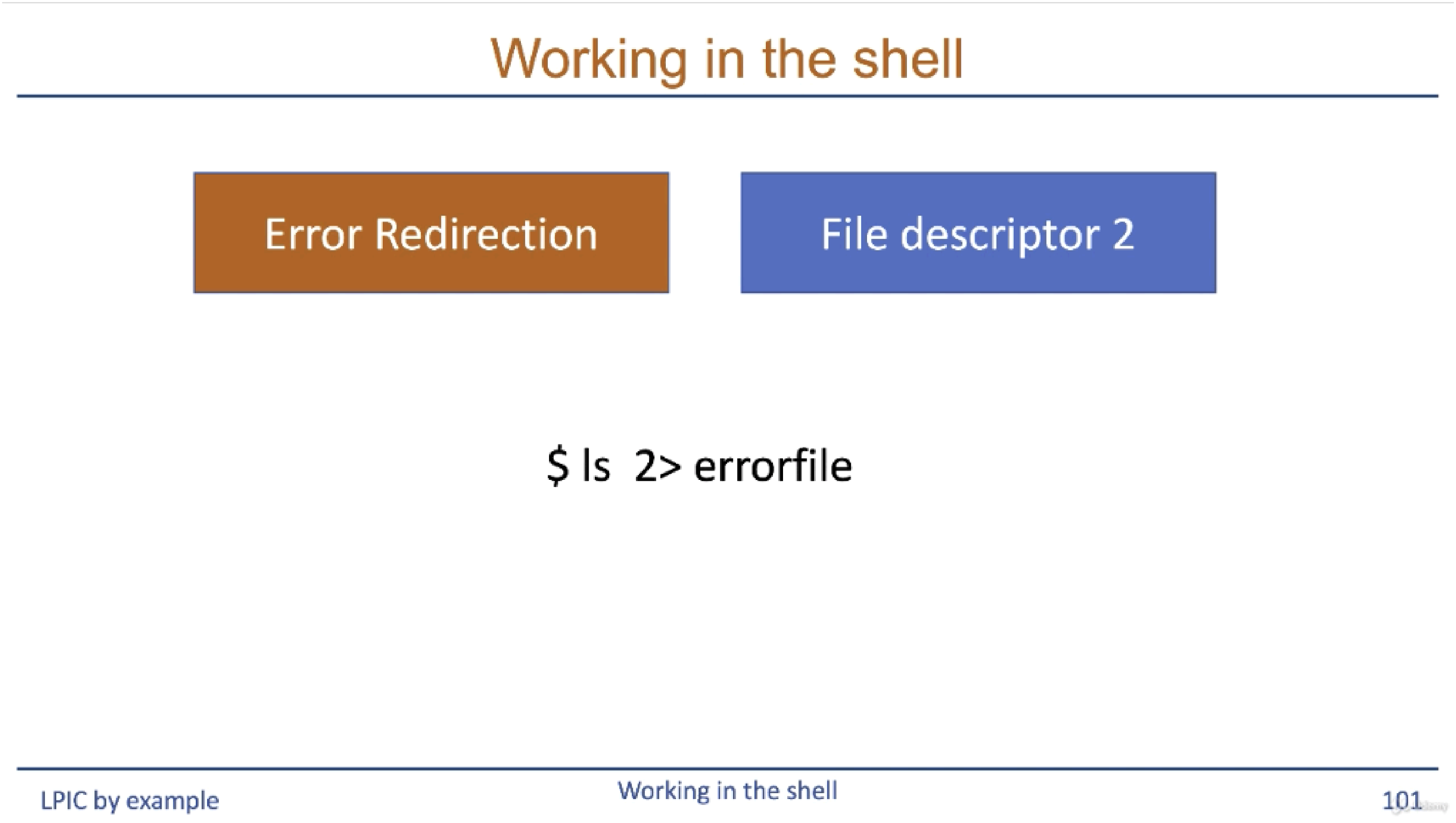
Will empty the file

```
$ ps > newfile
```

Will create the file

```
$ ps >> existingfile
```

Will append

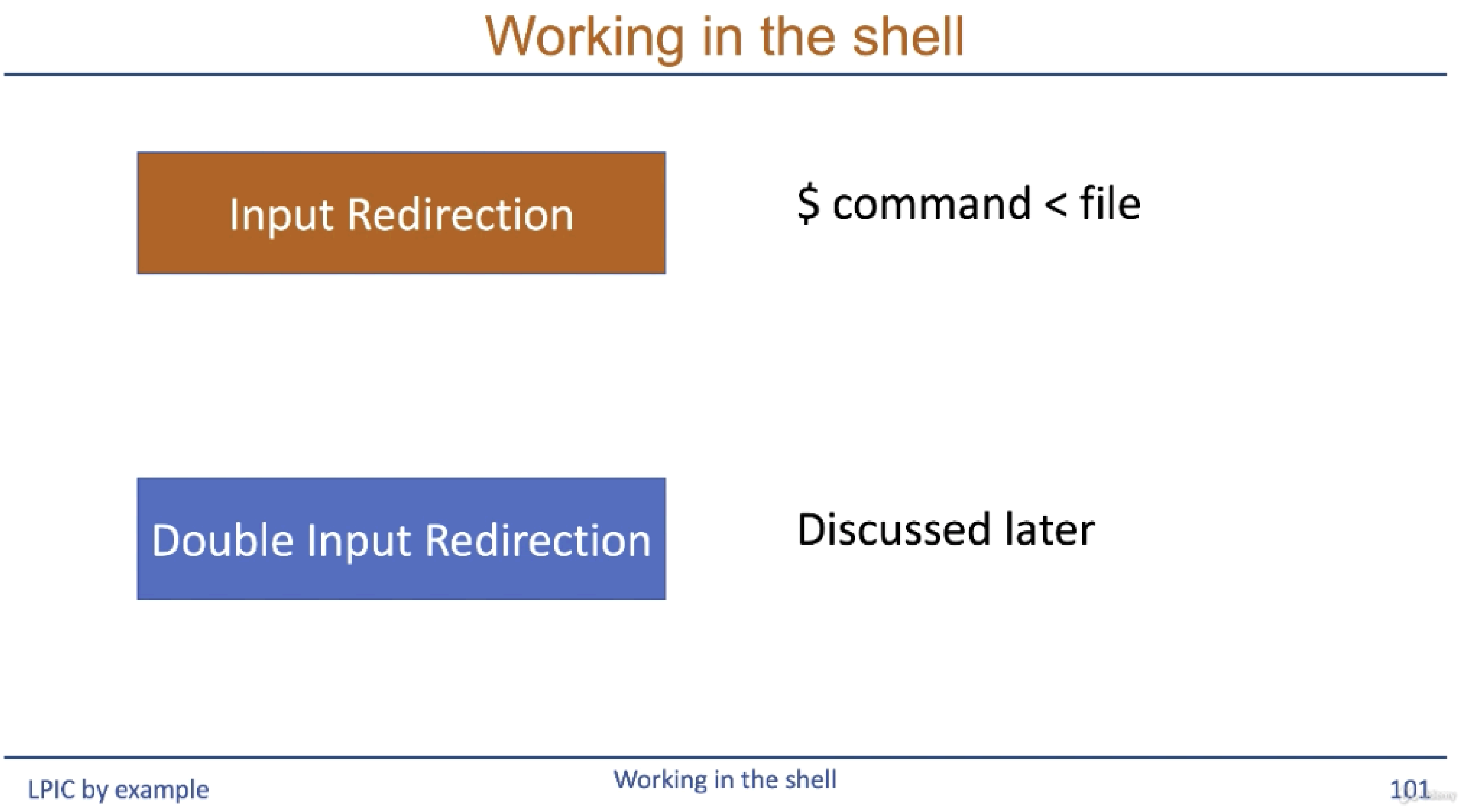


Working in the shell

Error Redirection

File descriptor 2

```
$ ls  2> errorfile
```



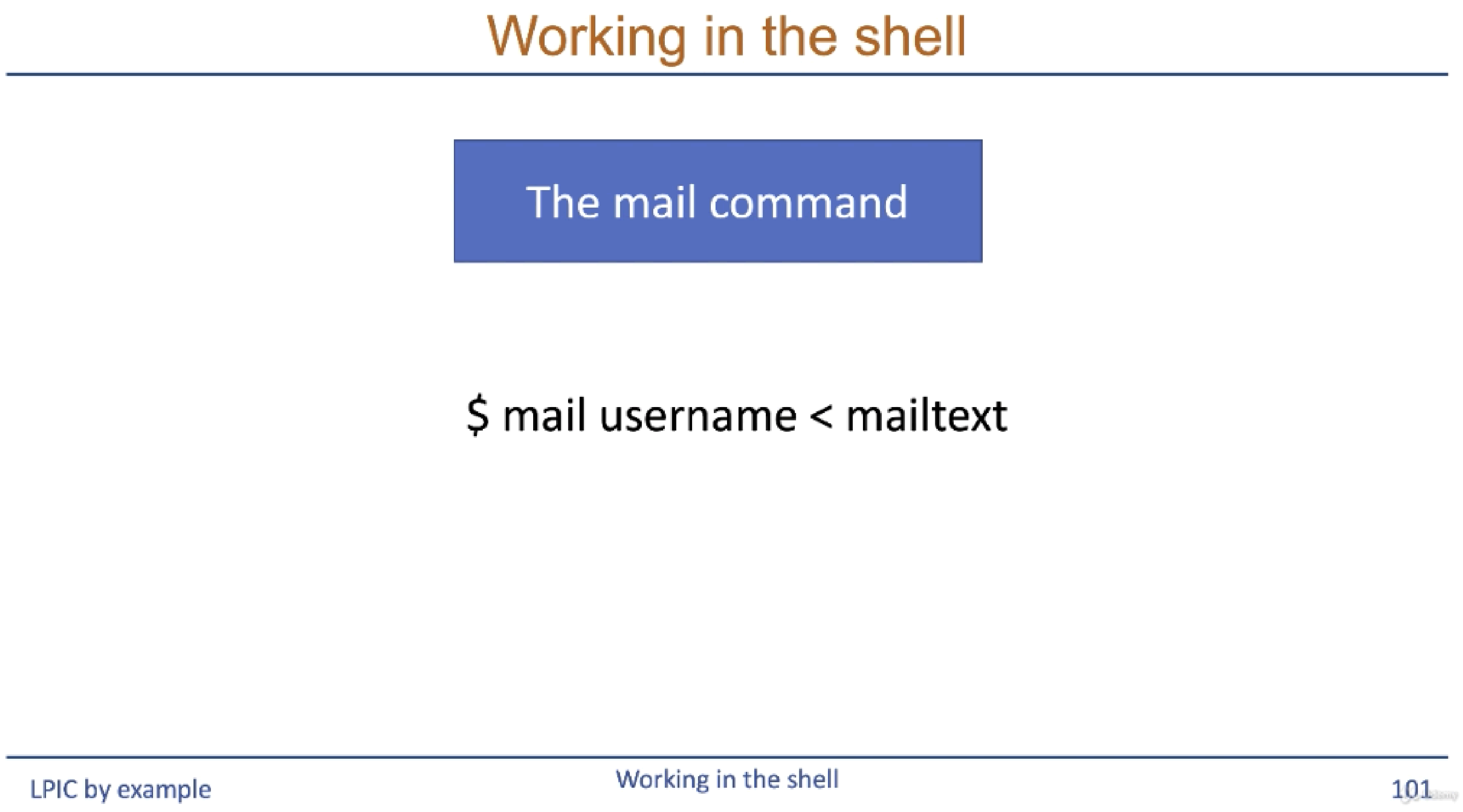
Working in the shell

Input Redirection

```
$ command < file
```

Double Input Redirection

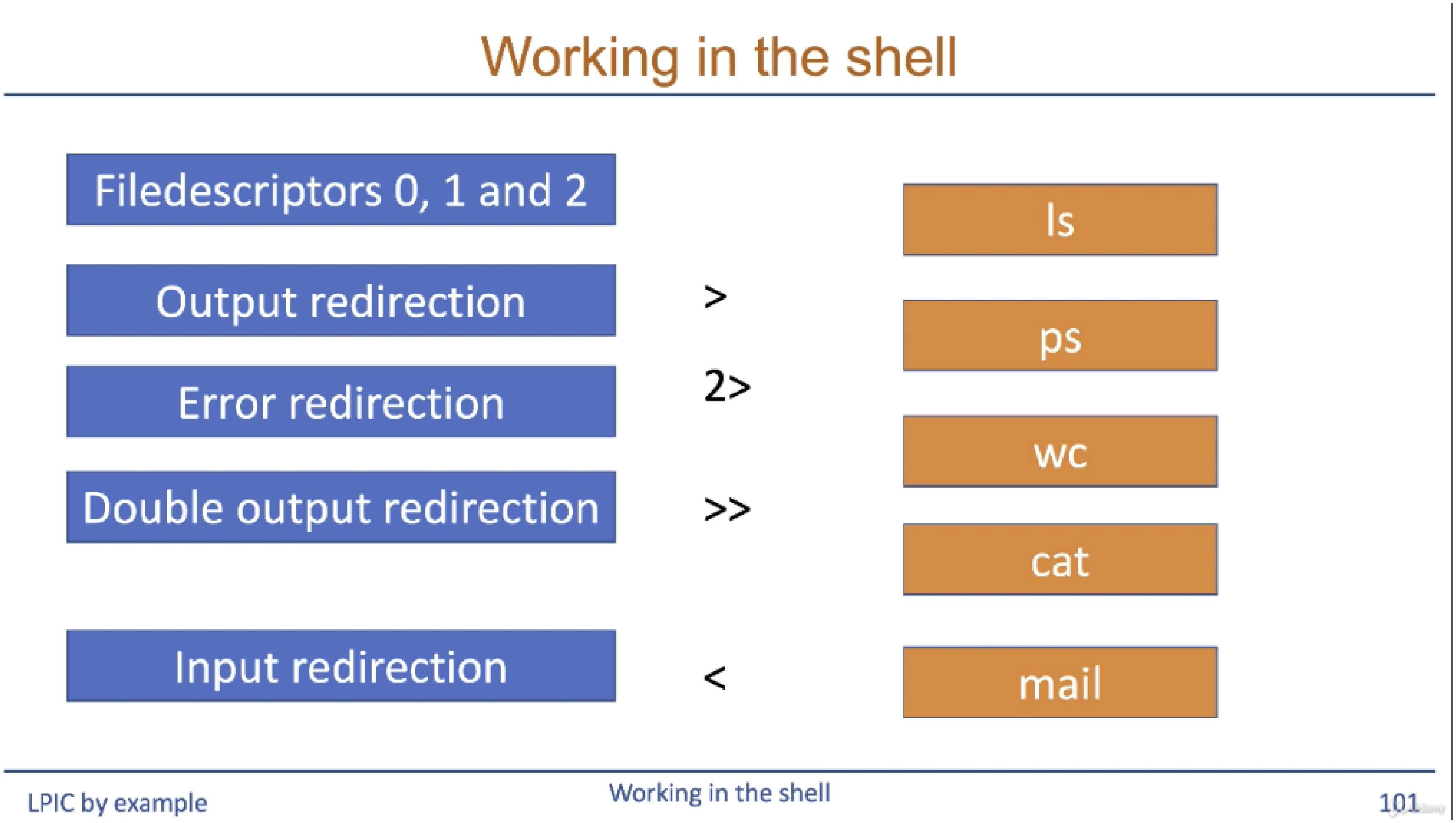
Discussed later



Working in the shell

The mail command

```
$ mail username < mailtext
```



Working in the shell

Filedescriptors 0, 1 and 2

Output redirection >

Error redirection 2>

Double output redirection >>

Input redirection <

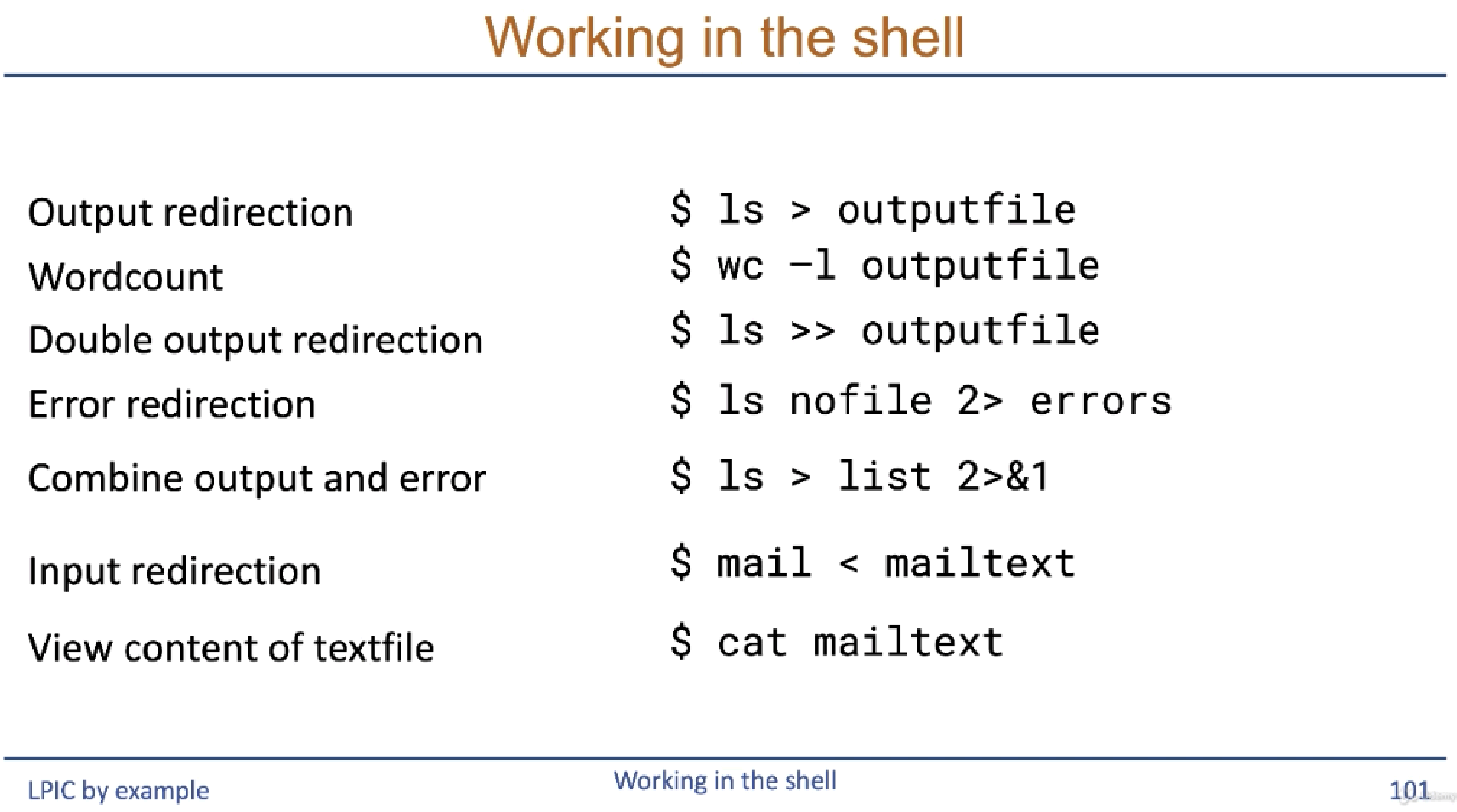
ls

ps

wc

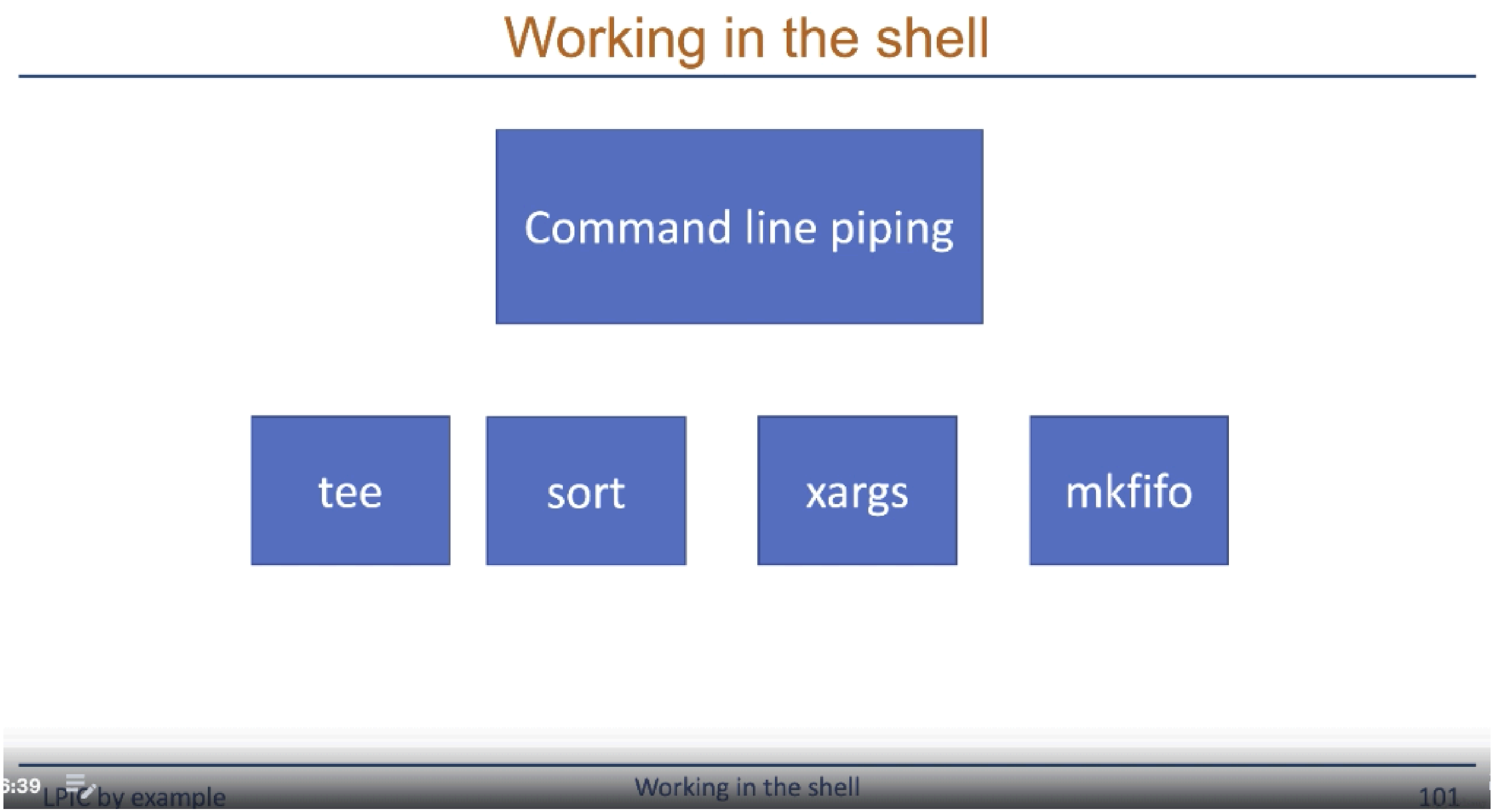
cat

mail



Working in the shell

Output redirection	`$ ls > outputfile`
Wordcount	`$ wc –l outputfile`
Double output redirection	`$ ls >> outputfile`
Error redirection	`$ ls nofile 2> errors`
Combine output and error	`$ ls > list 2>&1`
Input redirection	`$ mail < mailtext`
View content of textfile	`$ cat mailtext`



Working in the shell

Command line piping

tee

sort

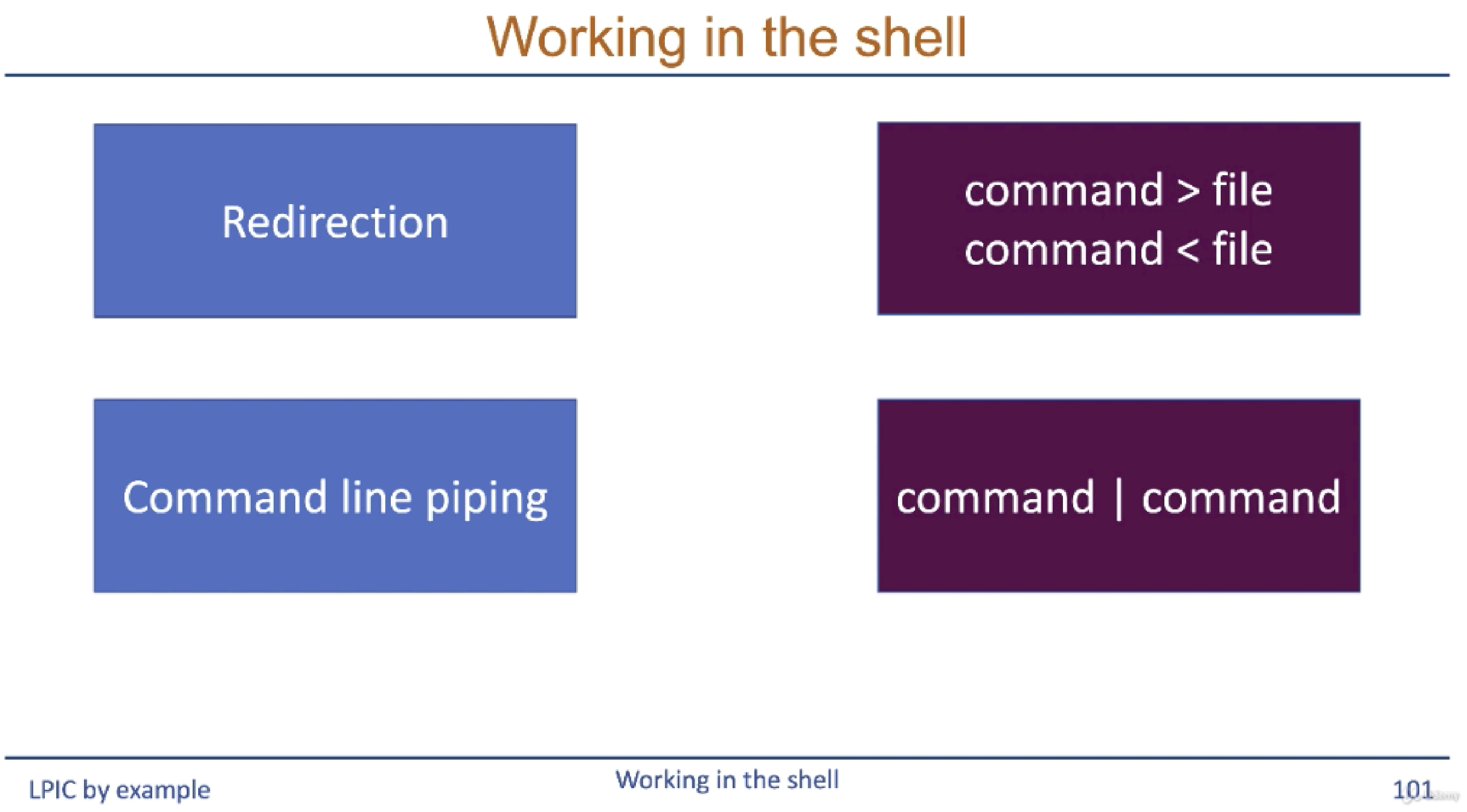
xargs

mkfifo

LPIC by example

Working in the shell

101



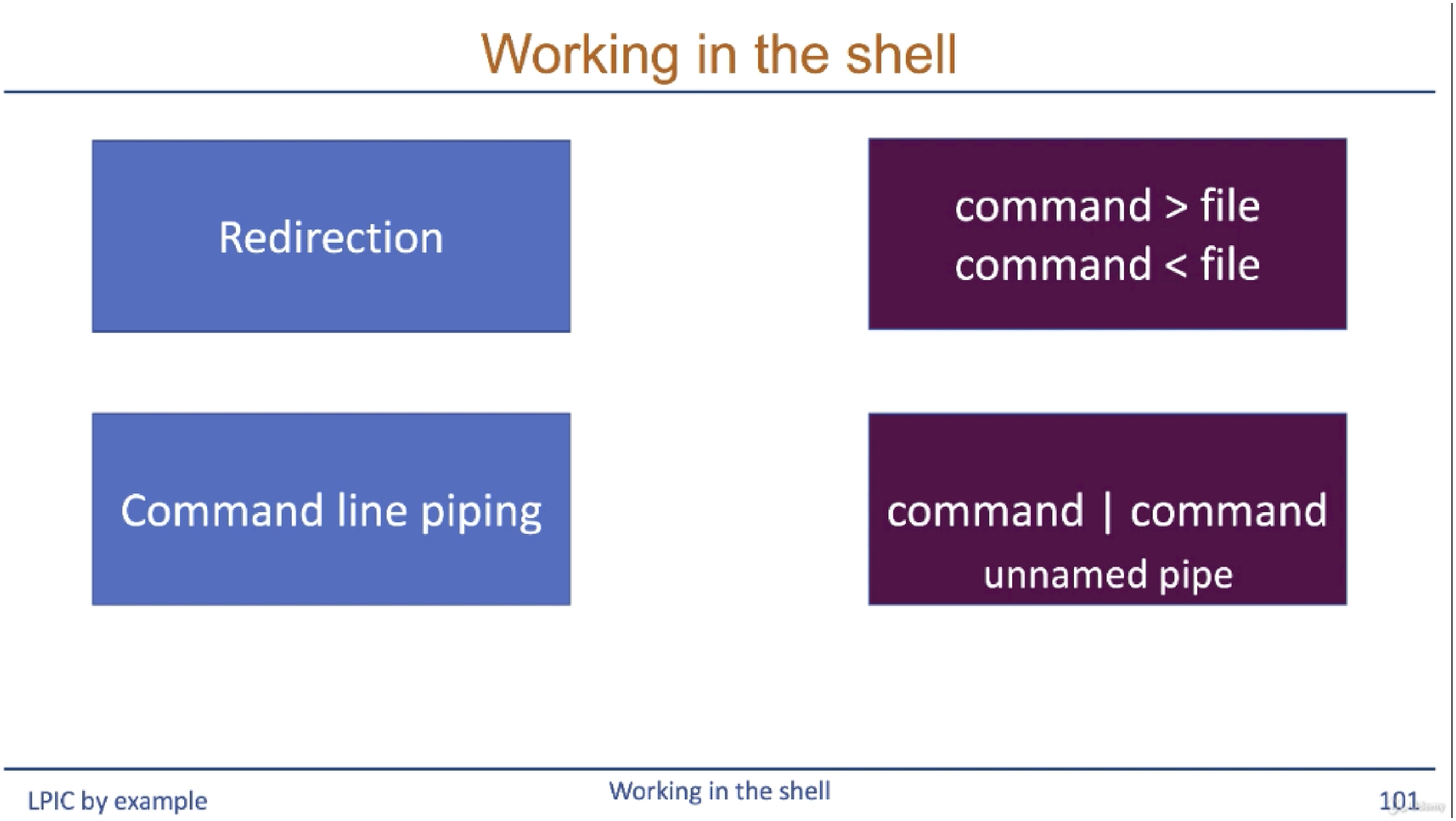
Working in the shell

Redirection

```
command > file
command < file
```

Command line piping

```
command | command
```



Working in the shell

Redirection

command > file
command < file

Command line piping

command | command
unnamed pipe

Working in the shell

Command line piping

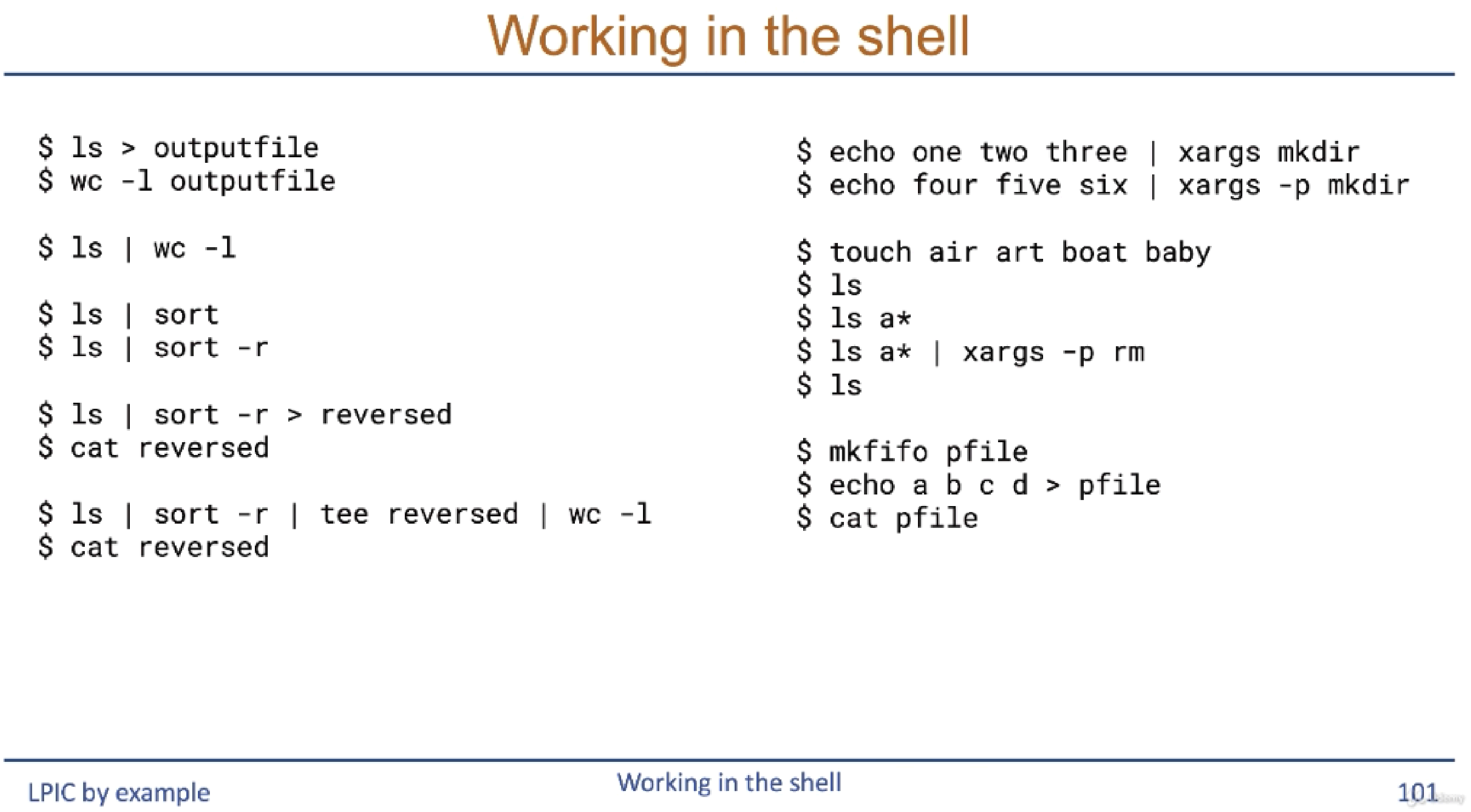
Combine with redirection

tee

xargs

sort

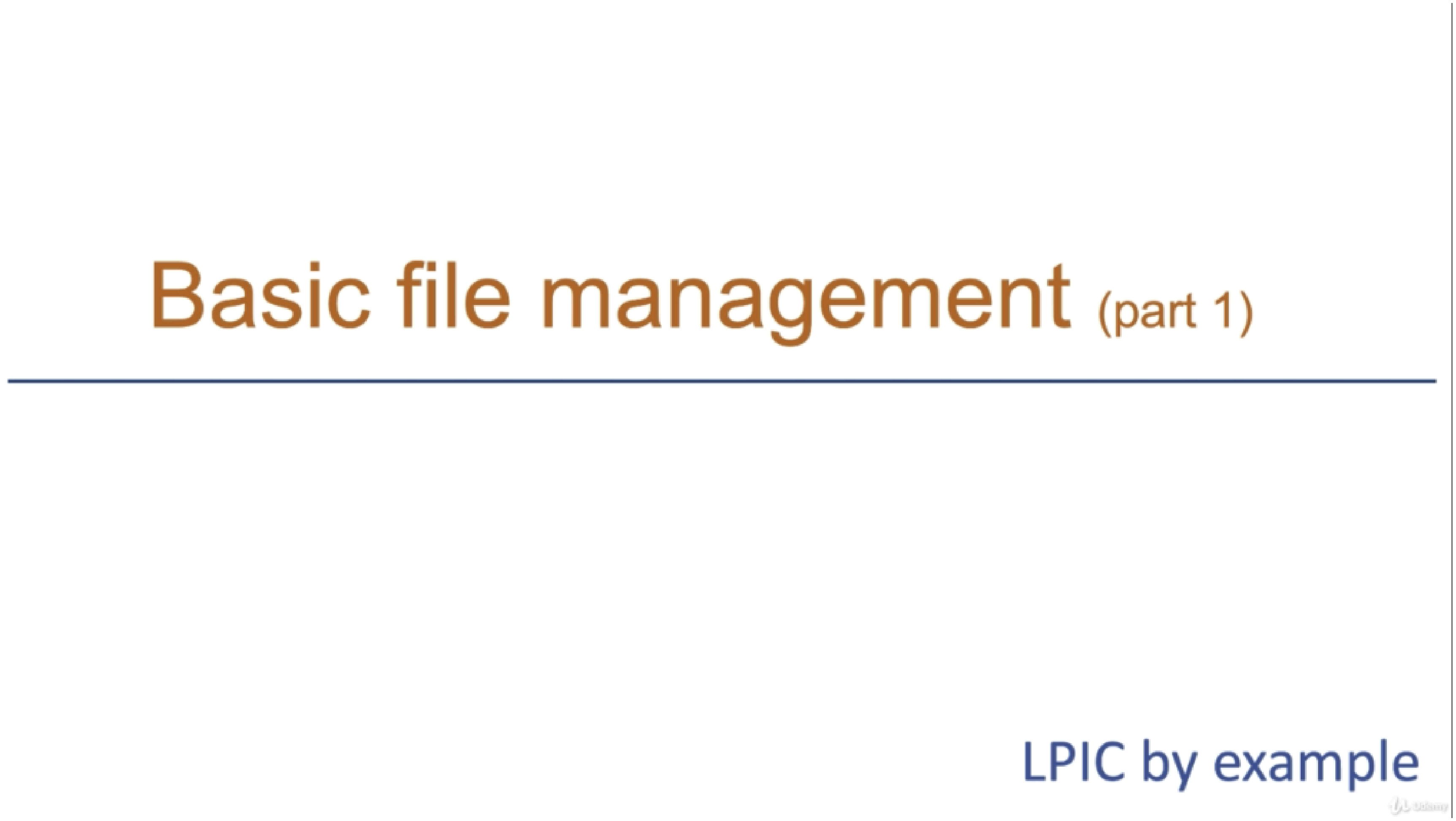
mkfifo



Working in the shell

```
$ ls > outputfile
$ wc -l outputfile

$ ls | wc -l

$ ls | sort
$ ls | sort -r

$ ls | sort -r > reversed
$ cat reversed

$ ls | sort -r | tee reversed | wc -l
$ cat reversed
```

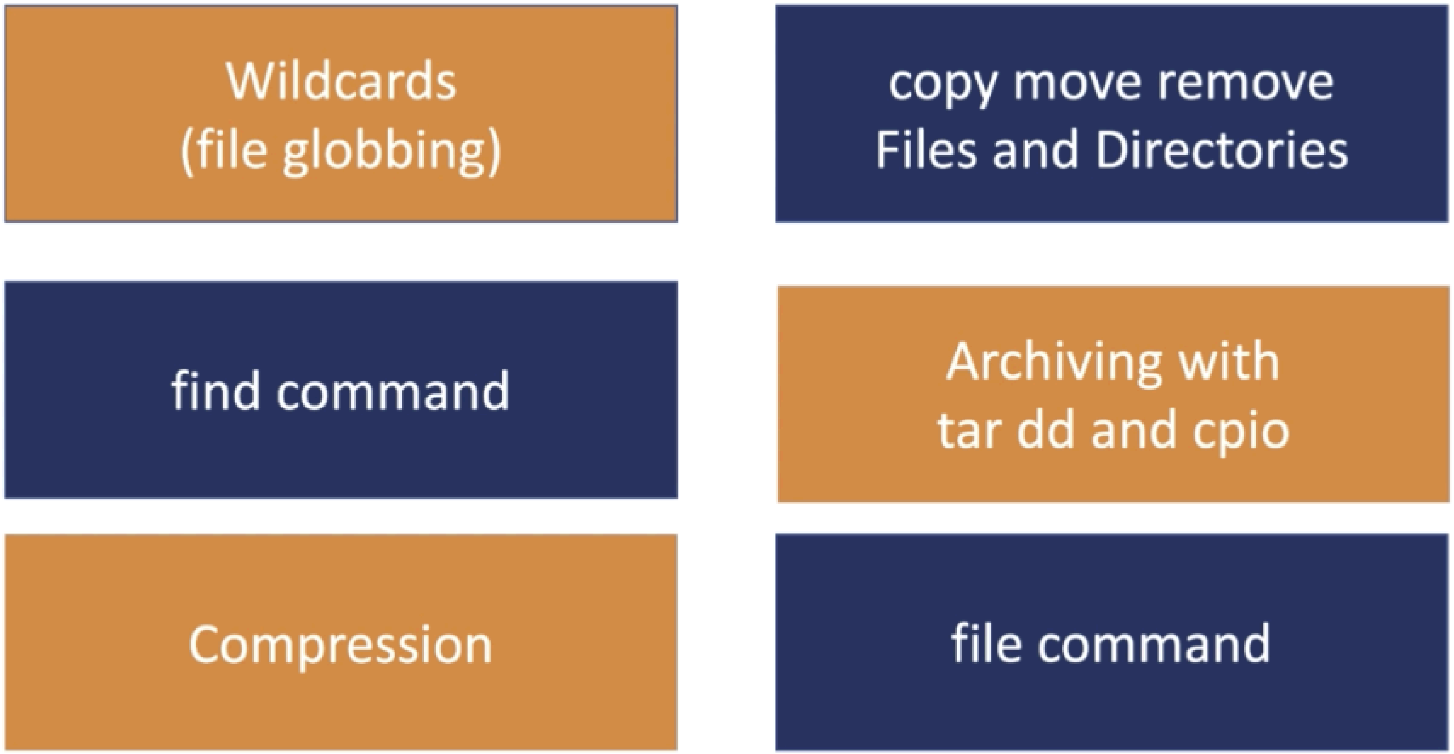
```
$ echo one two three | xargs mkdir
$ echo four five six | xargs -p mkdir

$ touch air art boat baby
$ ls
$ ls a*
$ ls a* | xargs -p rm
$ ls

$ mkfifo pfile
$ echo a b c d > pfile
$ cat pfile
```



Basic file management (part 1)

LPIC by example

Basic file management



Wildcards
(file globbing)

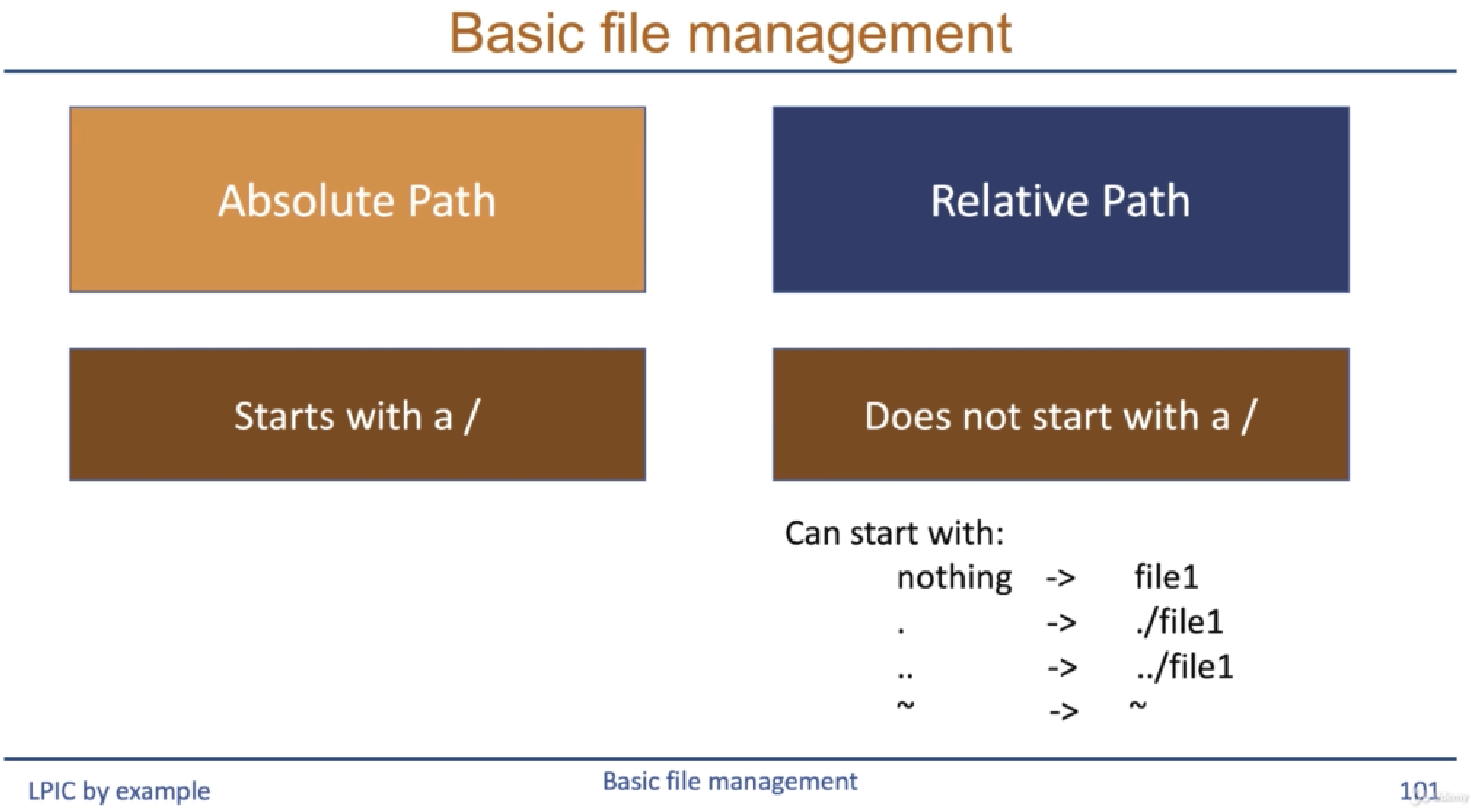
copy move remove
Files and Directories

find command

Archiving with
tar dd and cpio

Compression

file command



Basic file management

Absolute Path

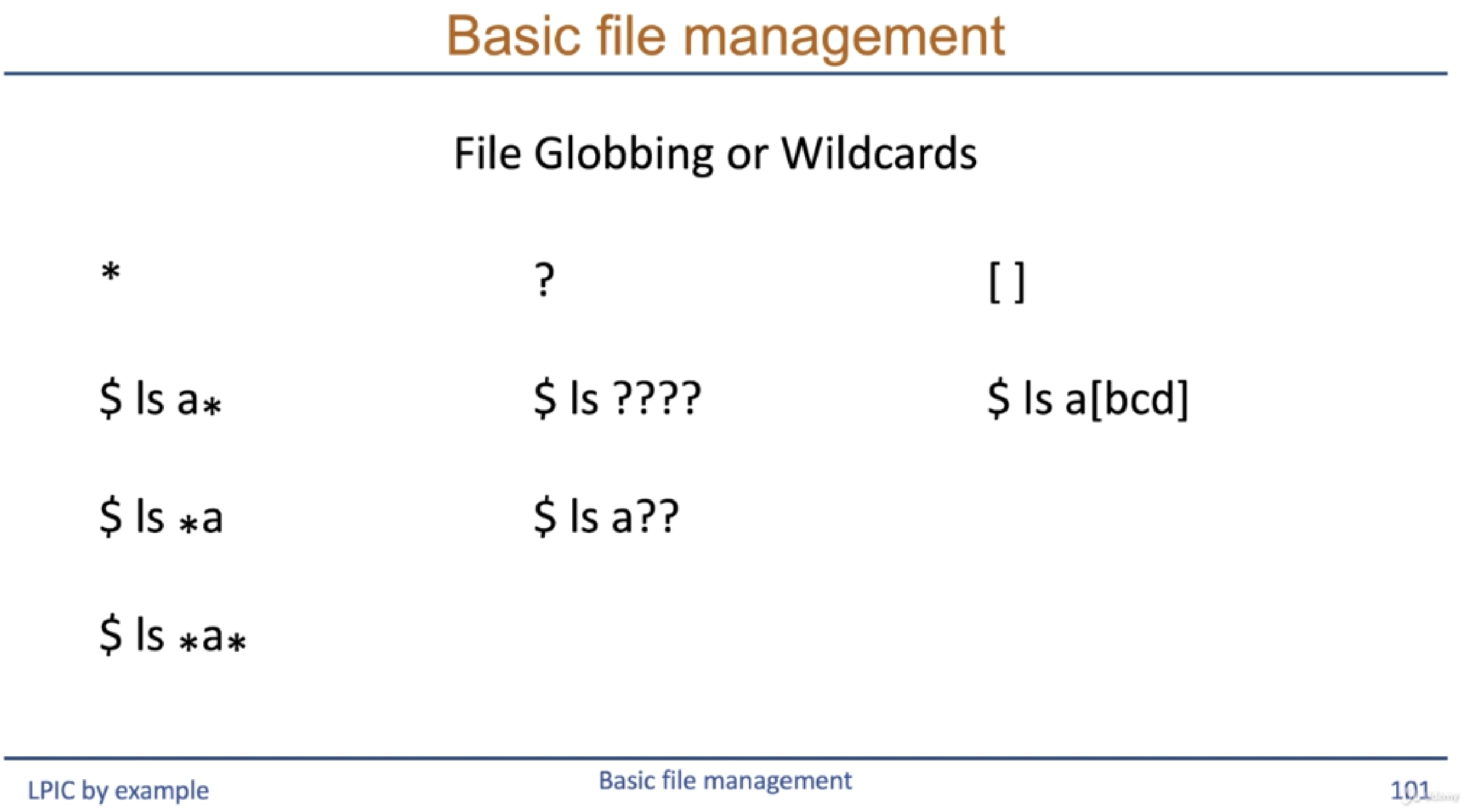
Starts with a /

Relative Path

Does not start with a /

Can start with:

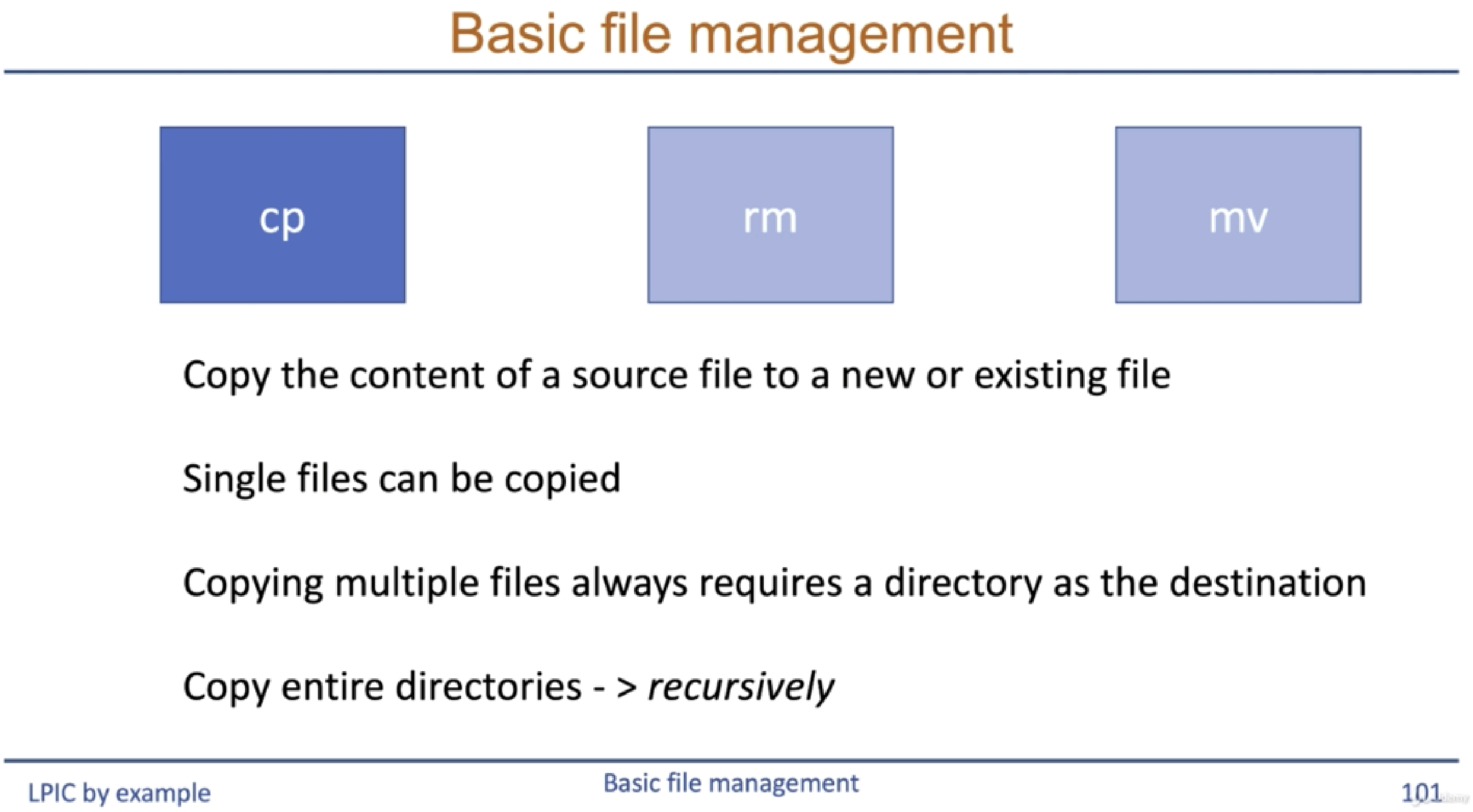
nothing	->	file1
.	->	./file1
..	->	../file1
~	->	~



Basic file management

File Globbing or Wildcards

*	?	[ ]
$ ls a*	$ ls ????	$ ls a[bcd]
$ ls *a	$ ls a??	
$ ls *a*		



Basic file management

cp

rm

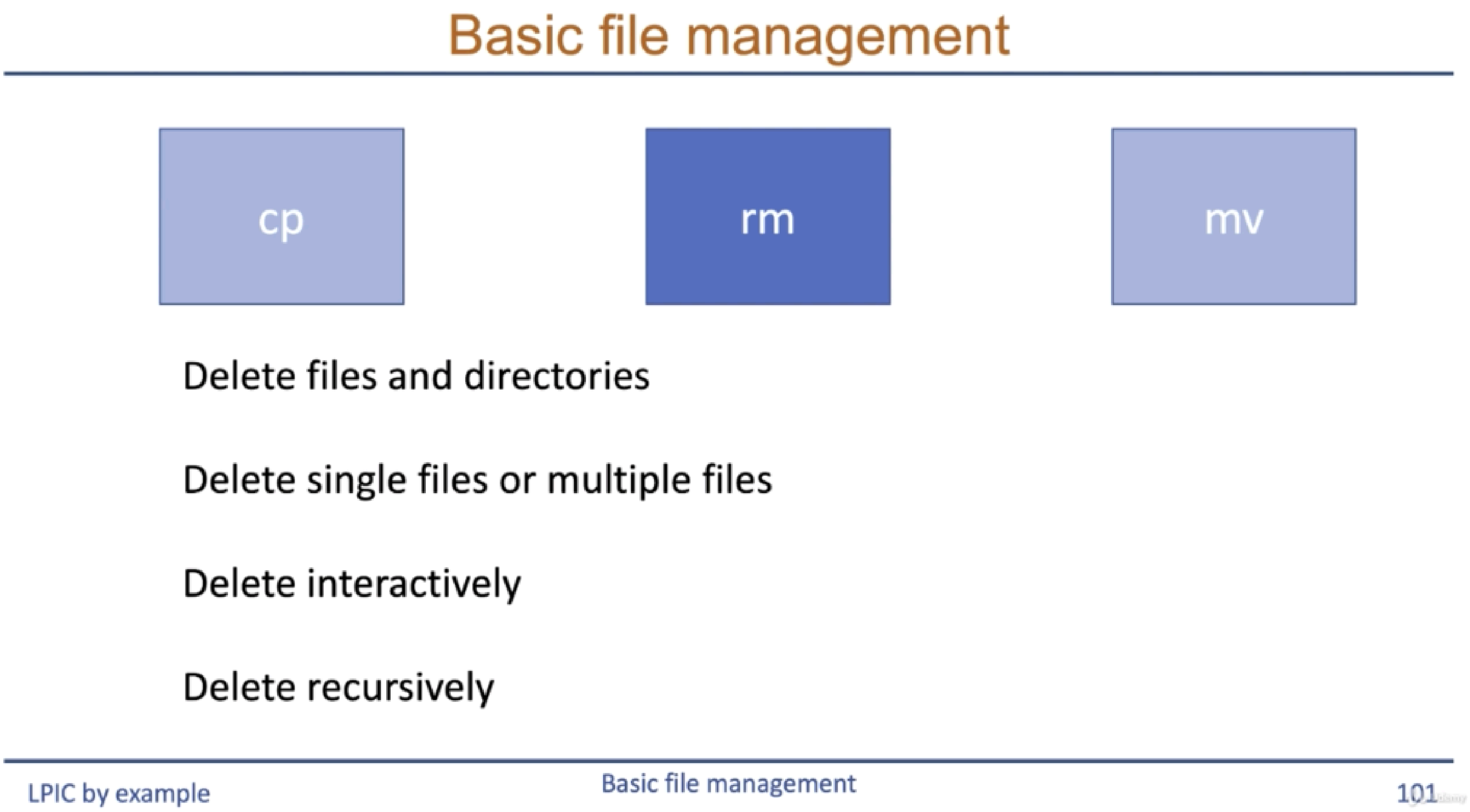
mv

Copy the content of a source file to a new or existing file

Single files can be copied

Copying multiple files always requires a directory as the destination

Copy entire directories - > *recursively*



Basic file management

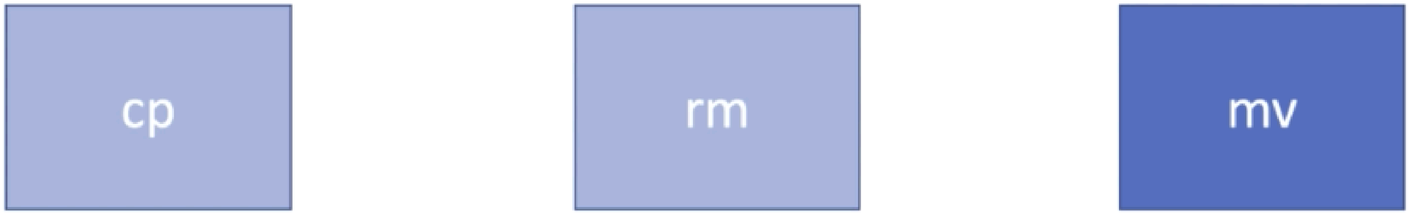
Delete files and directories

Delete single files or multiple files

Delete interactively

Delete recursively

Basic file management



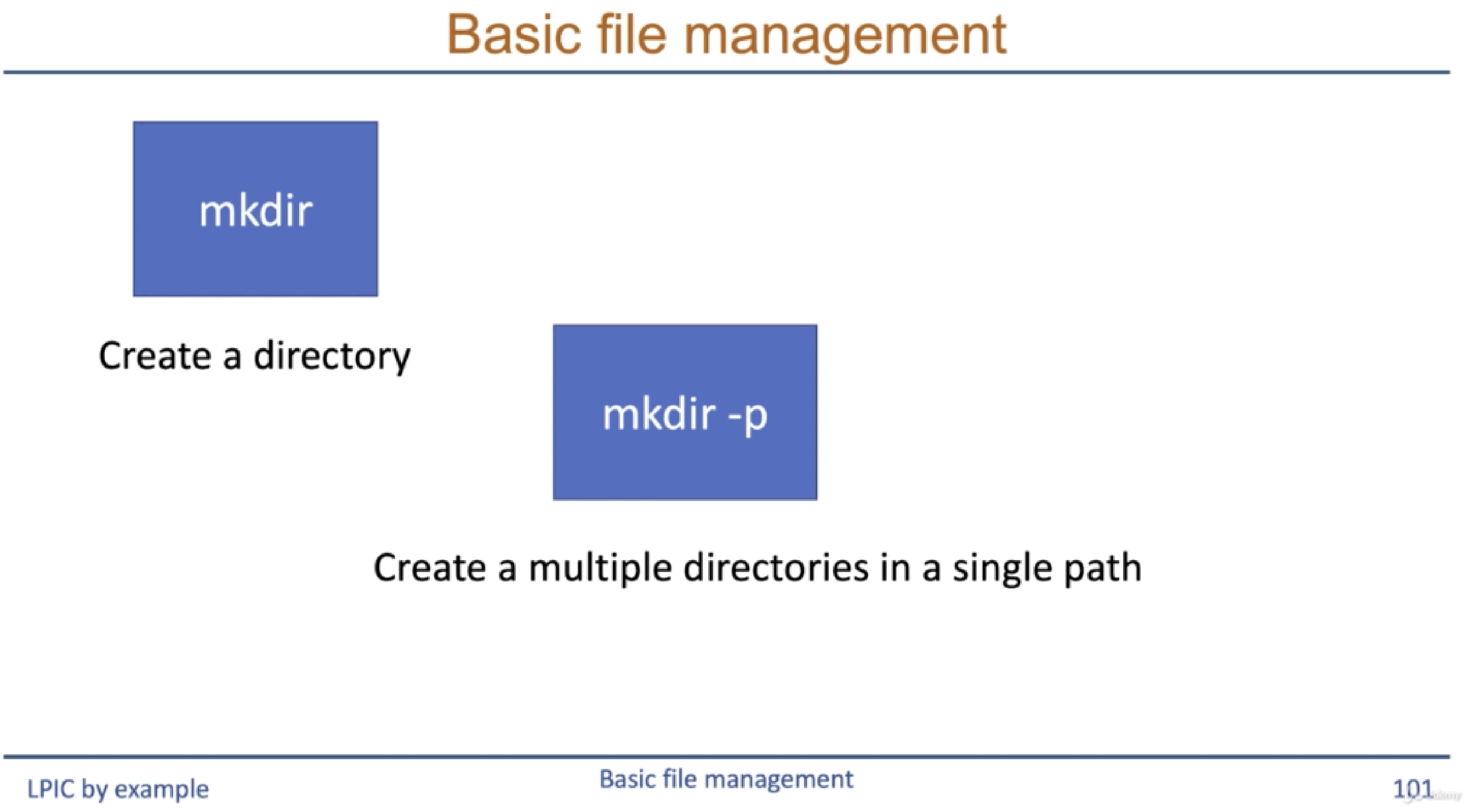
cp

rm

mv

Renames

Moves and renames



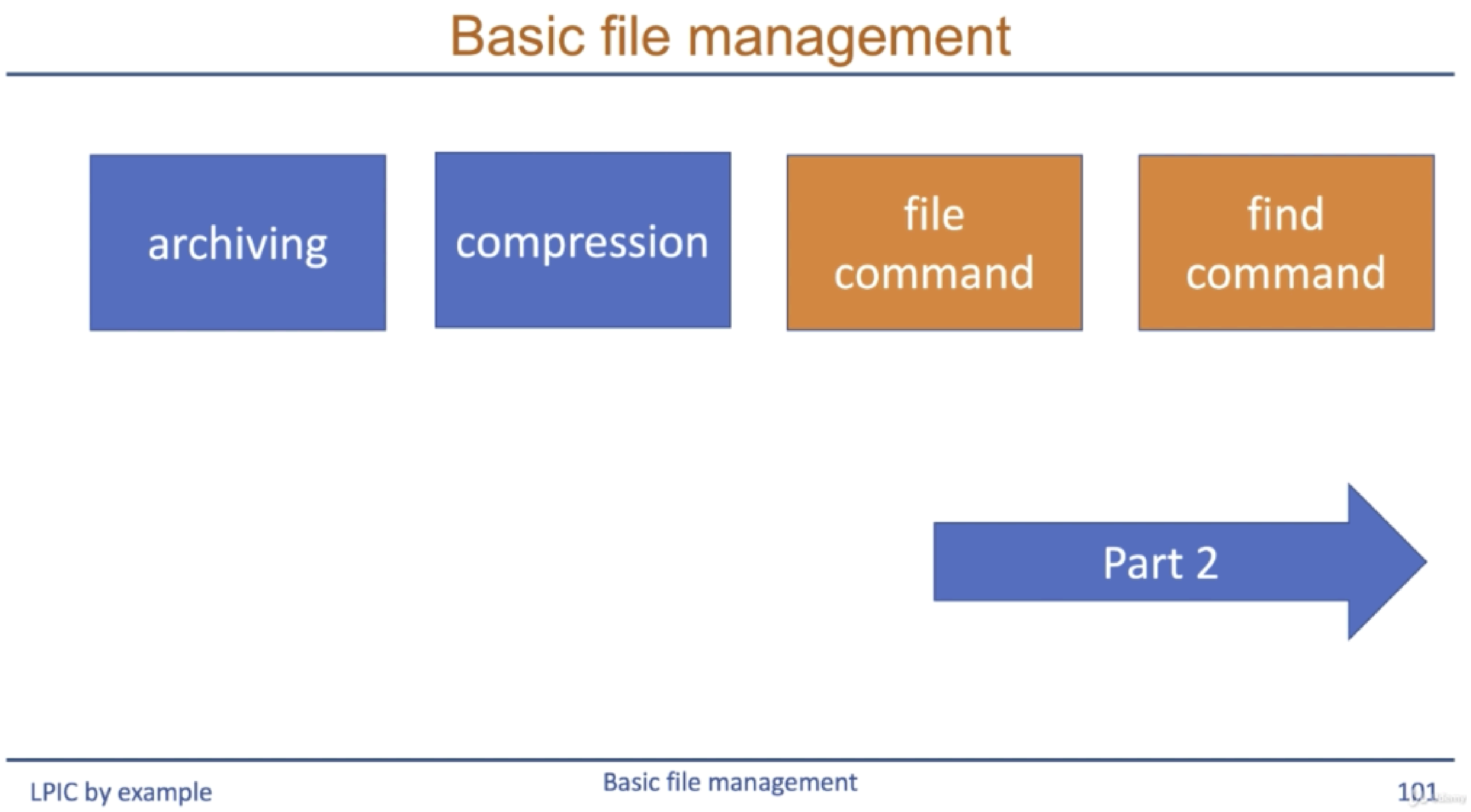
Basic file management

mkdir

Create a directory

mkdir -p

Create a multiple directories in a single path



Basic file management

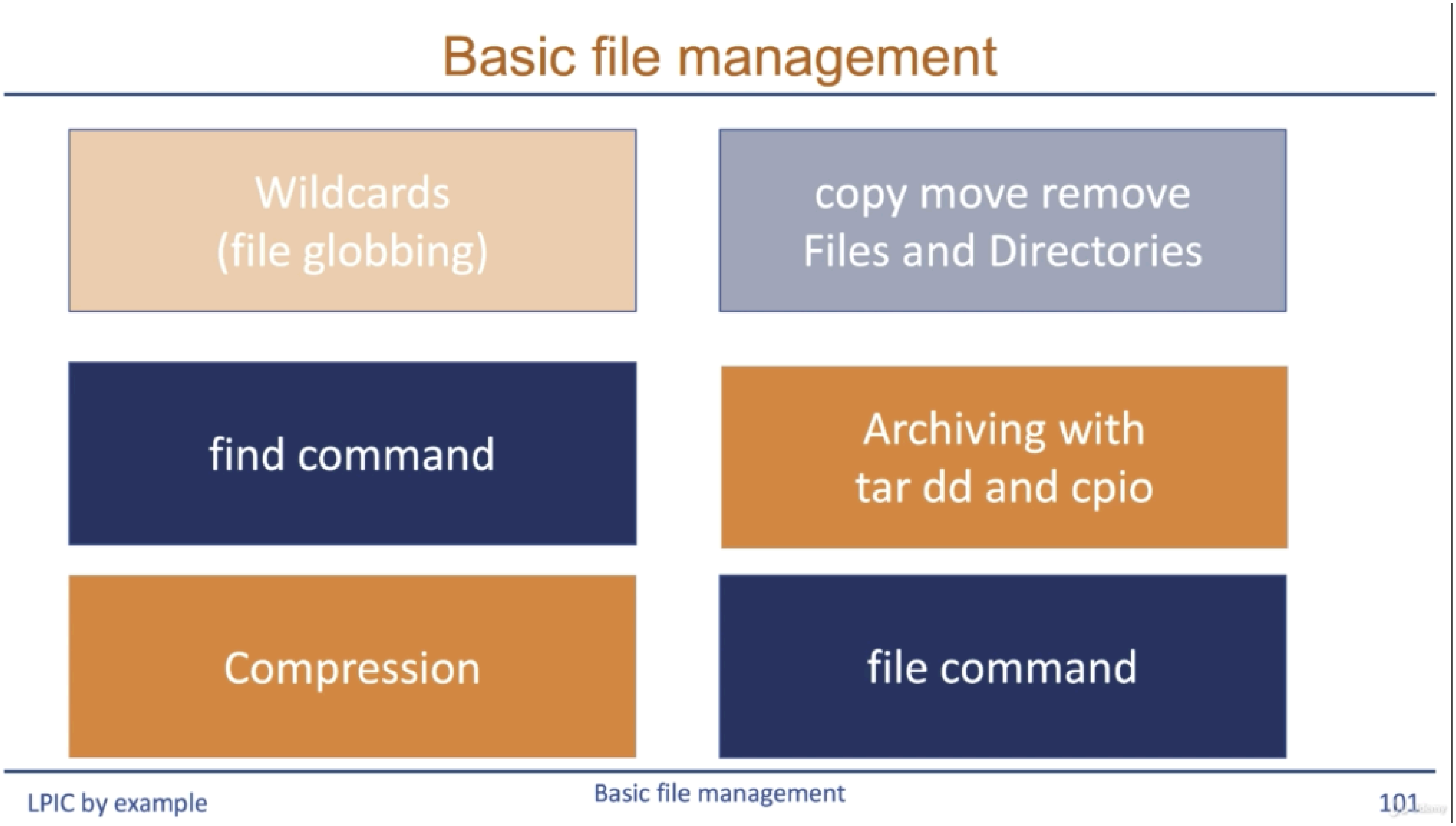
archiving

compression

file command

find command

Part 2



Basic file management

Wildcards
(file globbing)

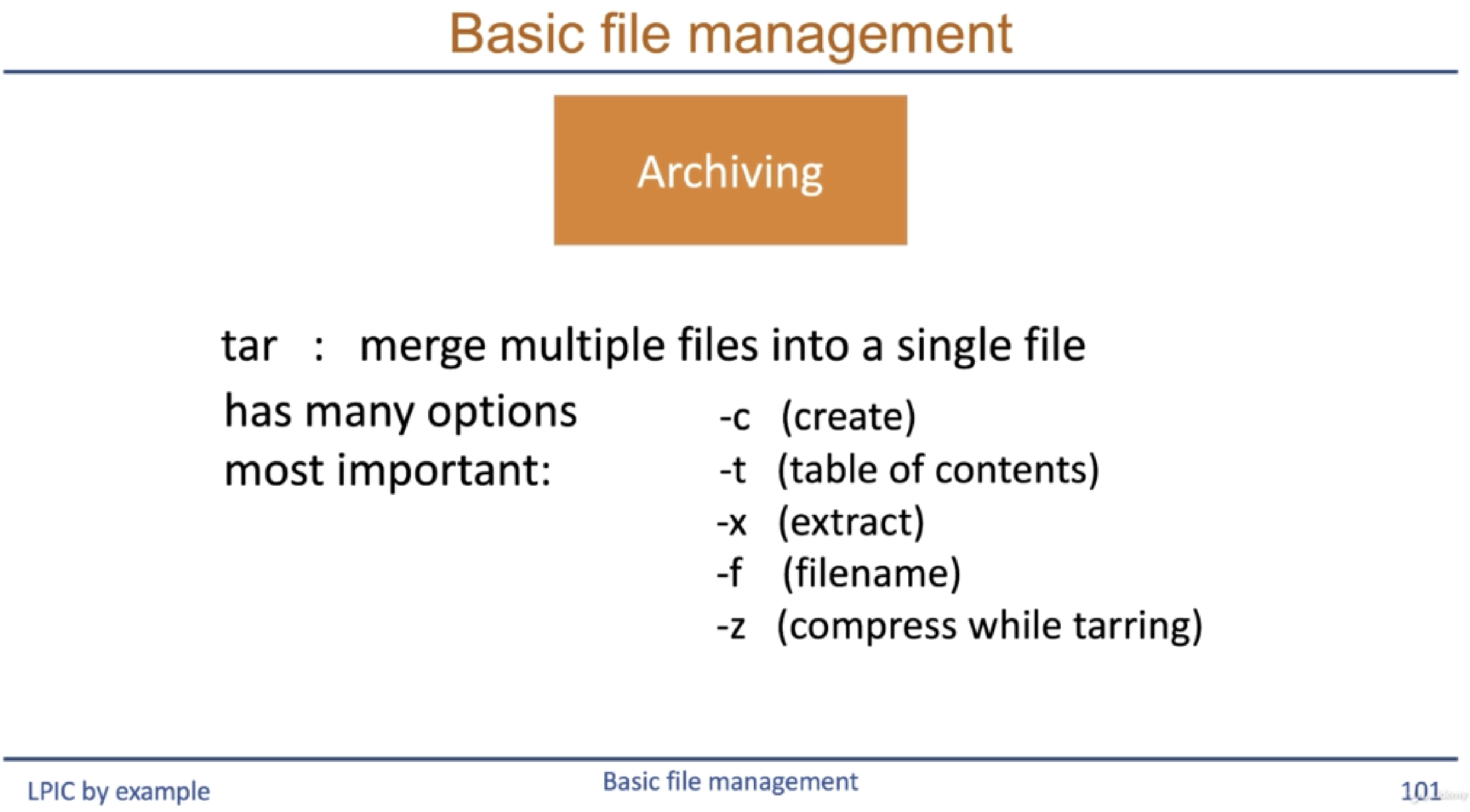
copy move remove
Files and Directories

find command

Archiving with
tar dd and cpio

Compression

file command



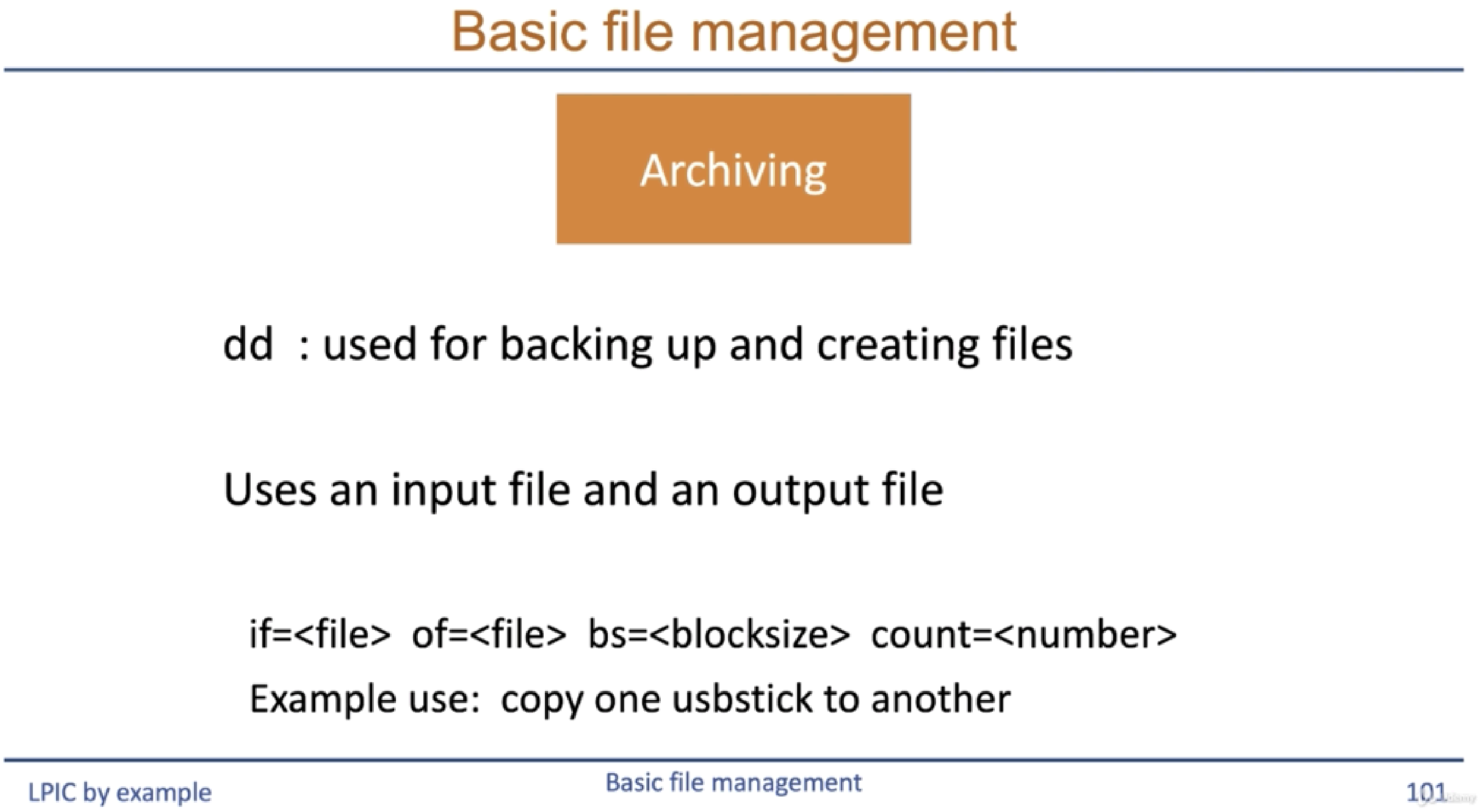
Basic file management

Archiving

tar : merge multiple files into a single file

has many options
most important:

- -c (create)
- -t (table of contents)
- -x (extract)
- -f (filename)
- -z (compress while tarring)



Basic file management

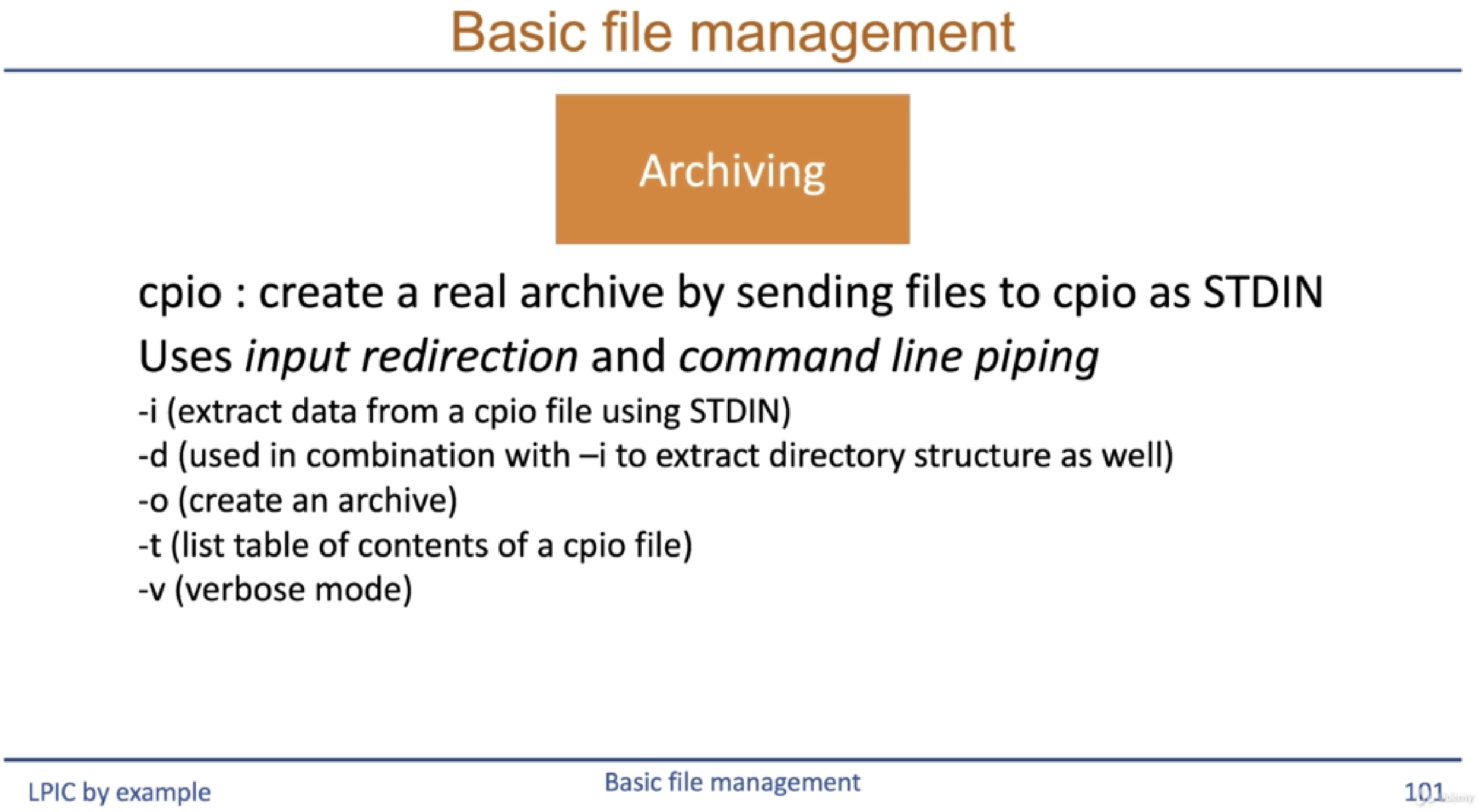
Archiving

dd : used for backing up and creating files

Uses an input file and an output file

```
if=<file>  of=<file>  bs=<blocksize>  count=<number>
```

Example use: copy one usbstick to another



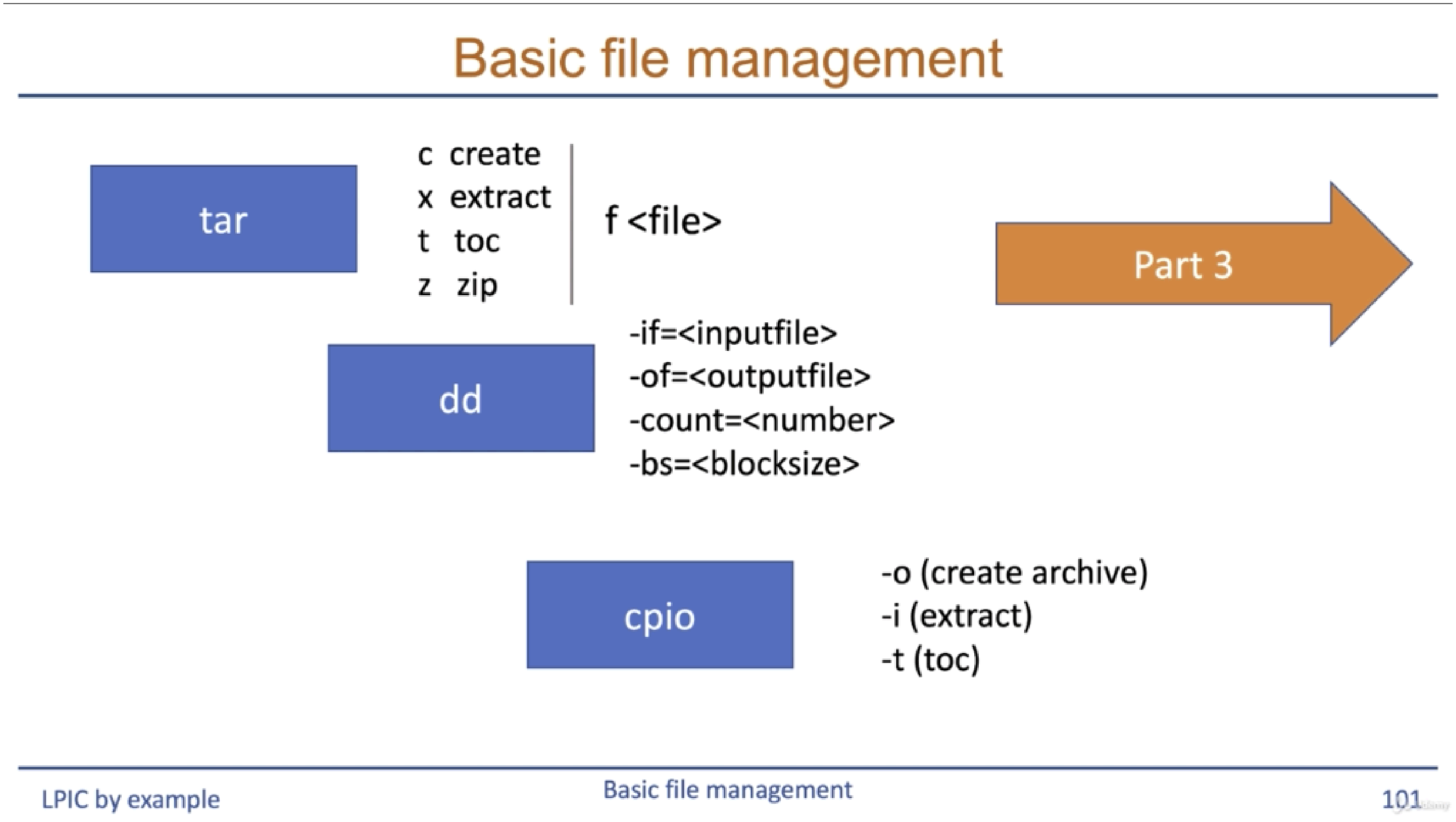
Basic file management

Archiving

cpio : create a real archive by sending files to cpio as STDIN

Uses *input redirection* and *command line piping*

-i (extract data from a cpio file using STDIN)
-d (used in combination with –i to extract directory structure as well)
-o (create an archive)
-t (list table of contents of a cpio file)
-v (verbose mode)



Basic file management

tar

c create
x extract
t toc
z zip

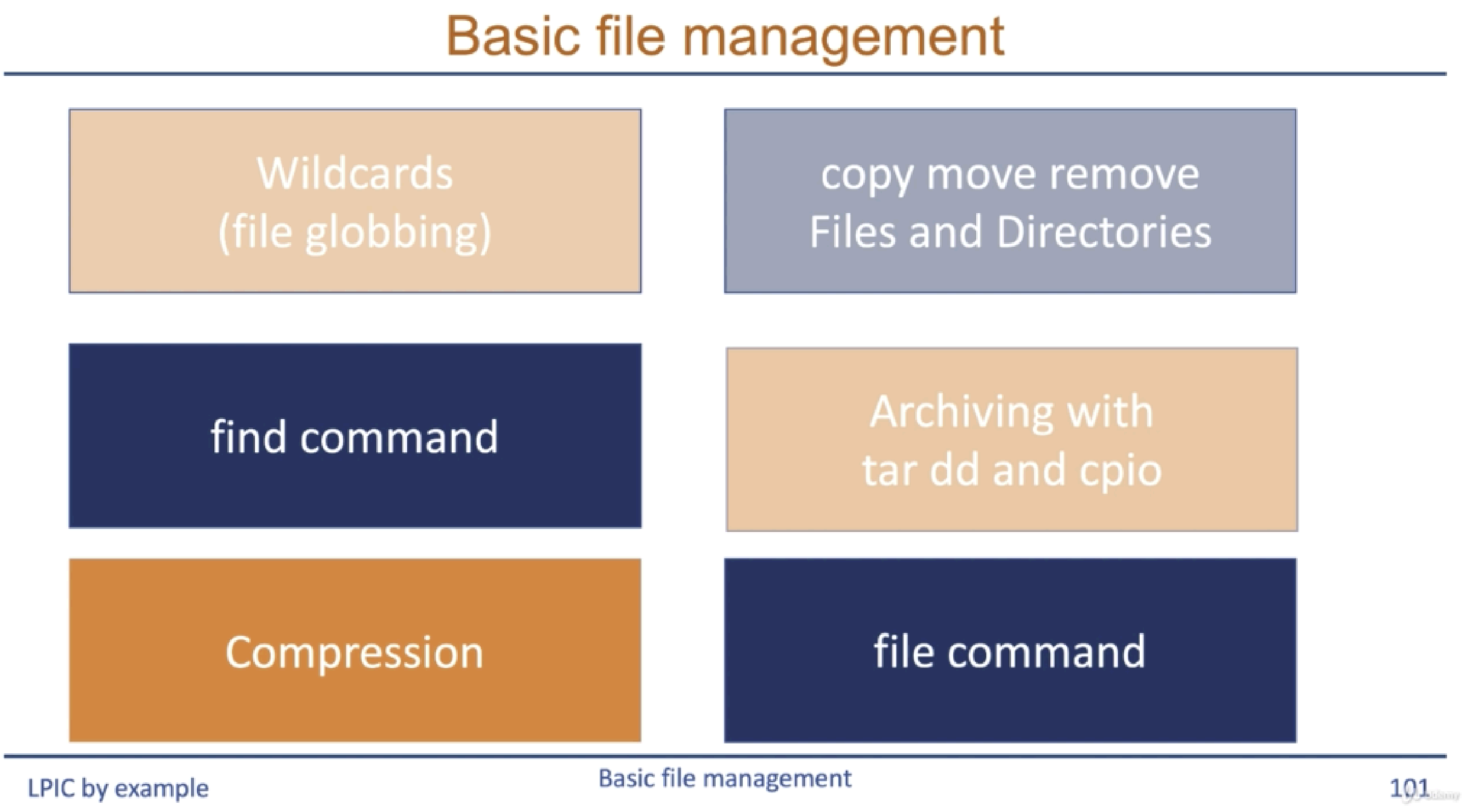
f <file>

dd

-if=<inputfile>
-of=<outputfile>
-count=<number>
-bs=<blocksize>

cpio

-o (create archive)
-i (extract)
-t (toc)



Basic file management

Wildcards
(file globbing)

copy move remove
Files and Directories

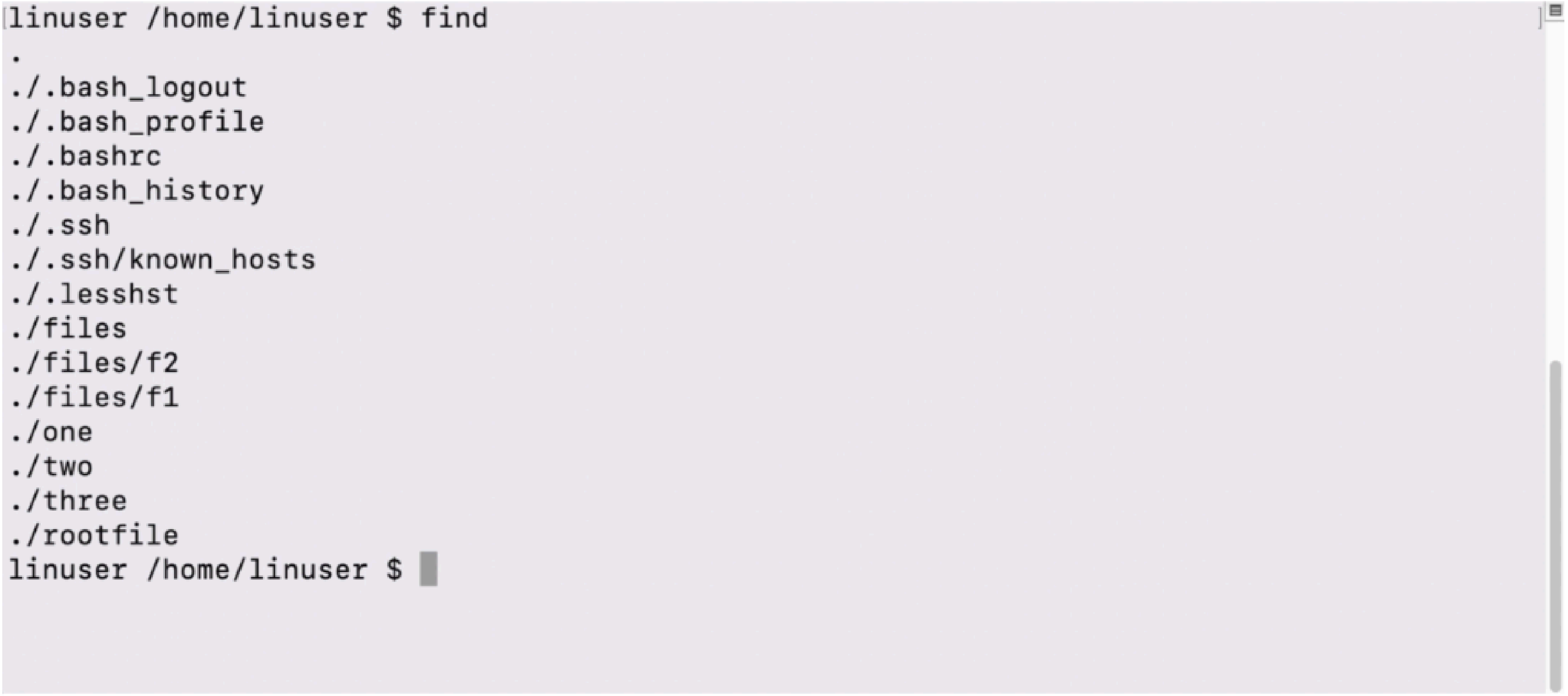
find command

Archiving with
tar dd and cpio

Compression

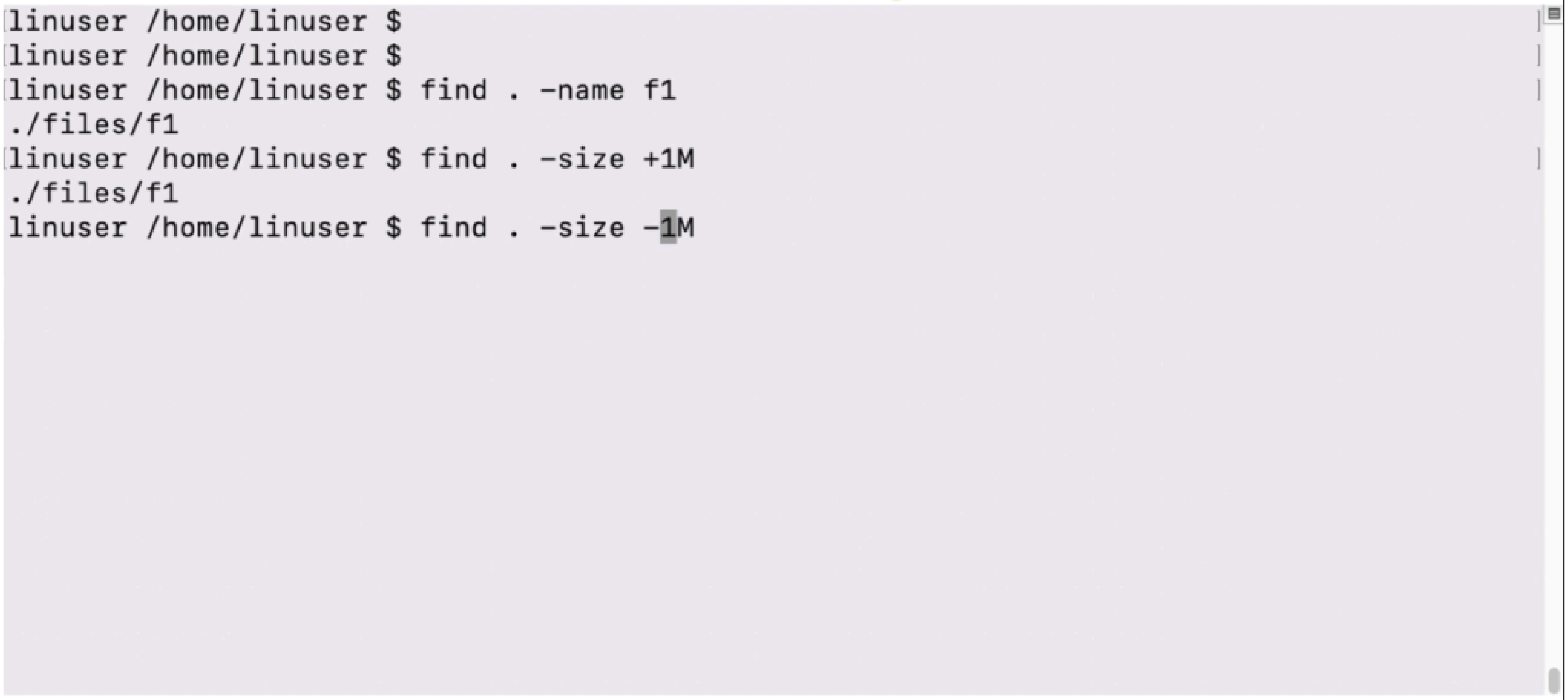
file command

Basic file management



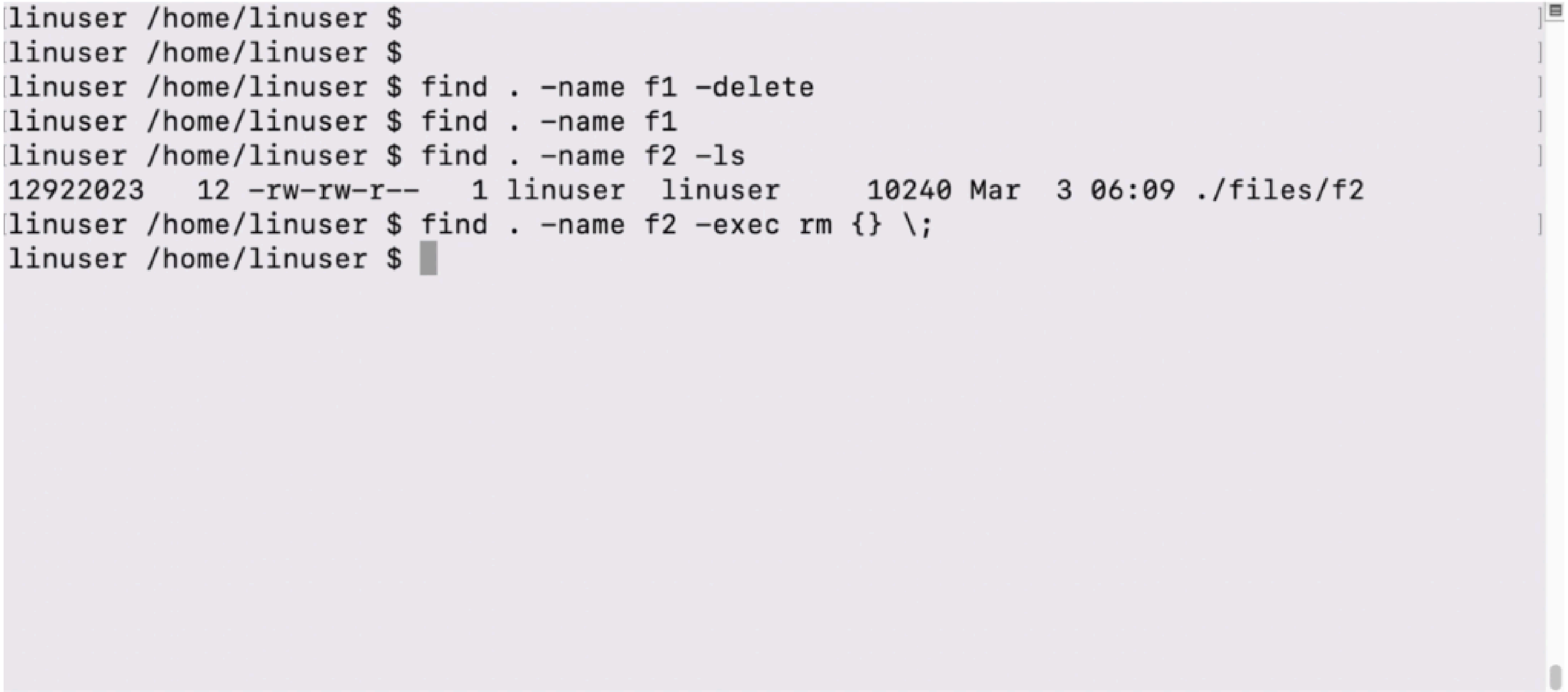
```
linuser /home/linuser $ find
.
./.bash_logout
./.bash_profile
./.bashrc
./.bash_history
./.ssh
./.ssh/known_hosts
./.lesshst
./files
./files/f2
./files/f1
./one
./two
./three
./rootfile
linuser /home/linuser $ 
```

Basic file management

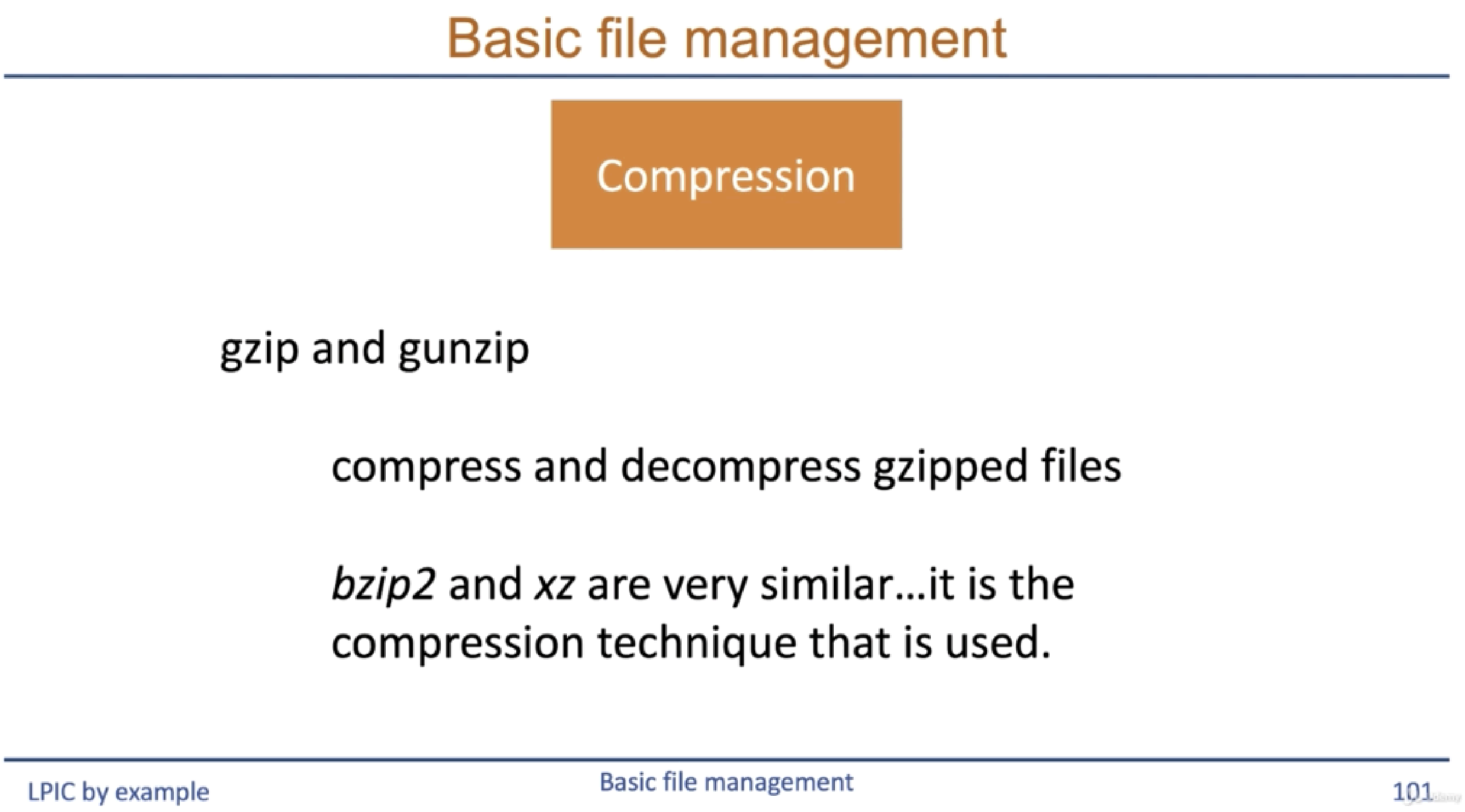


```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ find . -name f1
./files/f1
linuser /home/linuser $ find . -size +1M
./files/f1
linuser /home/linuser $ find . -size -1M
```

Basic file management



```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ find . -name f1 -delete
linuser /home/linuser $ find . -name f1
linuser /home/linuser $ find . -name f2 -ls
12922023   12 -rw-rw-r--   1 linuser  linuser     10240 Mar  3 06:09 ./files/f2
linuser /home/linuser $ find . -name f2 -exec rm {} \;
linuser /home/linuser $ 
```



Basic file management

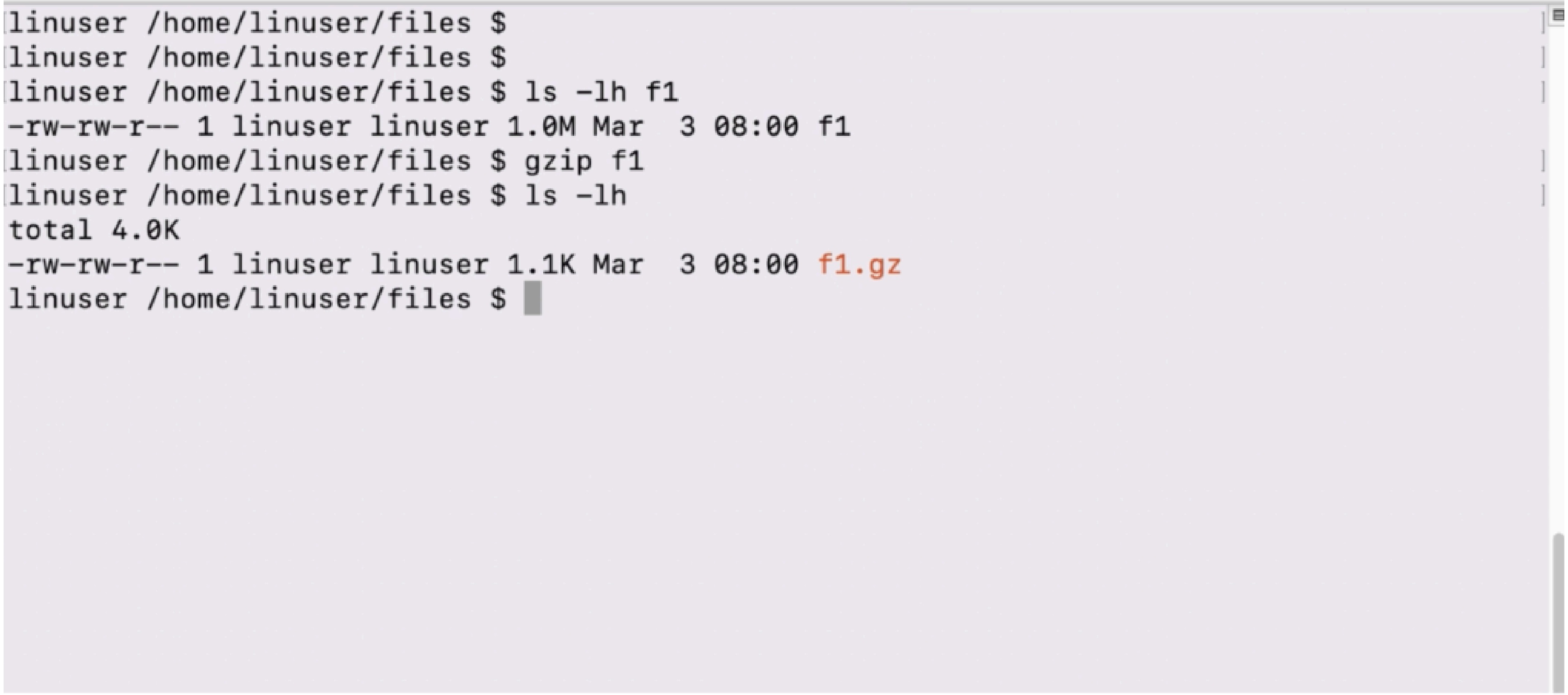
Compression

gzip and gunzip

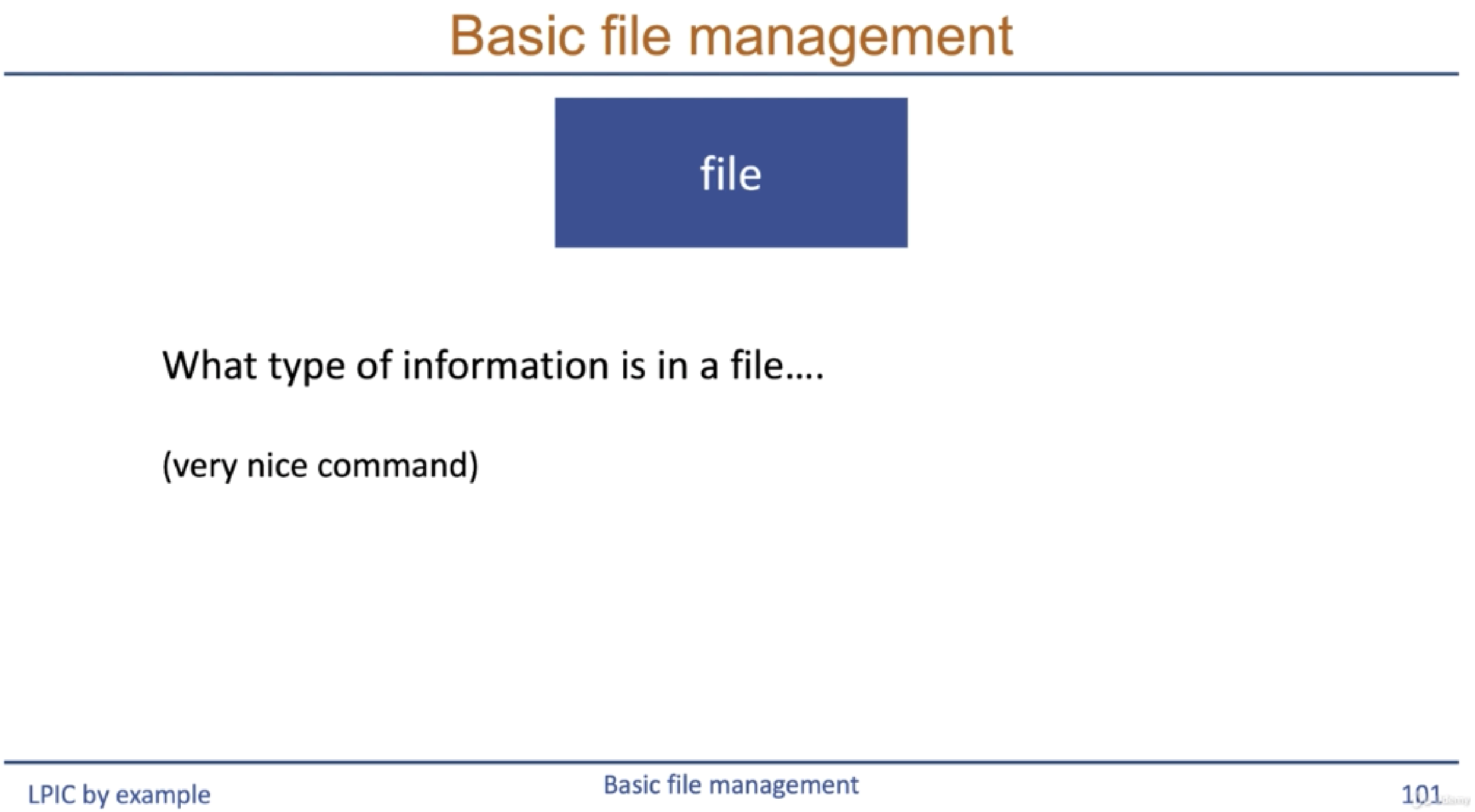
compress and decompress gzipped files

bzip2 and *xz* are very similar…it is the compression technique that is used.

Basic file management



```
linuser /home/linuser/files $
linuser /home/linuser/files $
linuser /home/linuser/files $ ls -lh f1
-rw-rw-r-- 1 linuser linuser 1.0M Mar  3 08:00 f1
linuser /home/linuser/files $ gzip f1
linuser /home/linuser/files $ ls -lh
total 4.0K
-rw-rw-r-- 1 linuser linuser 1.1K Mar  3 08:00 f1.gz
linuser /home/linuser/files $
```



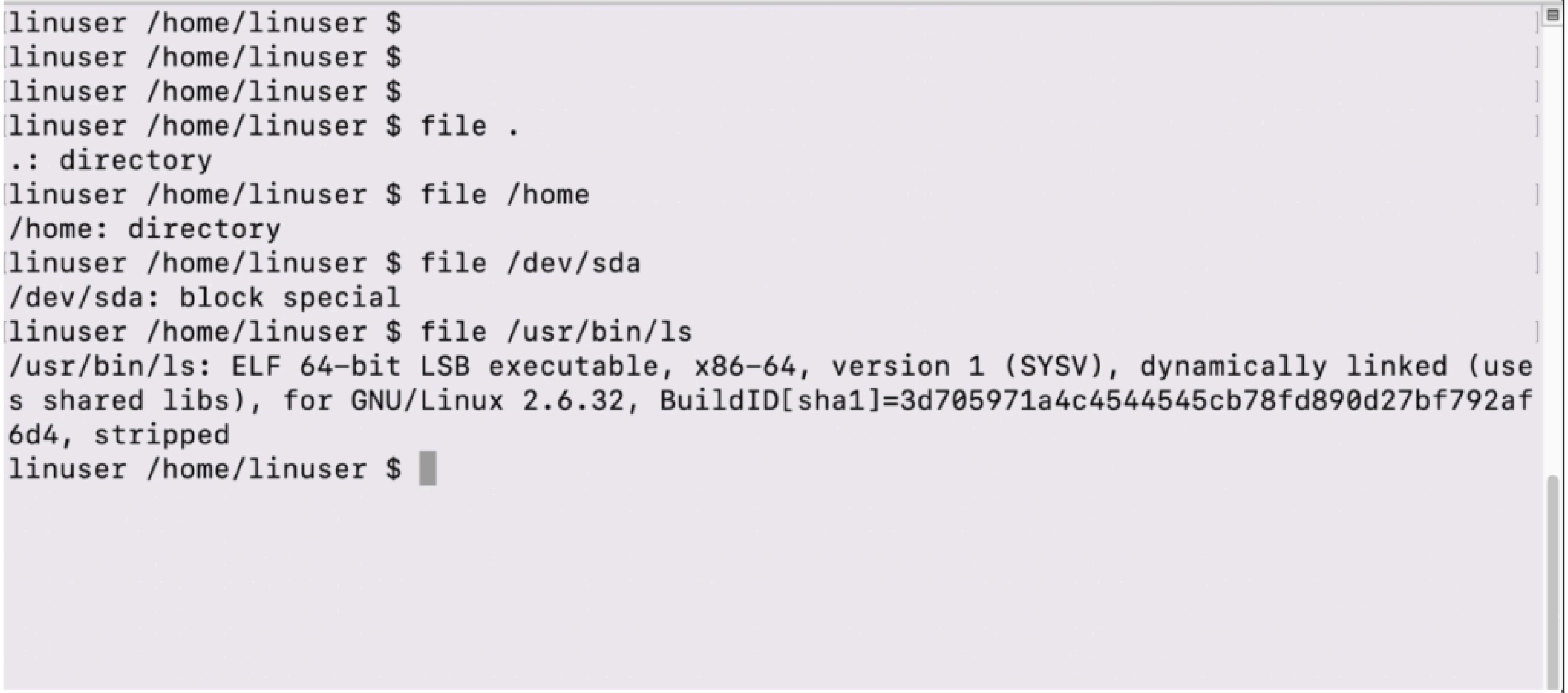
Basic file management

file

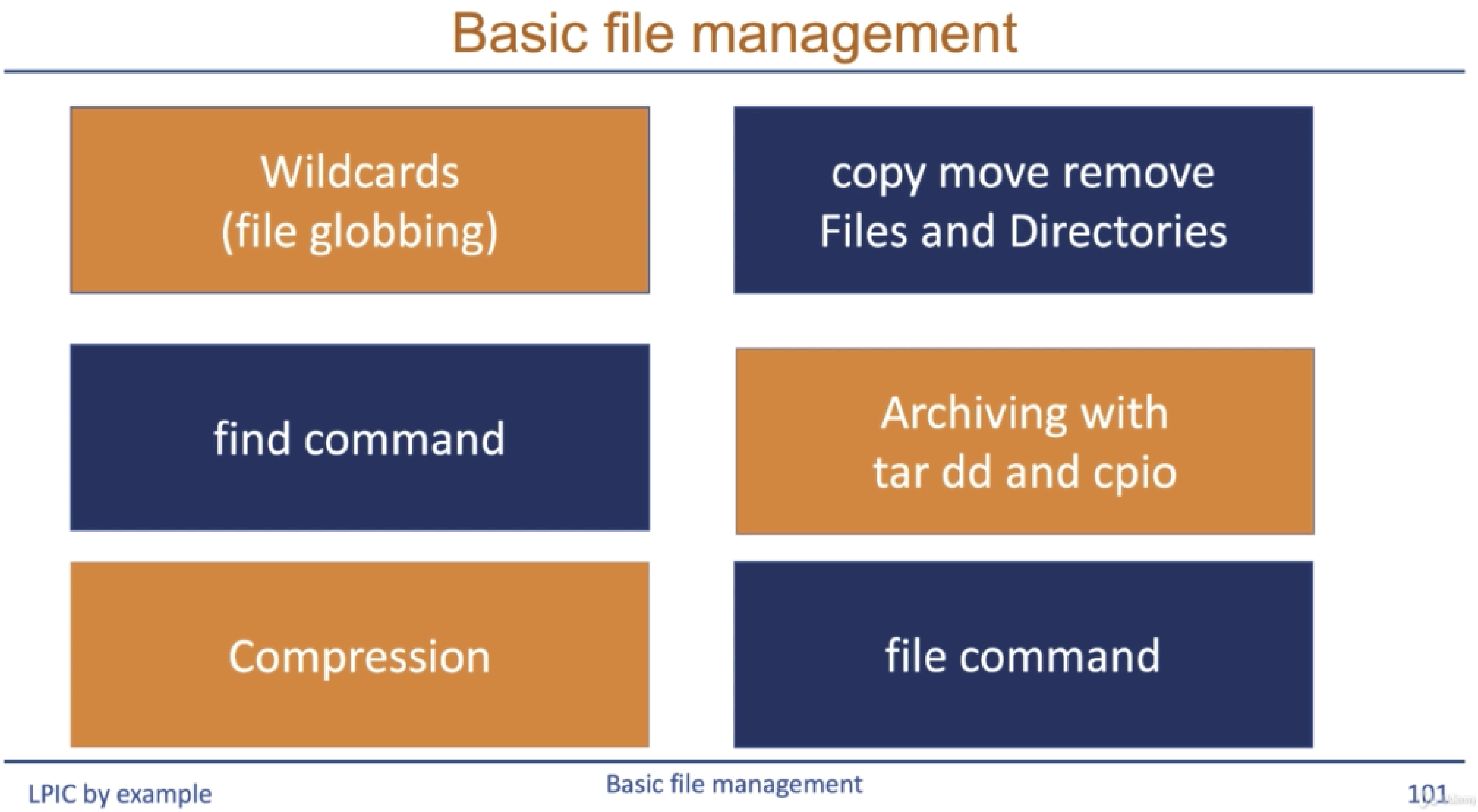
What type of information is in a file….

(very nice command)

Basic file management



```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ file .
.: directory
linuser /home/linuser $ file /home
/home: directory
linuser /home/linuser $ file /dev/sda
/dev/sda: block special
linuser /home/linuser $ file /usr/bin/ls
/usr/bin/ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (use
s shared libs), for GNU/Linux 2.6.32, BuildID[sha1]=3d705971a4c4544545cb78fd890d27bf792af
6d4, stripped
linuser /home/linuser $ 
```



Basic file management

Wildcards
(file globbing)

copy move remove
Files and Directories

find command

Archiving with
tar dd and cpio

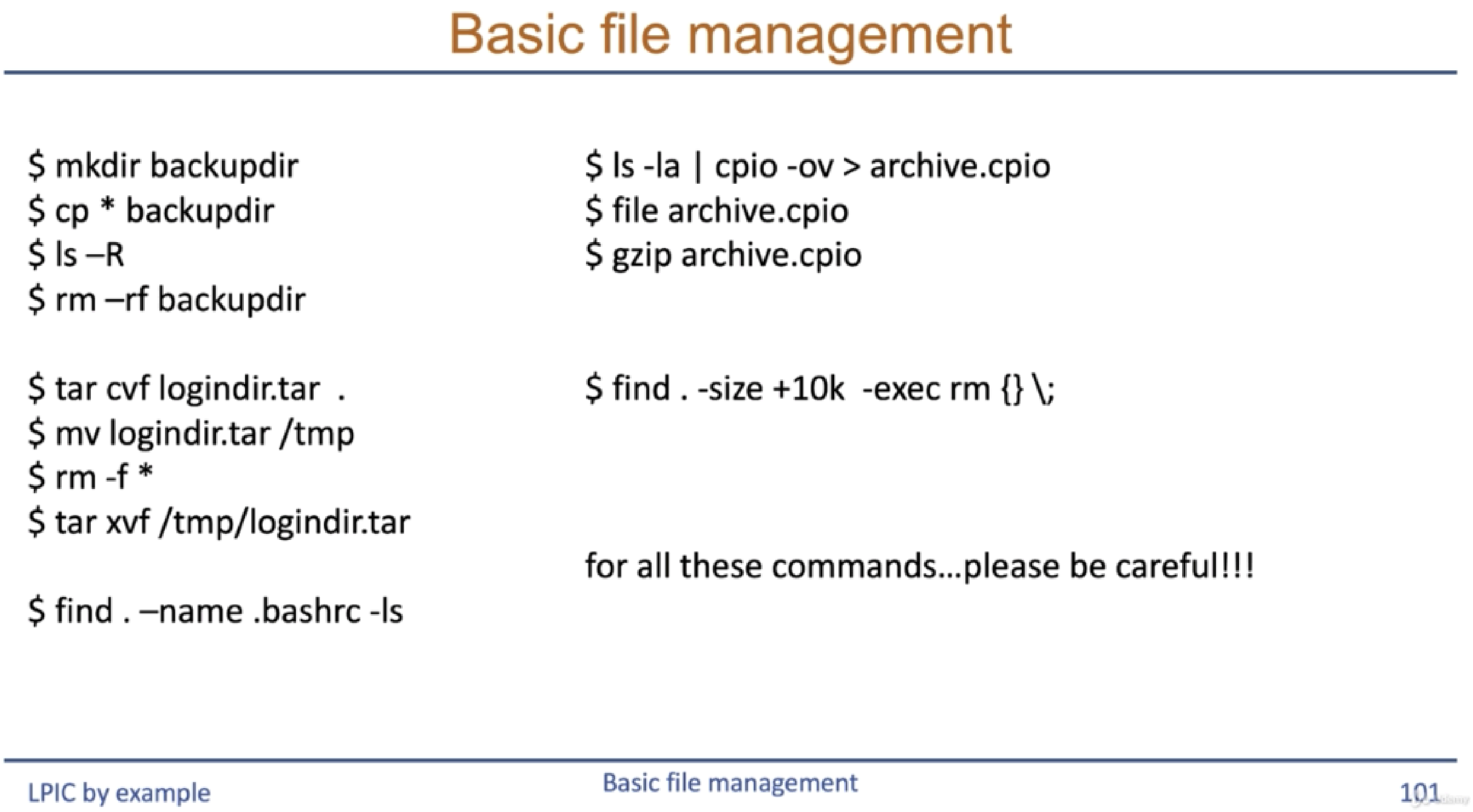
Compression

file command

LPIC by example

Basic file management

101



Basic file management

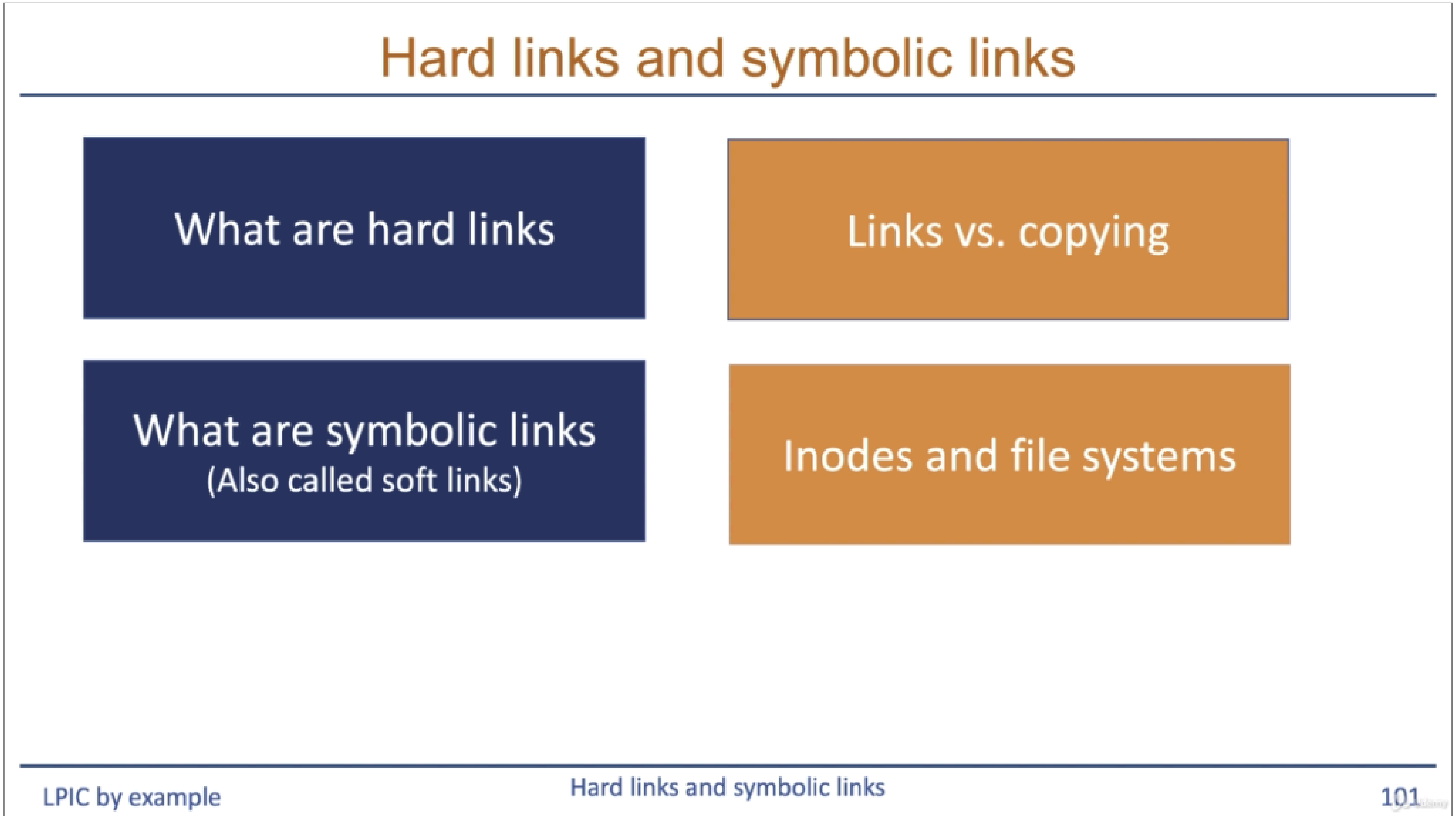
```
$ mkdir backupdir
$ cp * backupdir
$ ls –R
$ rm –rf backupdir

$ tar cvf logindir.tar  .
$ mv logindir.tar /tmp
$ rm -f *
$ tar xvf /tmp/logindir.tar

$ find . –name .bashrc -ls
```

```
$ ls -la | cpio -ov > archive.cpio
$ file archive.cpio
$ gzip archive.cpio

$ find . -size +10k  -exec rm {} \;
```

for all these commands…please be careful!!!

Hard links and symbolic links (part 1)

LPIC by example



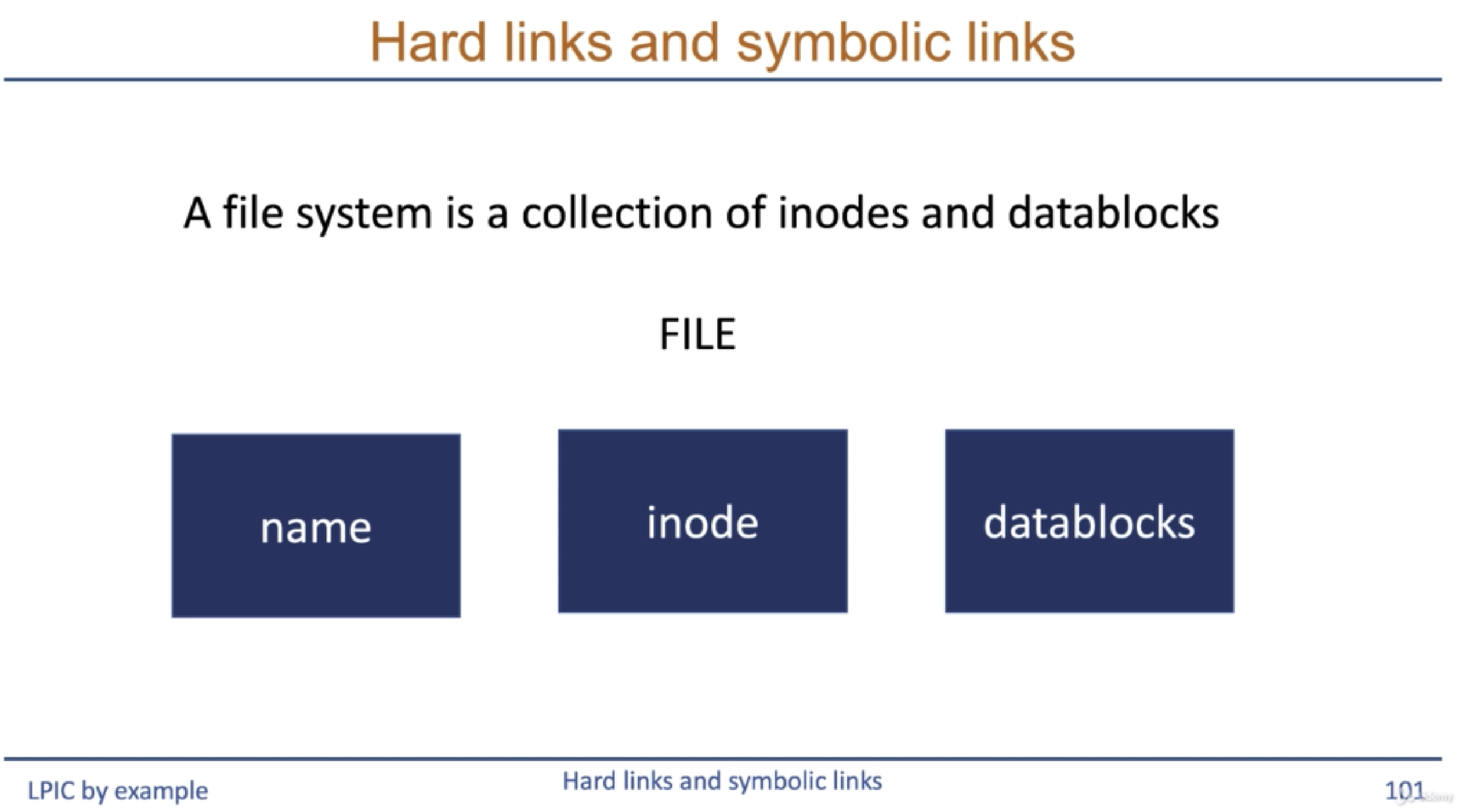
Hard links and symbolic links

What are hard links

Links vs. copying

What are symbolic links
(Also called soft links)

Inodes and file systems



Hard links and symbolic links

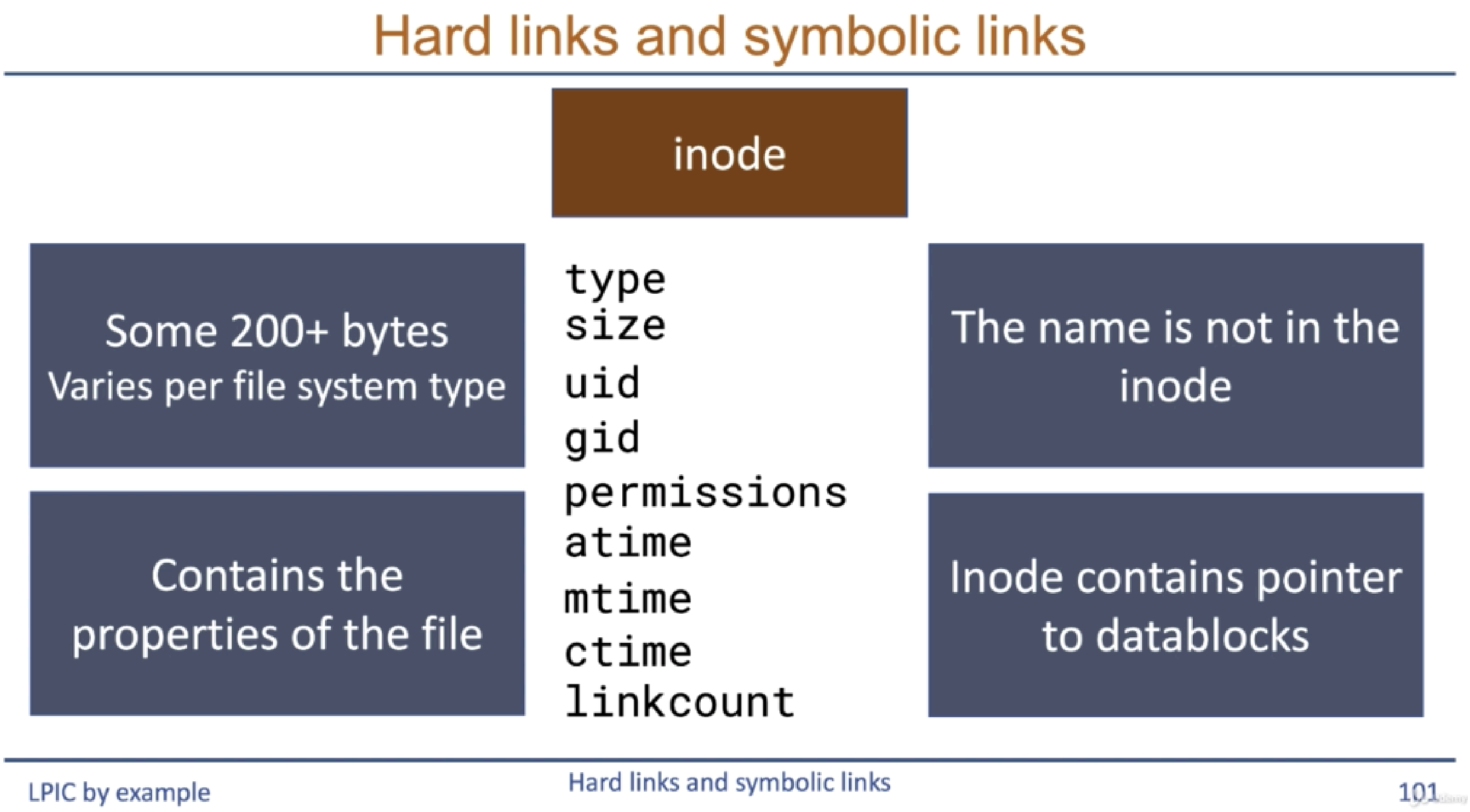
A file system is a collection of inodes and datablocks

FILE

name

inode

datablocks



Hard links and symbolic links

inode

Some 200+ bytes
Varies per file system type

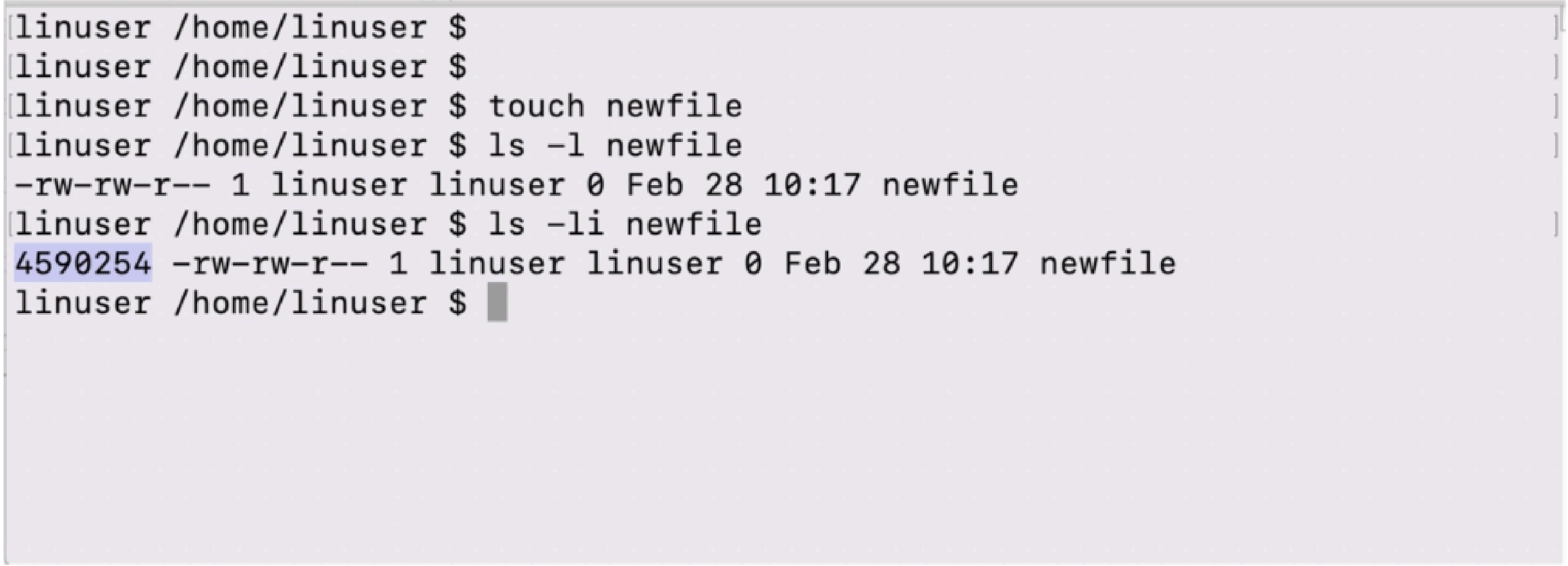
Contains the properties of the file

```
type
size
uid
gid
permissions
atime
mtime
ctime
linkcount
```

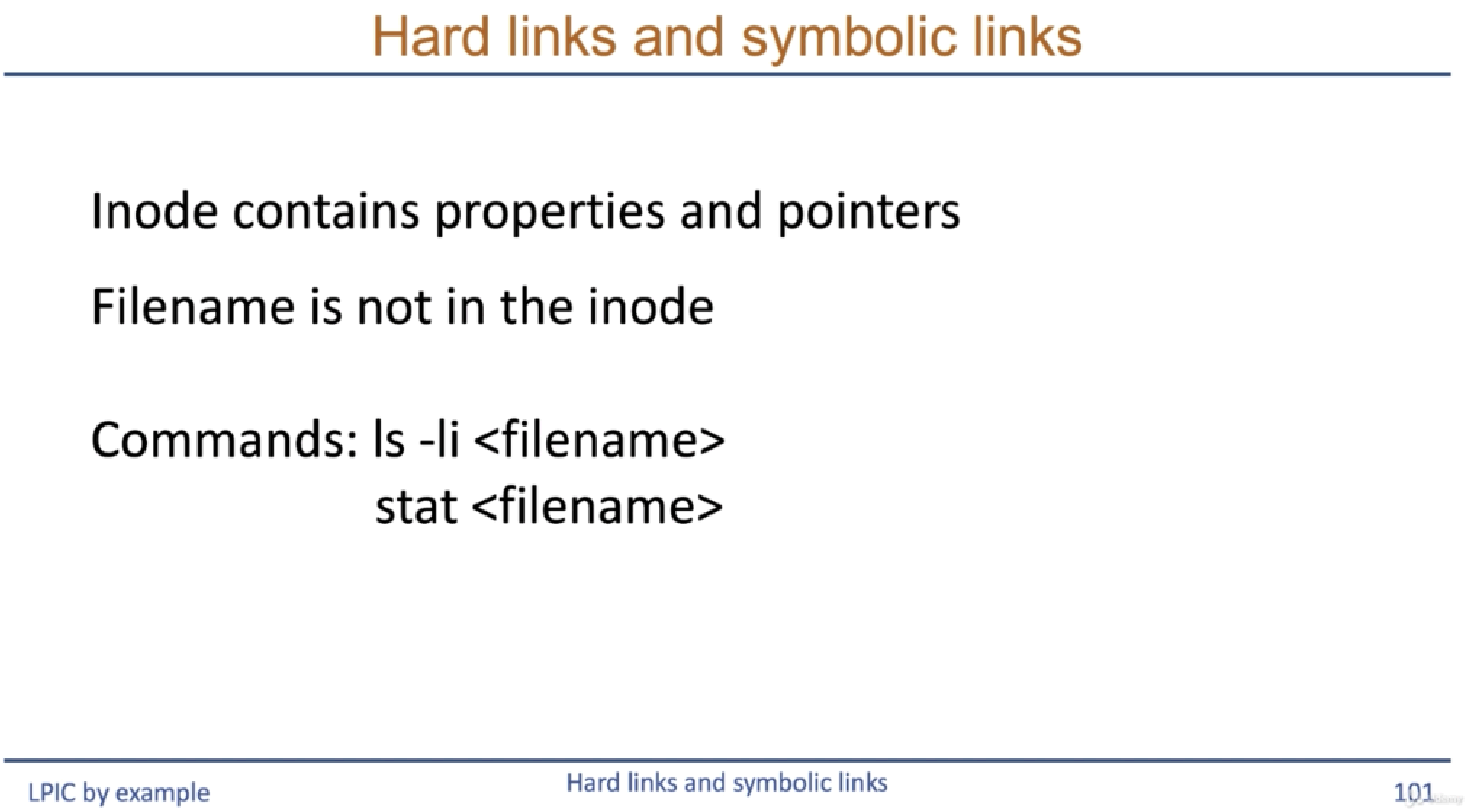
The name is not in the inode

Inode contains pointer to datablocks

Hard links and symbolic links



```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ touch newfile
linuser /home/linuser $ ls -l newfile
-rw-rw-r-- 1 linuser linuser 0 Feb 28 10:17 newfile
linuser /home/linuser $ ls -li newfile
4590254 -rw-rw-r-- 1 linuser linuser 0 Feb 28 10:17 newfile
linuser /home/linuser $
```

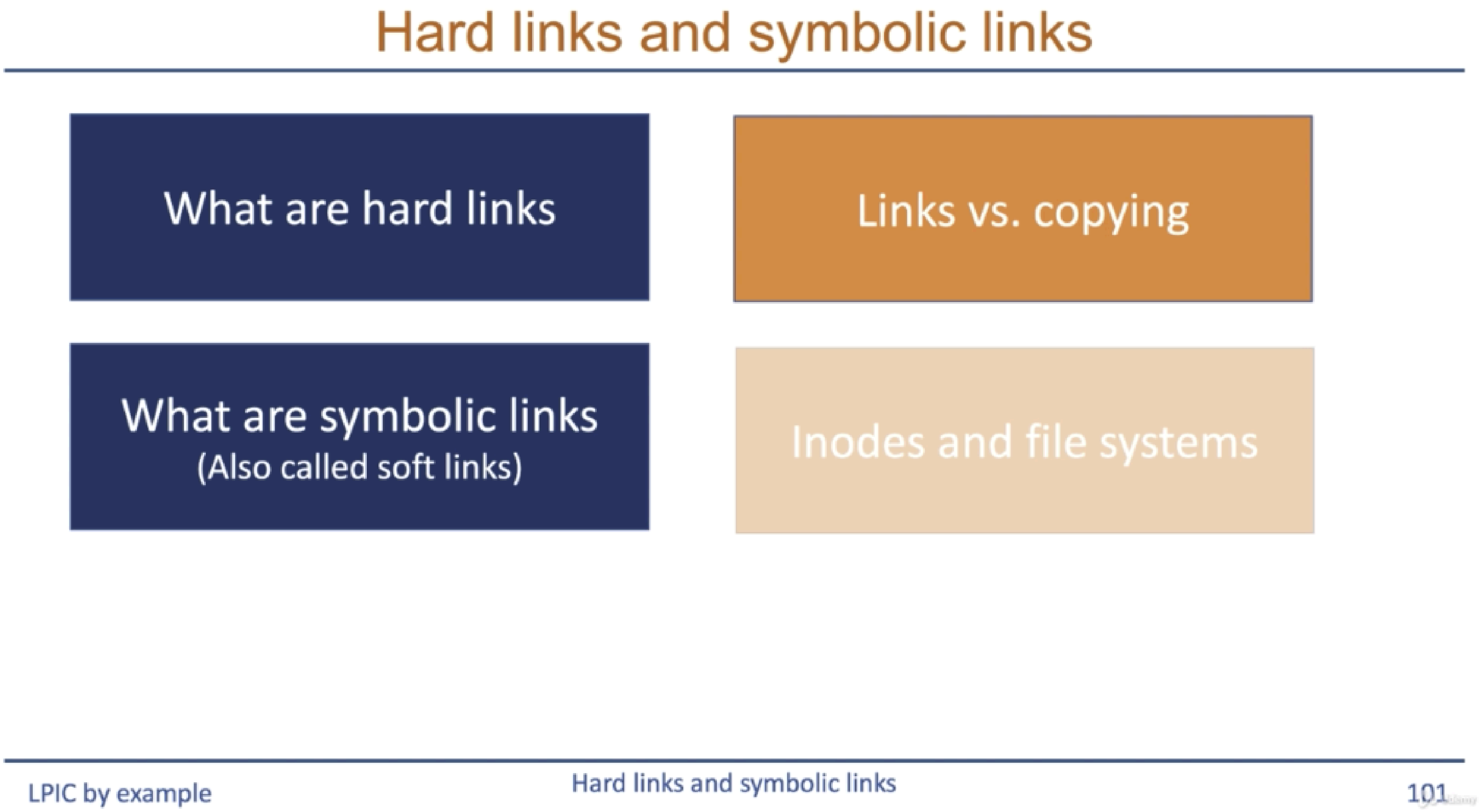


Hard links and symbolic links

Inode contains properties and pointers

Filename is not in the inode

Commands: ls -li <filename>
stat <filename>



Hard links and symbolic links

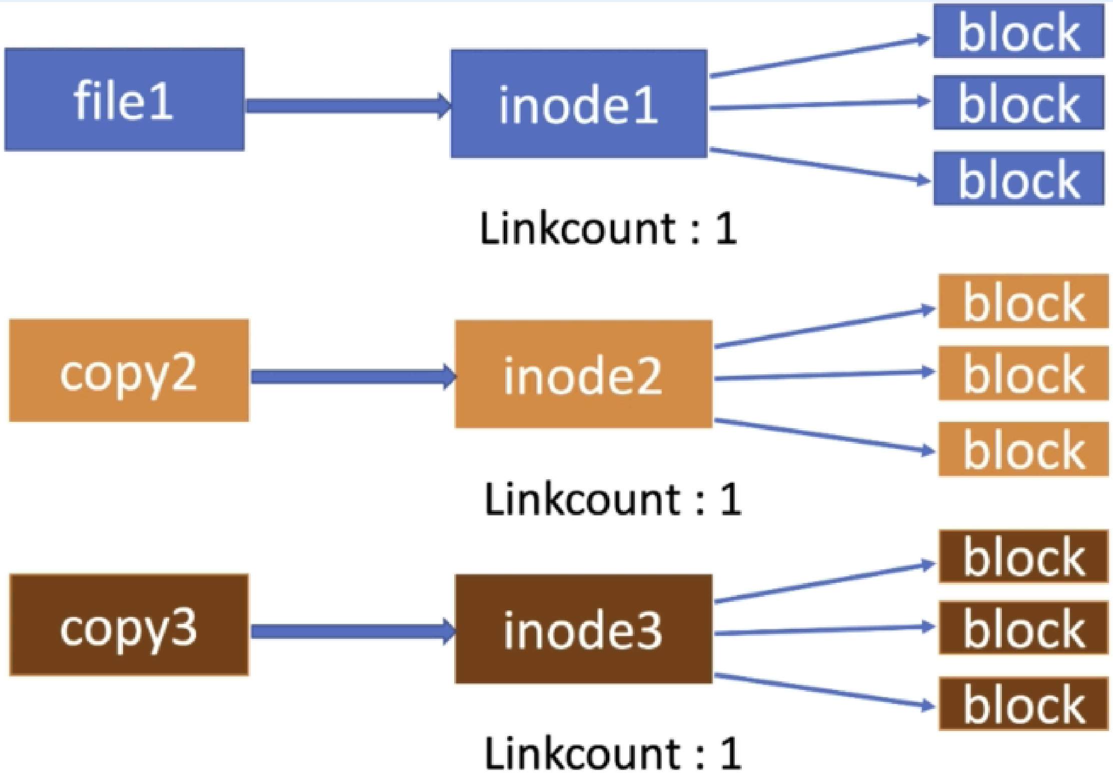
What are hard links

Links vs. copying

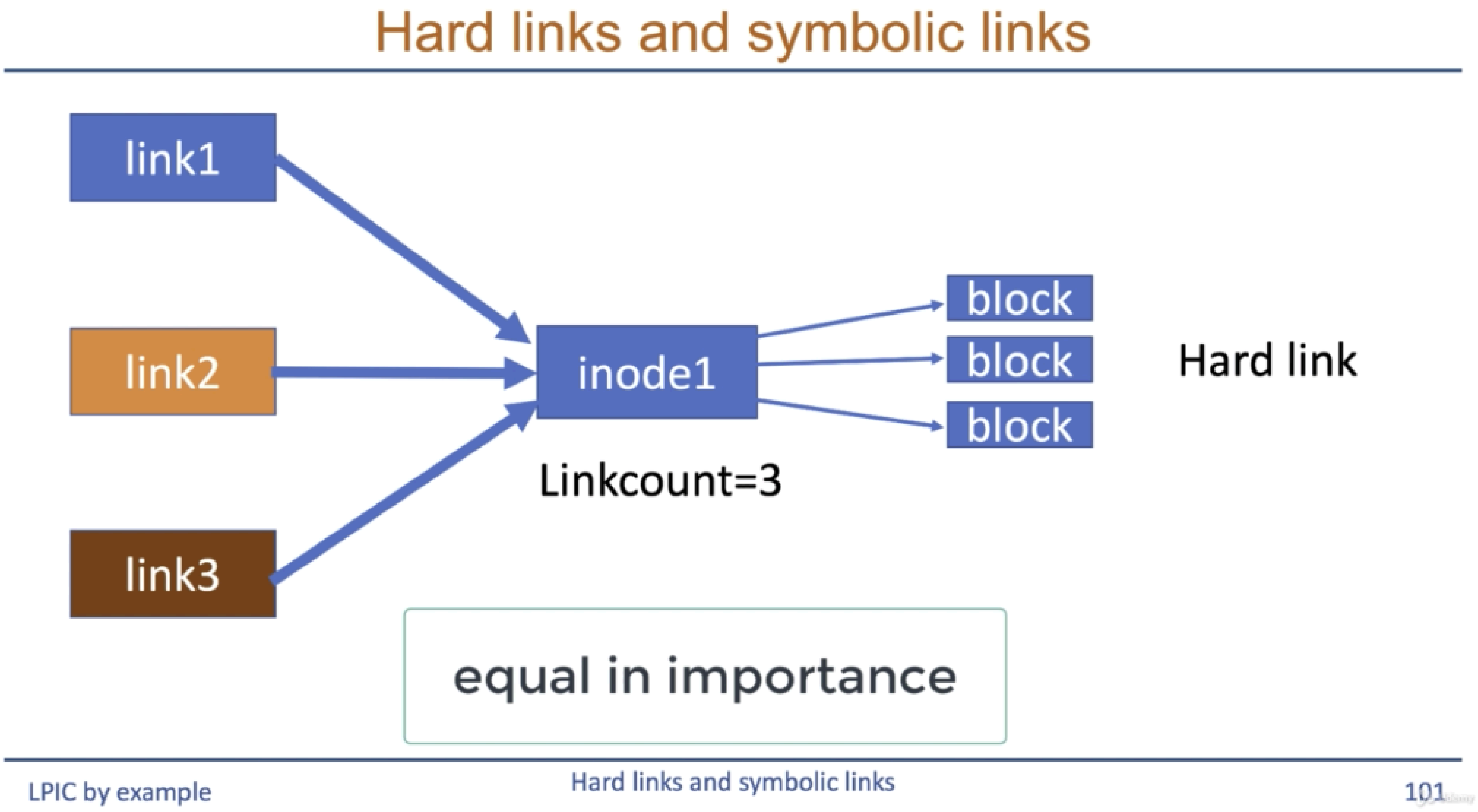
What are symbolic links
(Also called soft links)

Inodes and file systems

Hard links and symbolic links



Copy a file



Hard links and symbolic links

link1

link2

link3

inode1

block

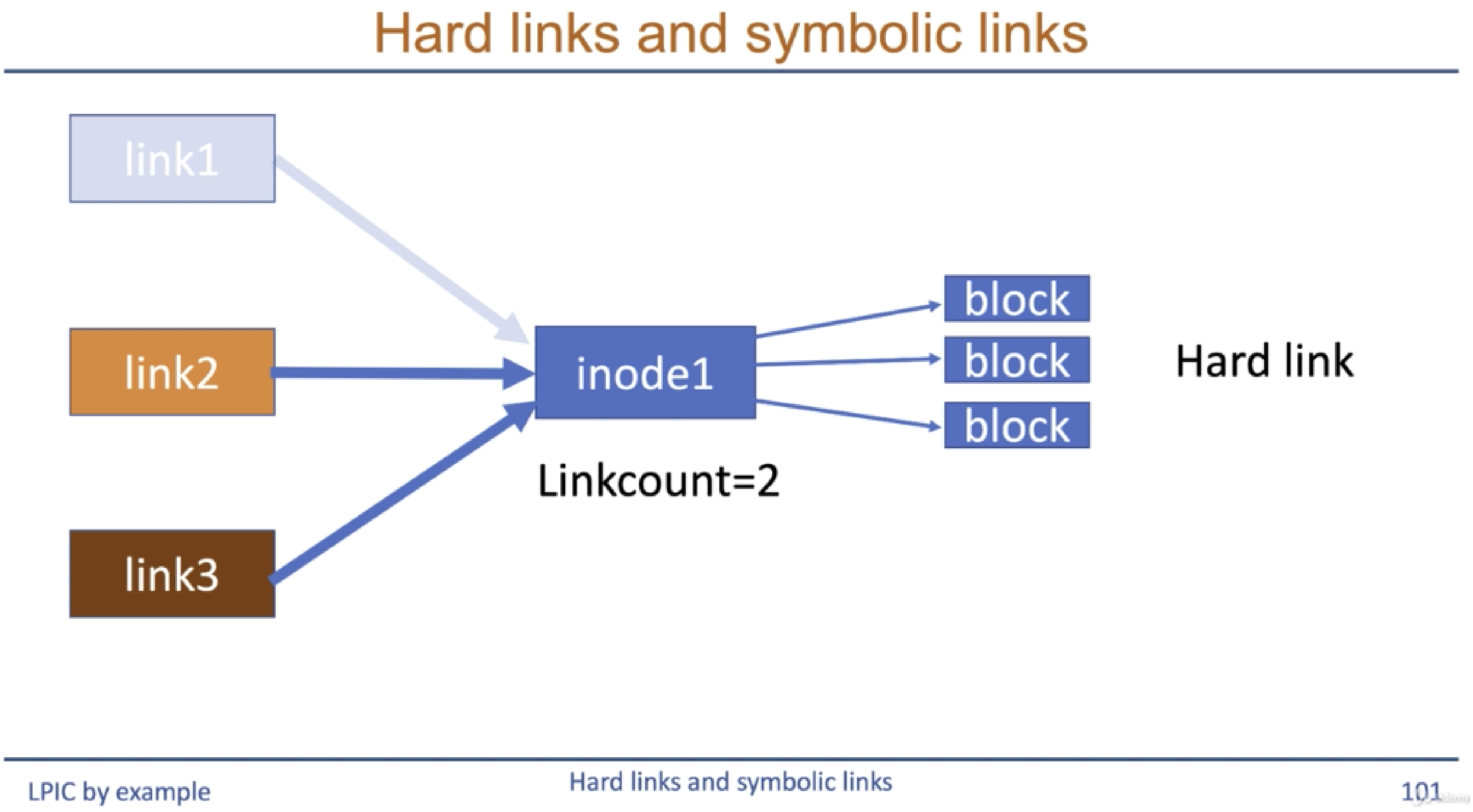
block

block

Linkcount=3

Hard link

equal in importance



Hard links and symbolic links

link1

link2

link3

inode1

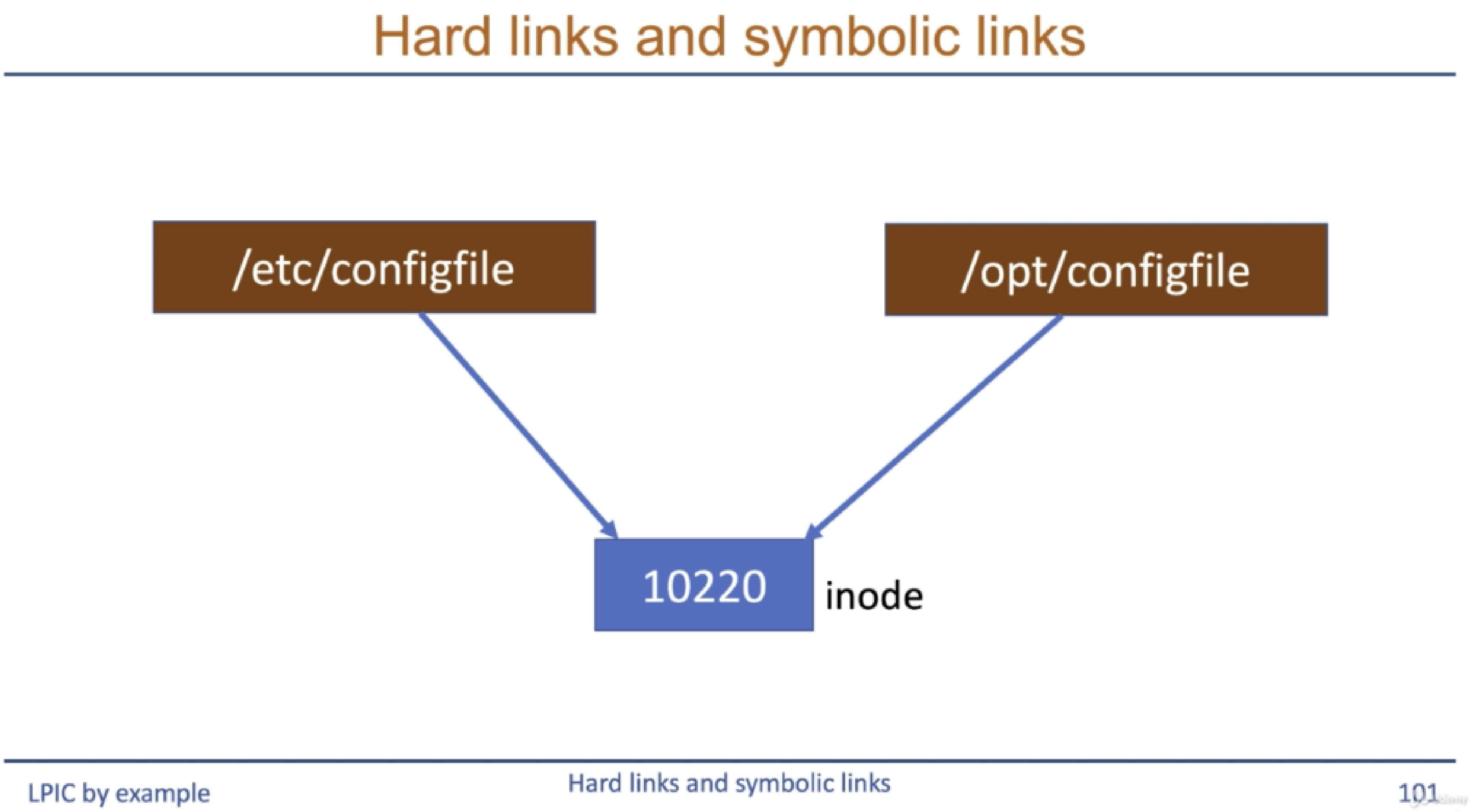
Linkcount=2

block

block

block

Hard link



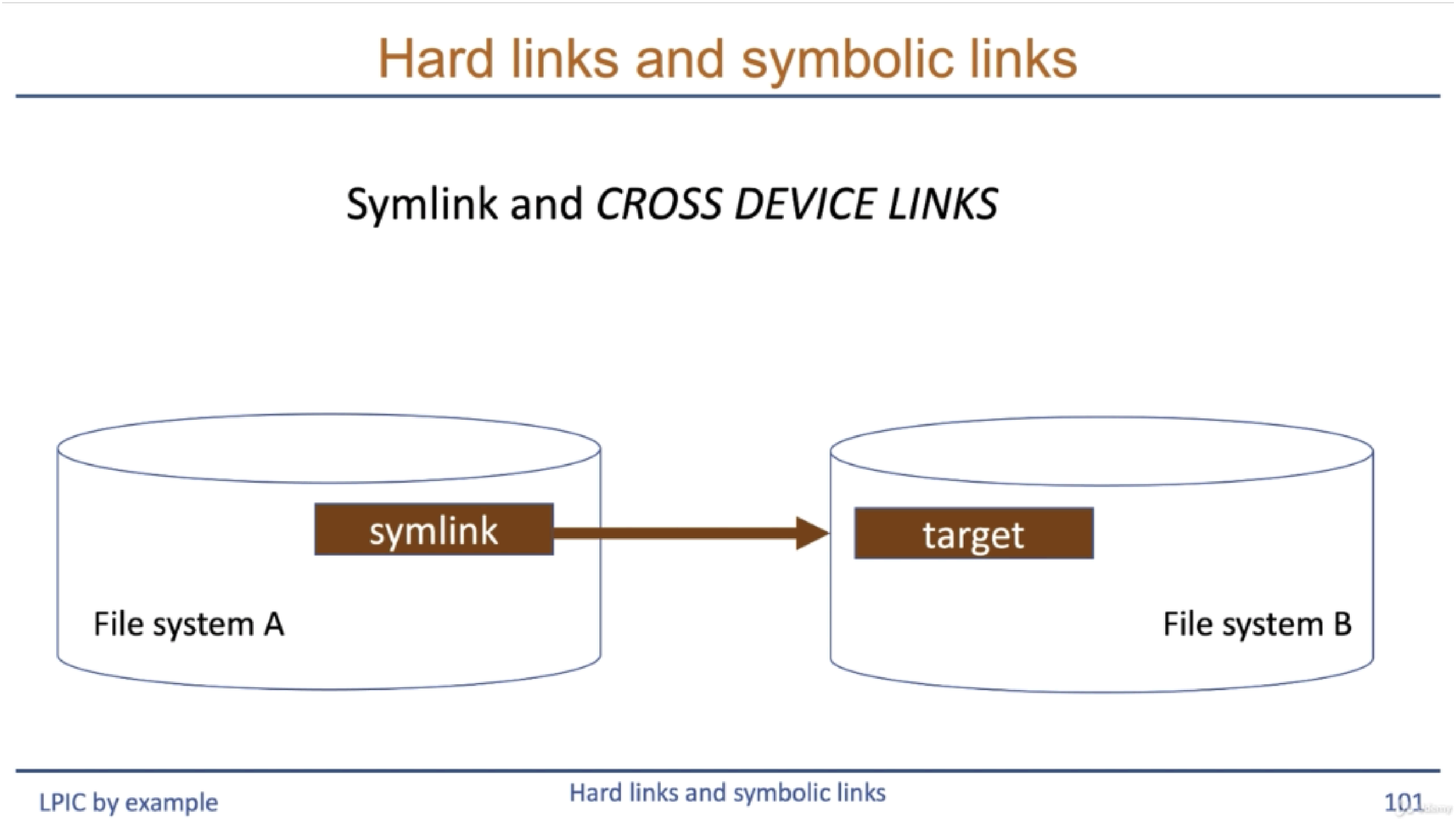
Hard links and symbolic links

/etc/configfile

/opt/configfile

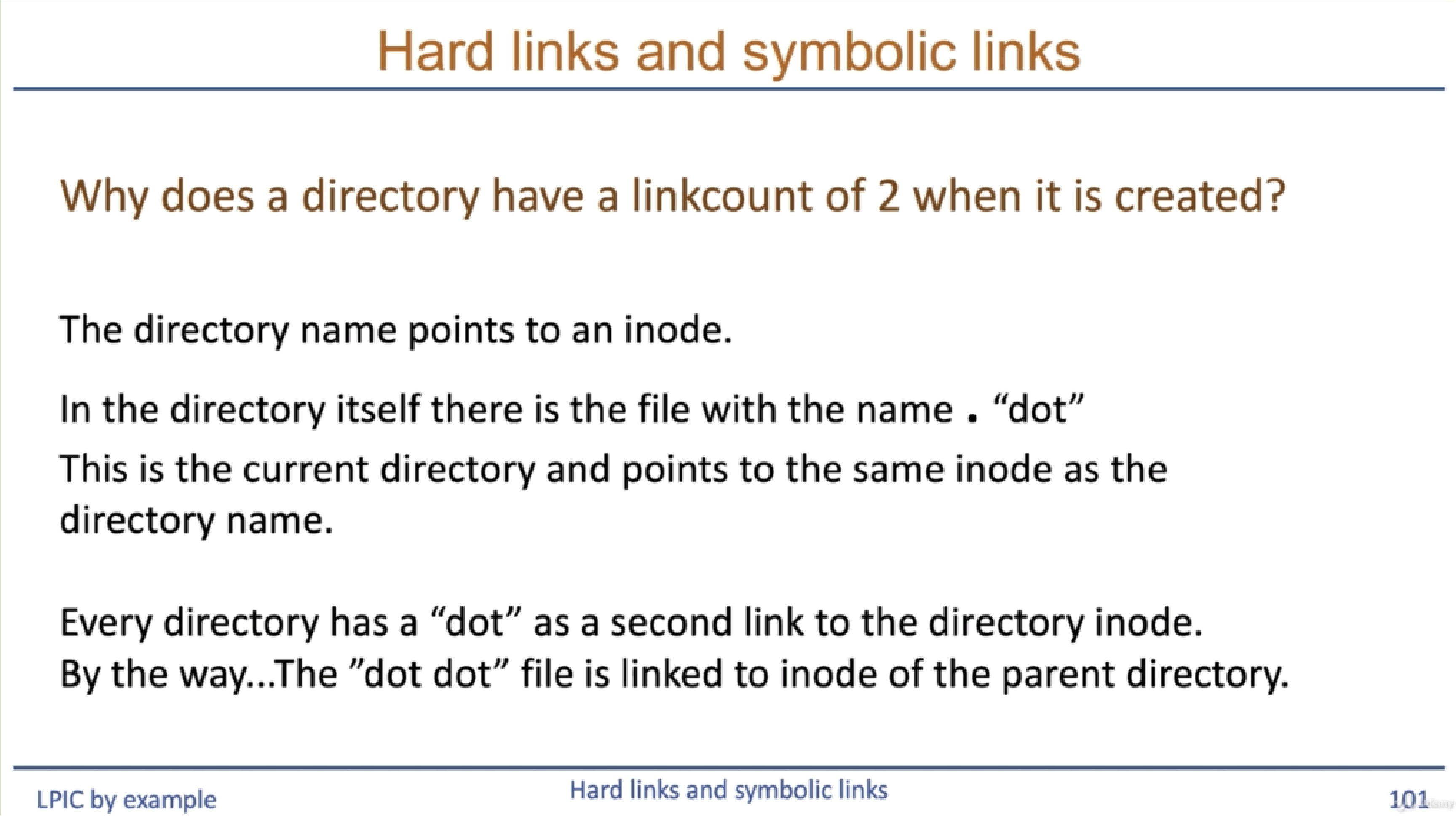
10220

inode



Hard links and symbolic links

Symlink and *CROSS DEVICE LINKS*



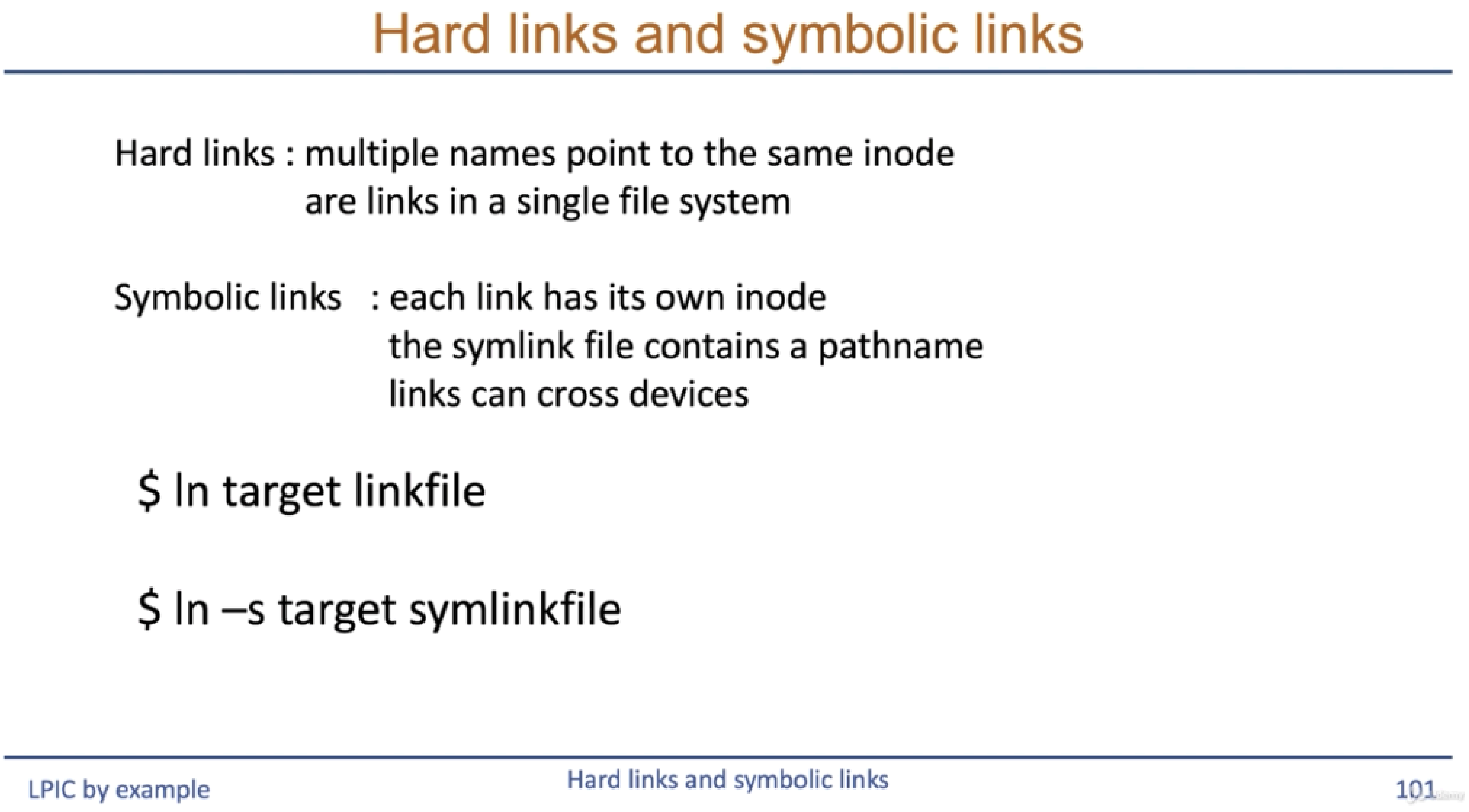
Hard links and symbolic links

Why does a directory have a linkcount of 2 when it is created?

The directory name points to an inode.

In the directory itself there is the file with the name . "dot"
This is the current directory and points to the same inode as the directory name.

Every directory has a "dot" as a second link to the directory inode.
By the way...The "dot dot" file is linked to inode of the parent directory.

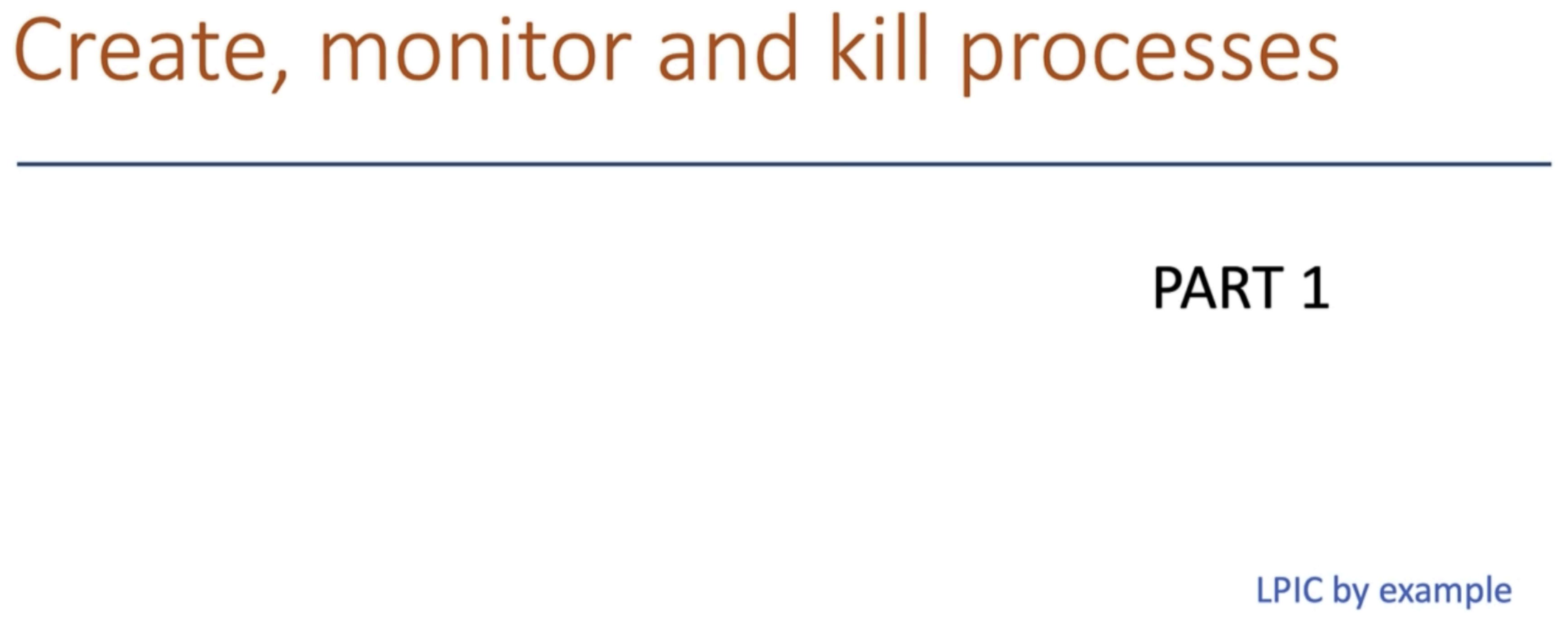


Hard links and symbolic links

Hard links : multiple names point to the same inode
are links in a single file system

Symbolic links : each link has its own inode
the symlink file contains a pathname
links can cross devices

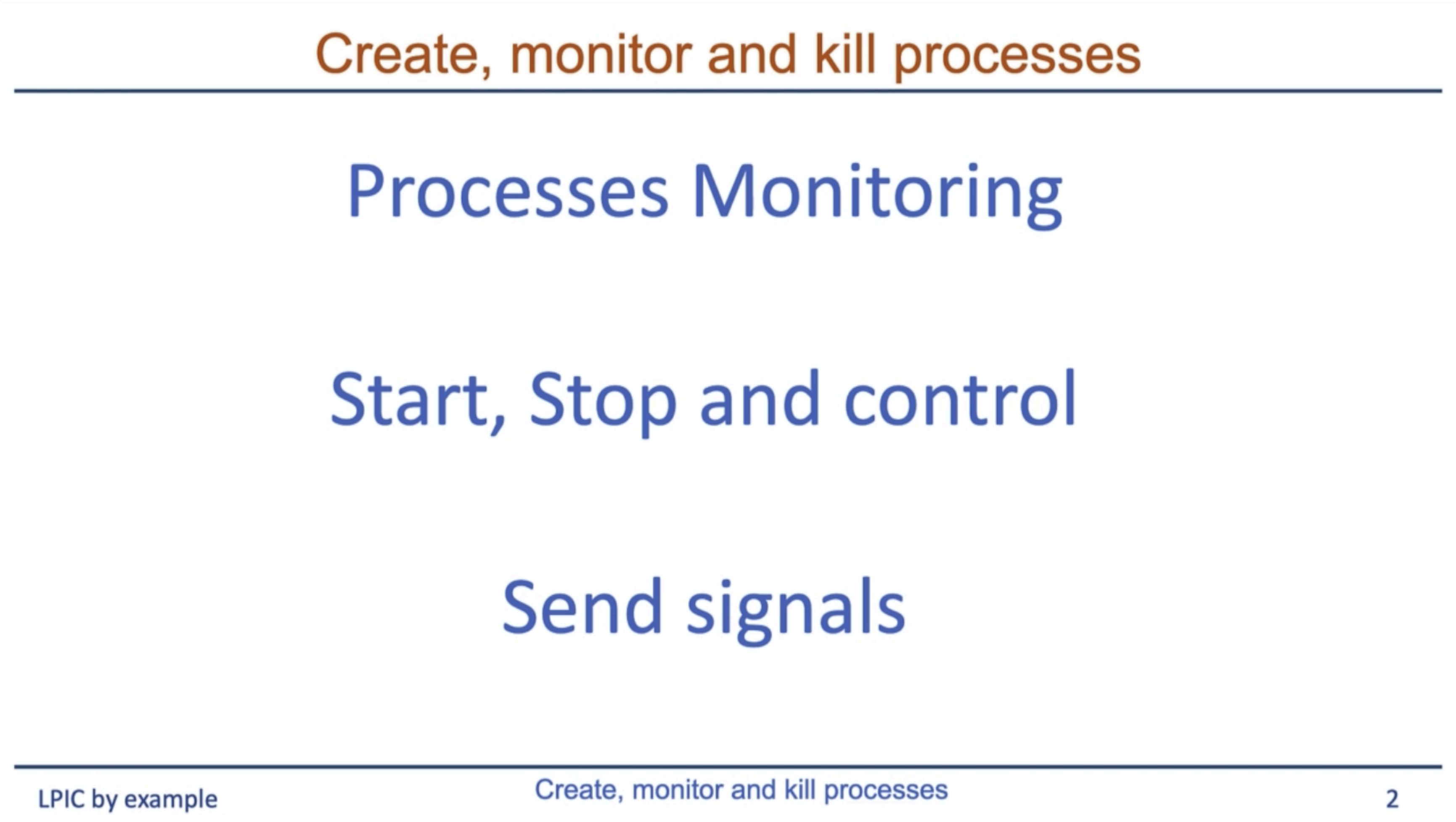
```
$ ln target linkfile

$ ln –s target symlinkfile
```



Create, monitor and kill processes

PART 1

LPIC by example

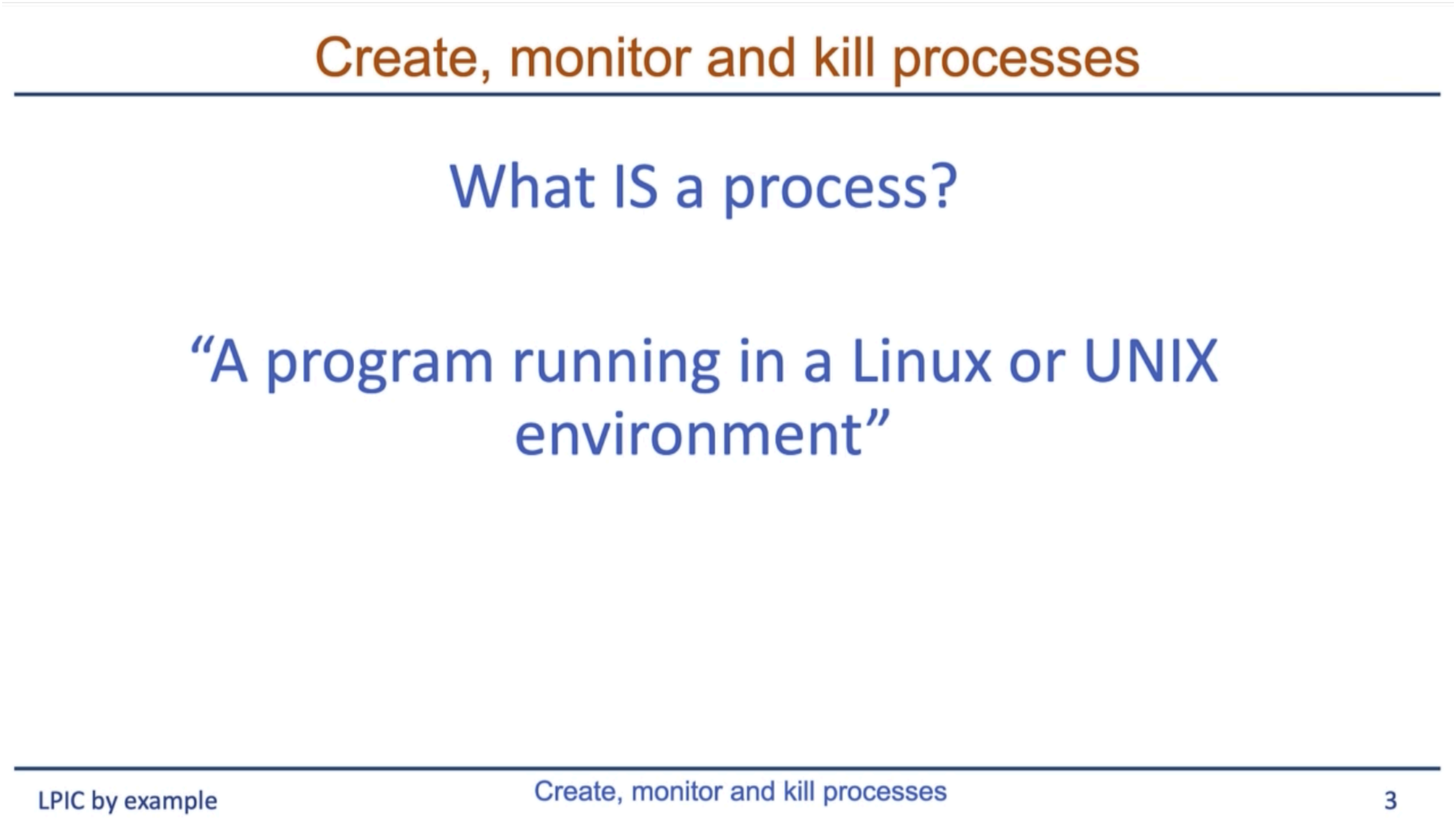


Create, monitor and kill processes

Processes Monitoring

Start, Stop and control

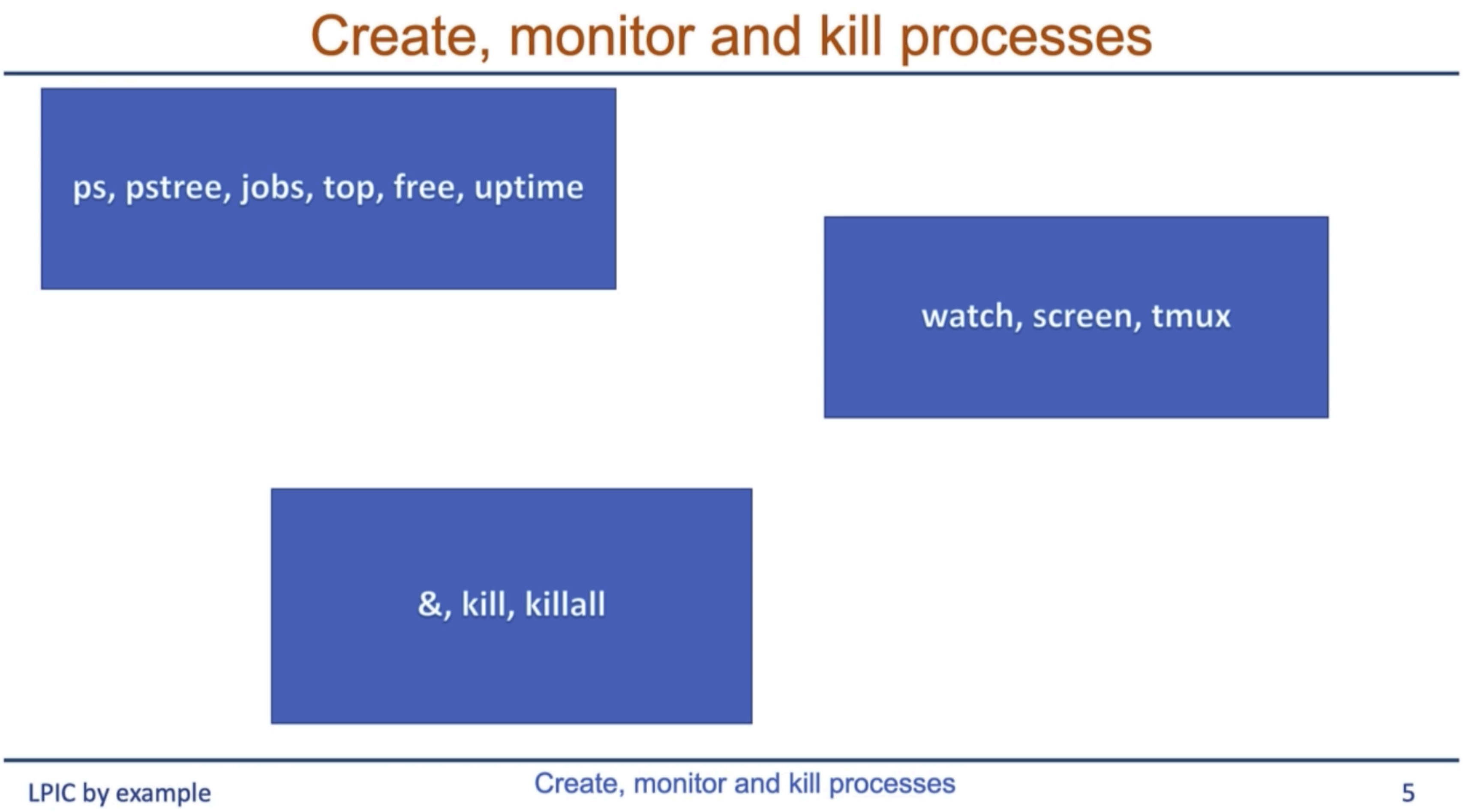
Send signals



Create, monitor and kill processes

What IS a process?

“A program running in a Linux or UNIX environment”

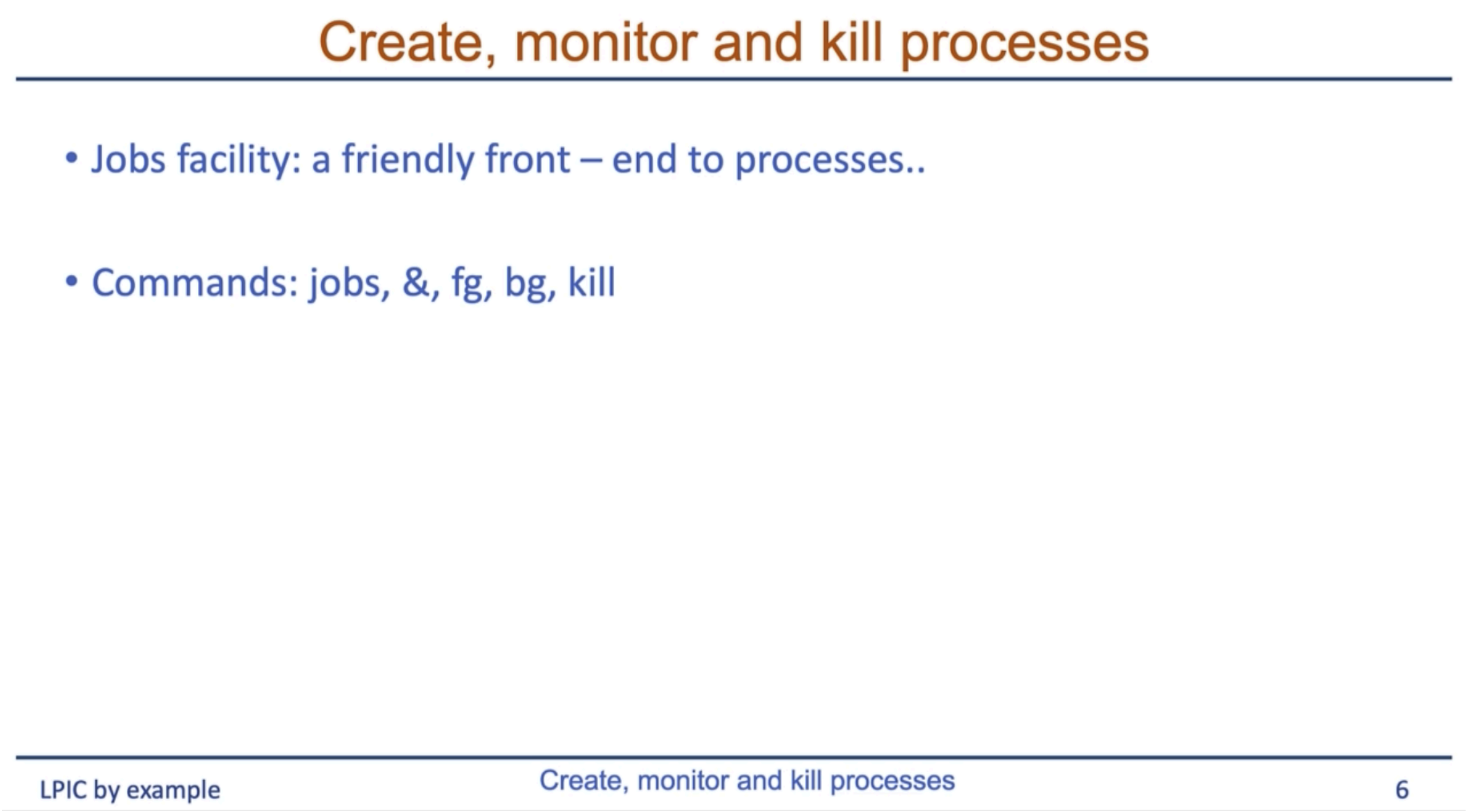


Create, monitor and kill processes

ps, pstree, jobs, top, free, uptime

watch, screen, tmux

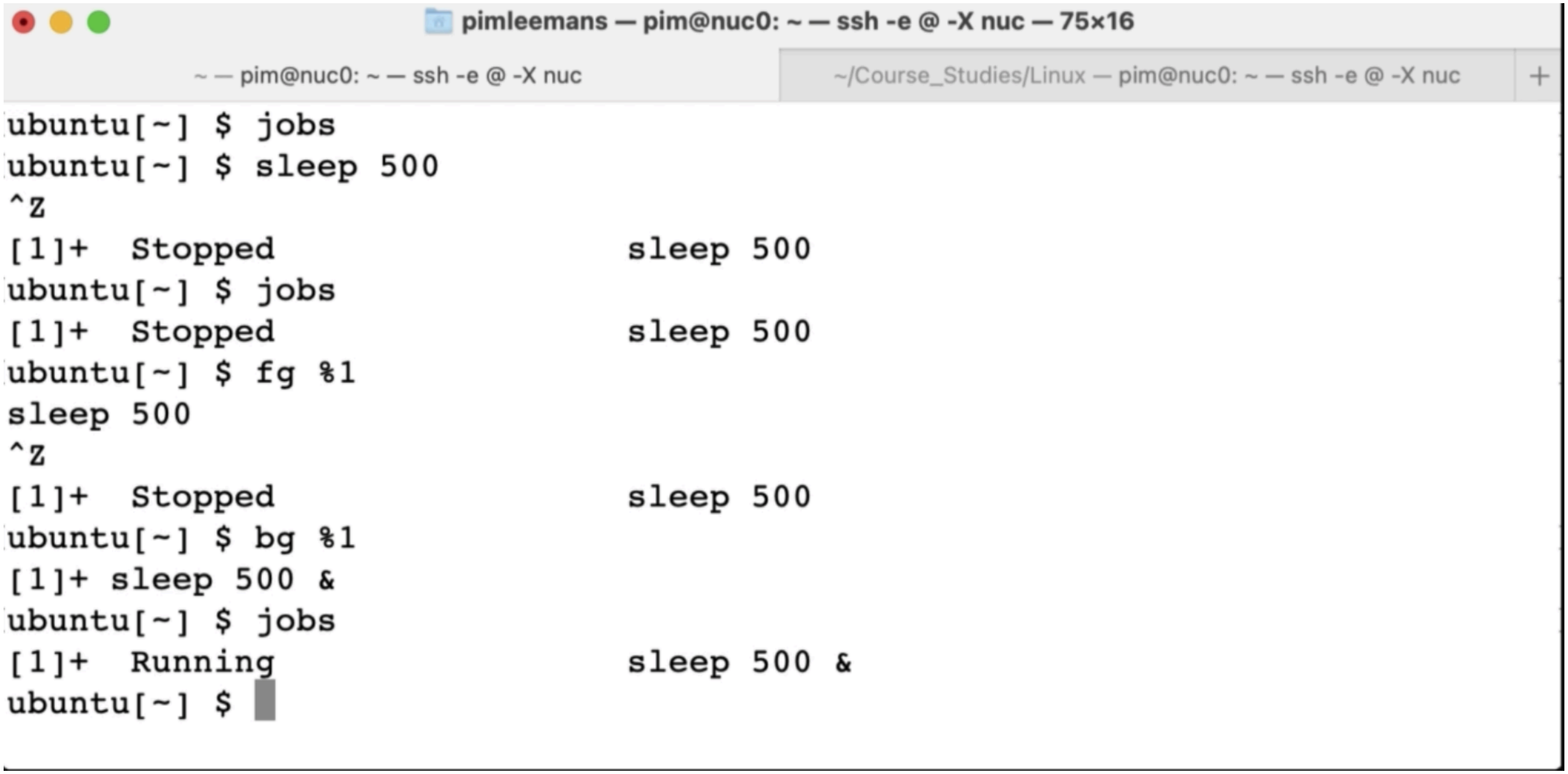
&, kill, killall



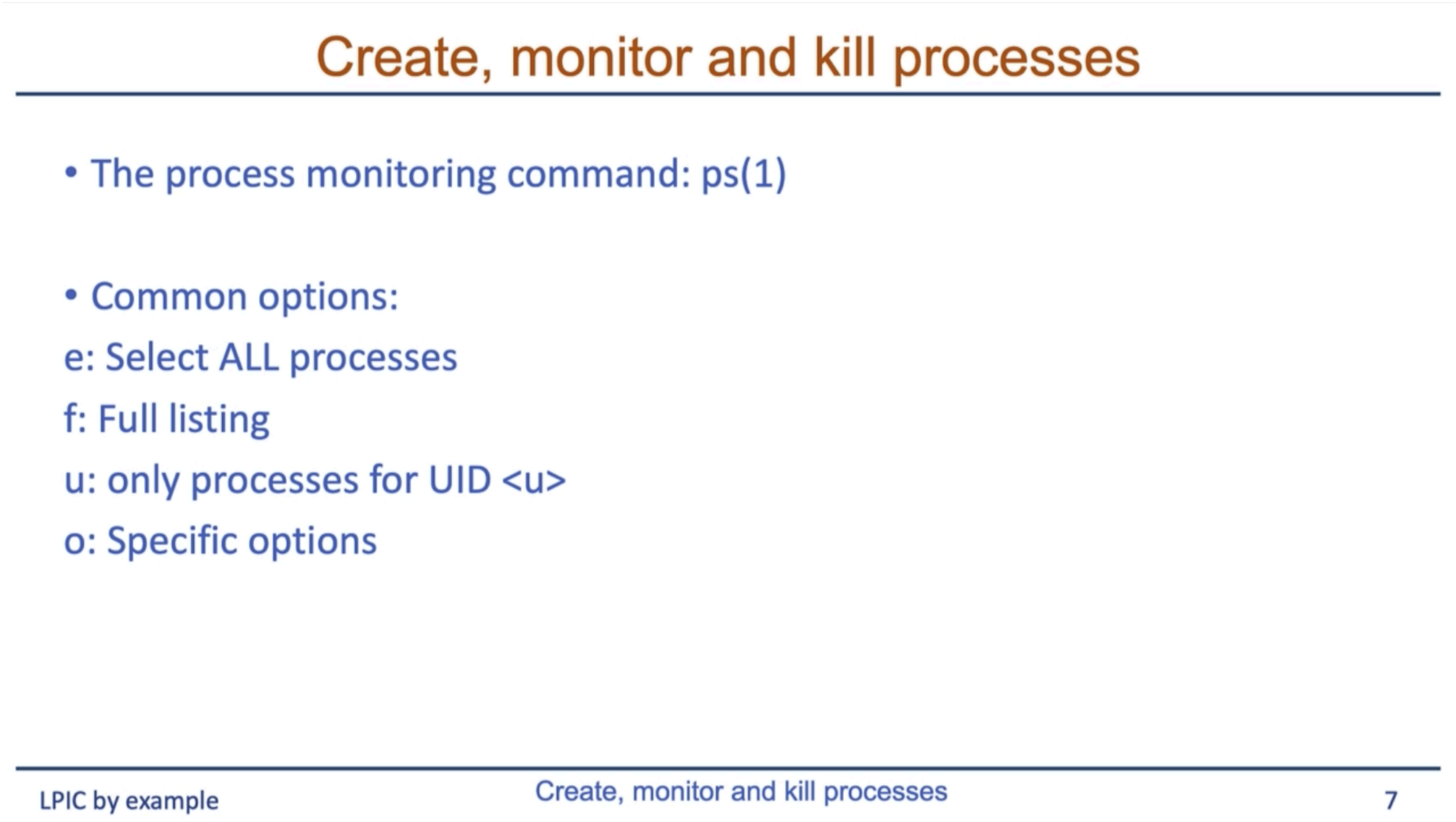
Create, monitor and kill processes

- Jobs facility: a friendly front – end to processes..

- Commands: jobs, &, fg, bg, kill


pimleemans — pim@nuc0: ~ — ssh -e @ -X nuc — 75×16
~ — pim@nuc0: ~ — ssh -e @ -X nuc
~/Course_Studies/Linux — pim@nuc0: ~ — ssh -e @ -X nuc

```
ubuntu[~] $ jobs
ubuntu[~] $ sleep 500
^Z
[1]+  Stopped                 sleep 500
ubuntu[~] $ jobs
[1]+  Stopped                 sleep 500
ubuntu[~] $ fg %1
sleep 500
^Z
[1]+  Stopped                 sleep 500
ubuntu[~] $ bg %1
[1]+ sleep 500 &
ubuntu[~] $ jobs
[1]+  Running                 sleep 500 &
ubuntu[~] $ 
```



Create, monitor and kill processes

- The process monitoring command: ps(1)

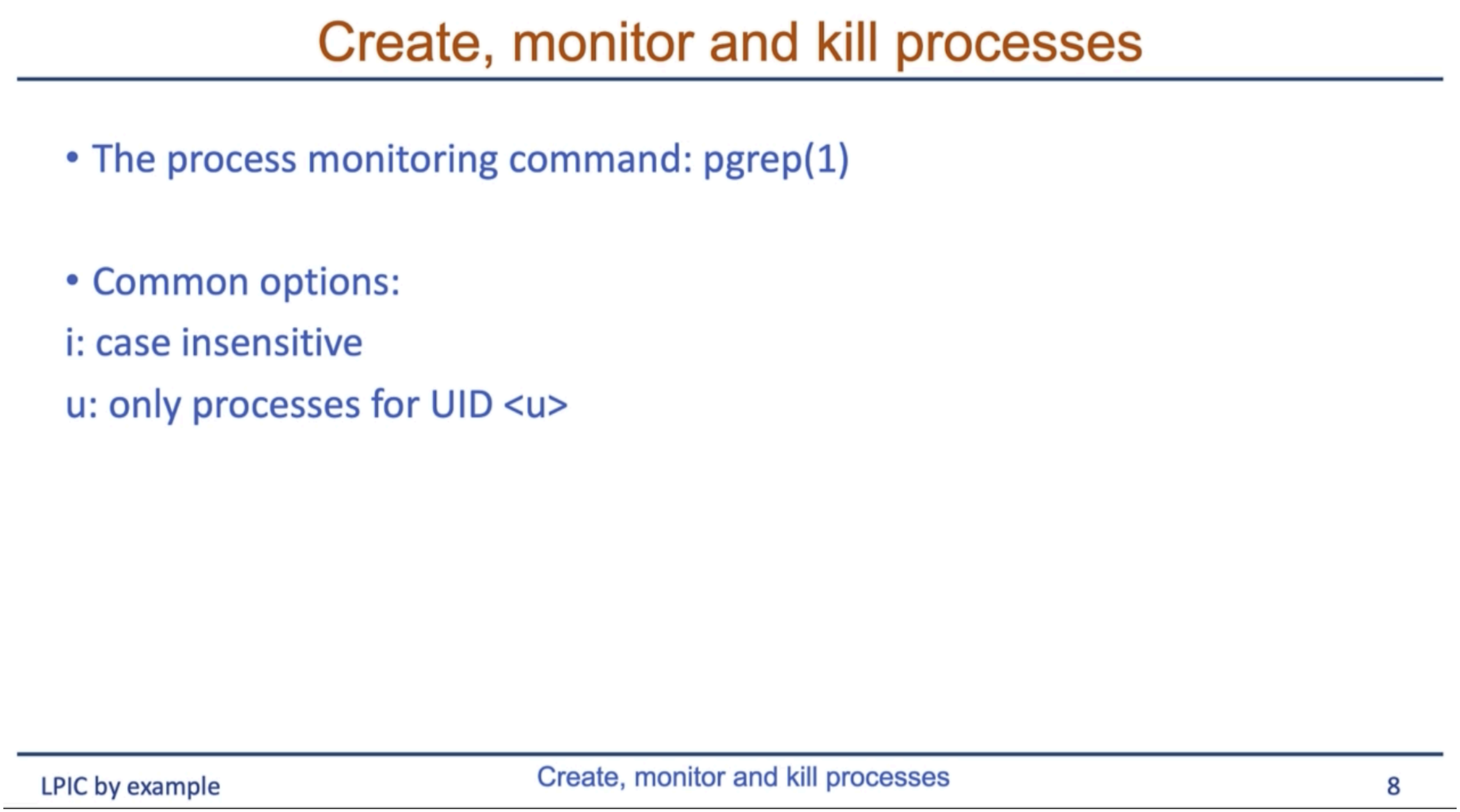
- Common options:

e: Select ALL processes

f: Full listing

u: only processes for UID <u>

o: Specific options



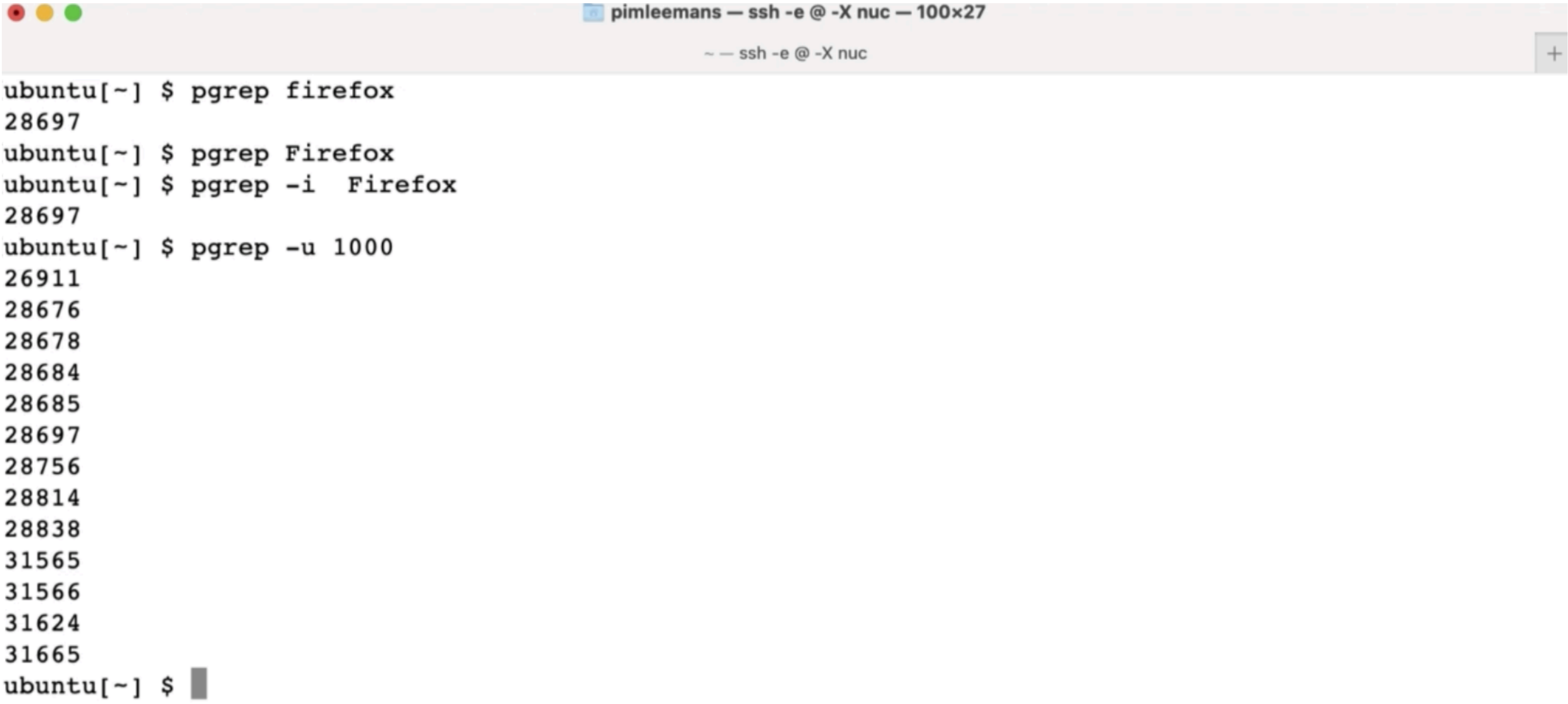
Create, monitor and kill processes

- The process monitoring command: pgrep(1)

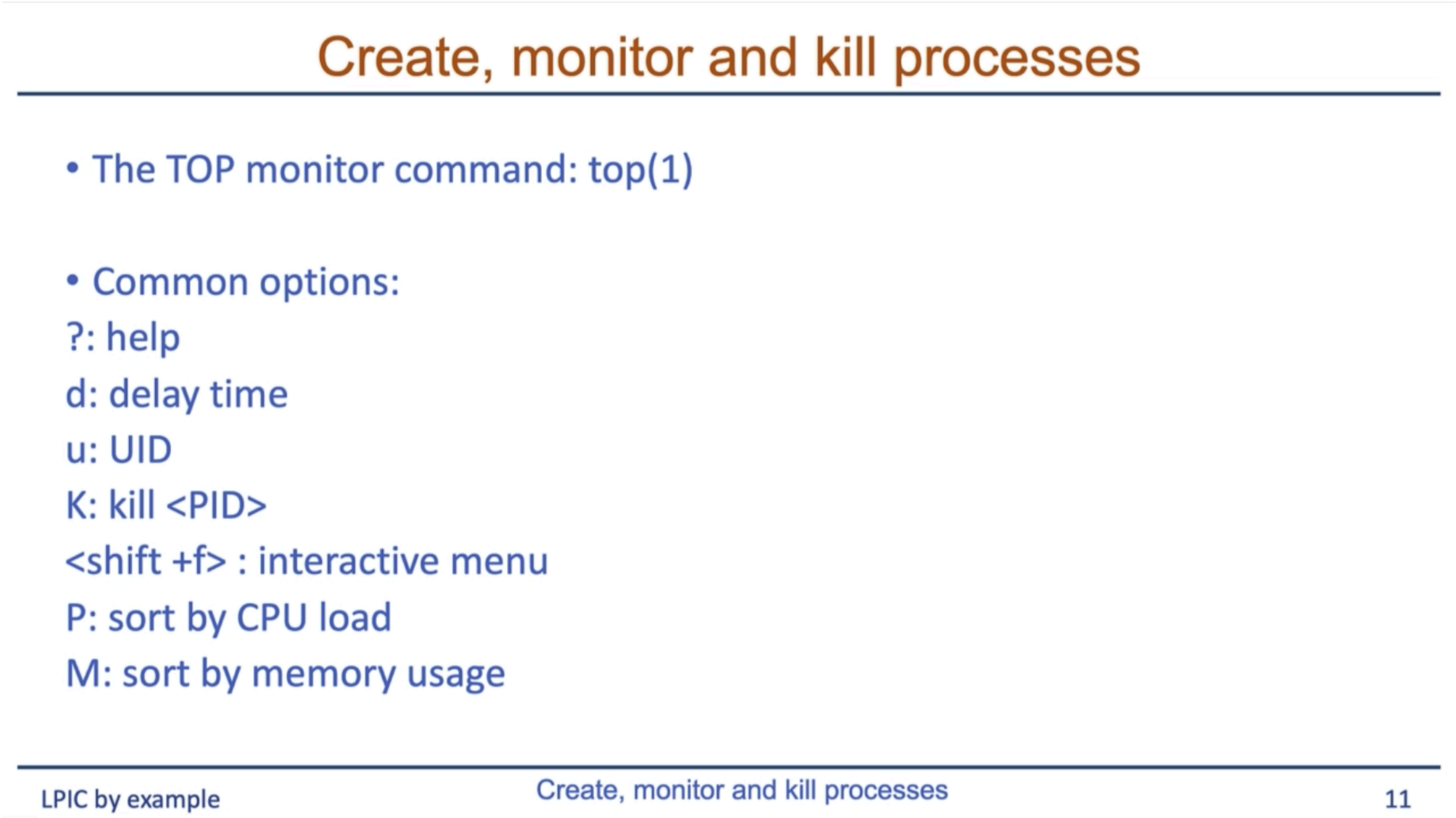
- Common options:

i: case insensitive

u: only processes for UID <u>



```
ubuntu[~] $ pgrep firefox
28697
ubuntu[~] $ pgrep Firefox
ubuntu[~] $ pgrep -i  Firefox
28697
ubuntu[~] $ pgrep -u 1000
26911
28676
28678
28684
28685
28697
28756
28814
28838
31565
31566
31624
31665
ubuntu[~] $ 
```



Create, monitor and kill processes

- The TOP monitor command: top(1)

- Common options:

?: help

d: delay time

u: UID

K: kill <PID>

<shift +f> : interactive menu

P: sort by CPU load

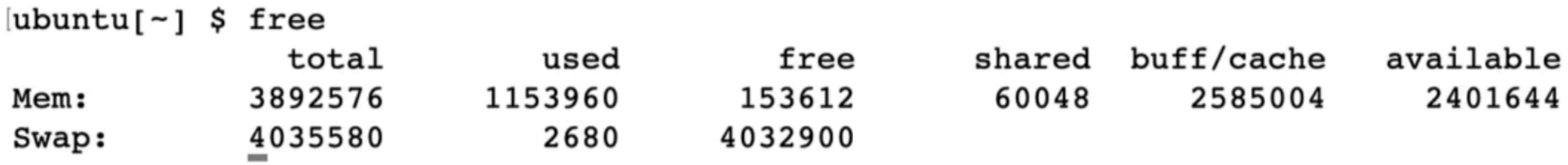
M: sort by memory usage

Create, monitor and kill processes

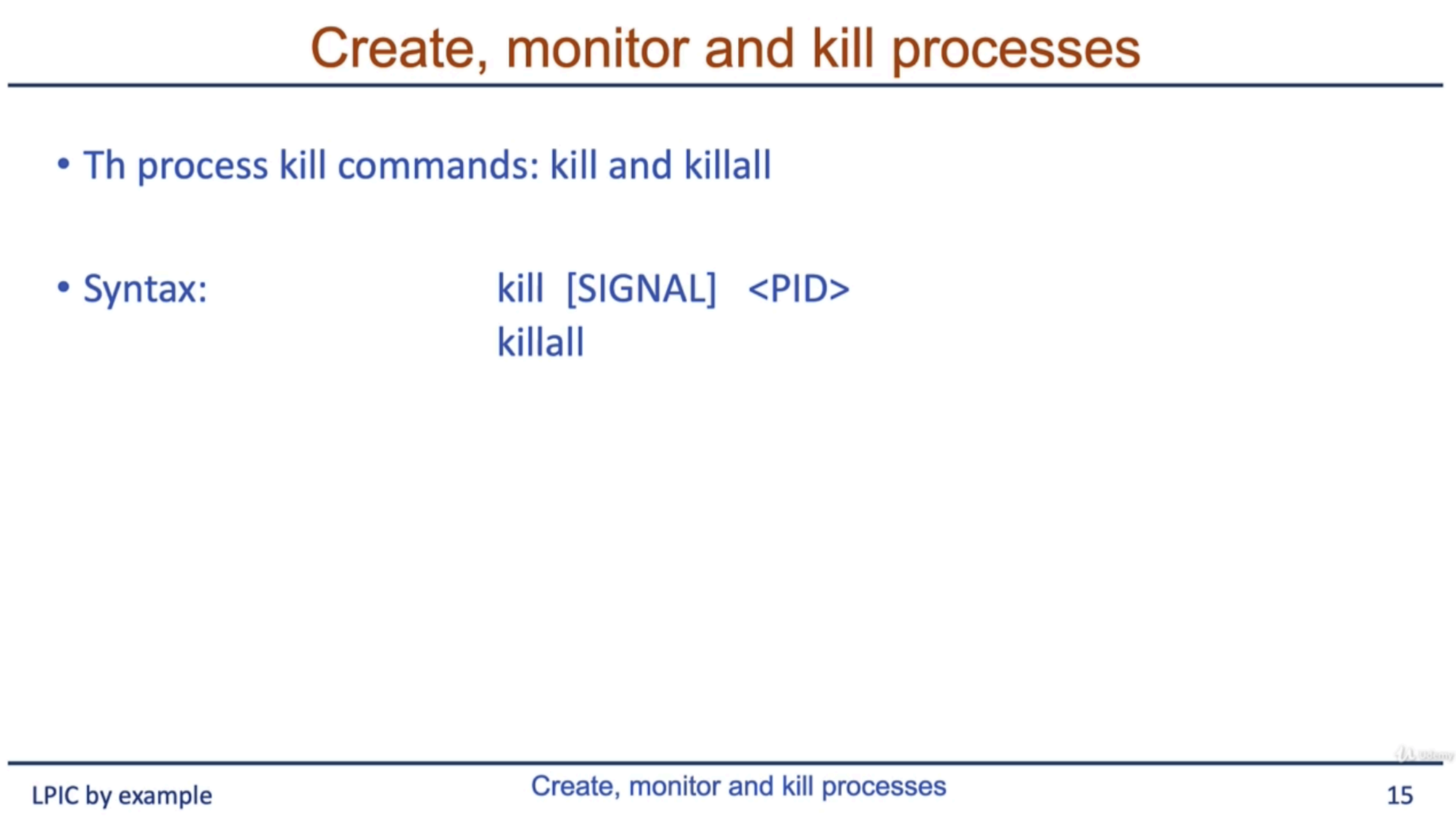
- Other process monitoring commands: free

- Common options:

m/g/h summarize in Mb, Gb, human readable



```
[ubuntu[~] $ free
              total        used        free      shared  buff/cache   available
Mem:        3892576     1153960      153612       60048     2585004     2401644
Swap:       4035580        2680     4032900
```

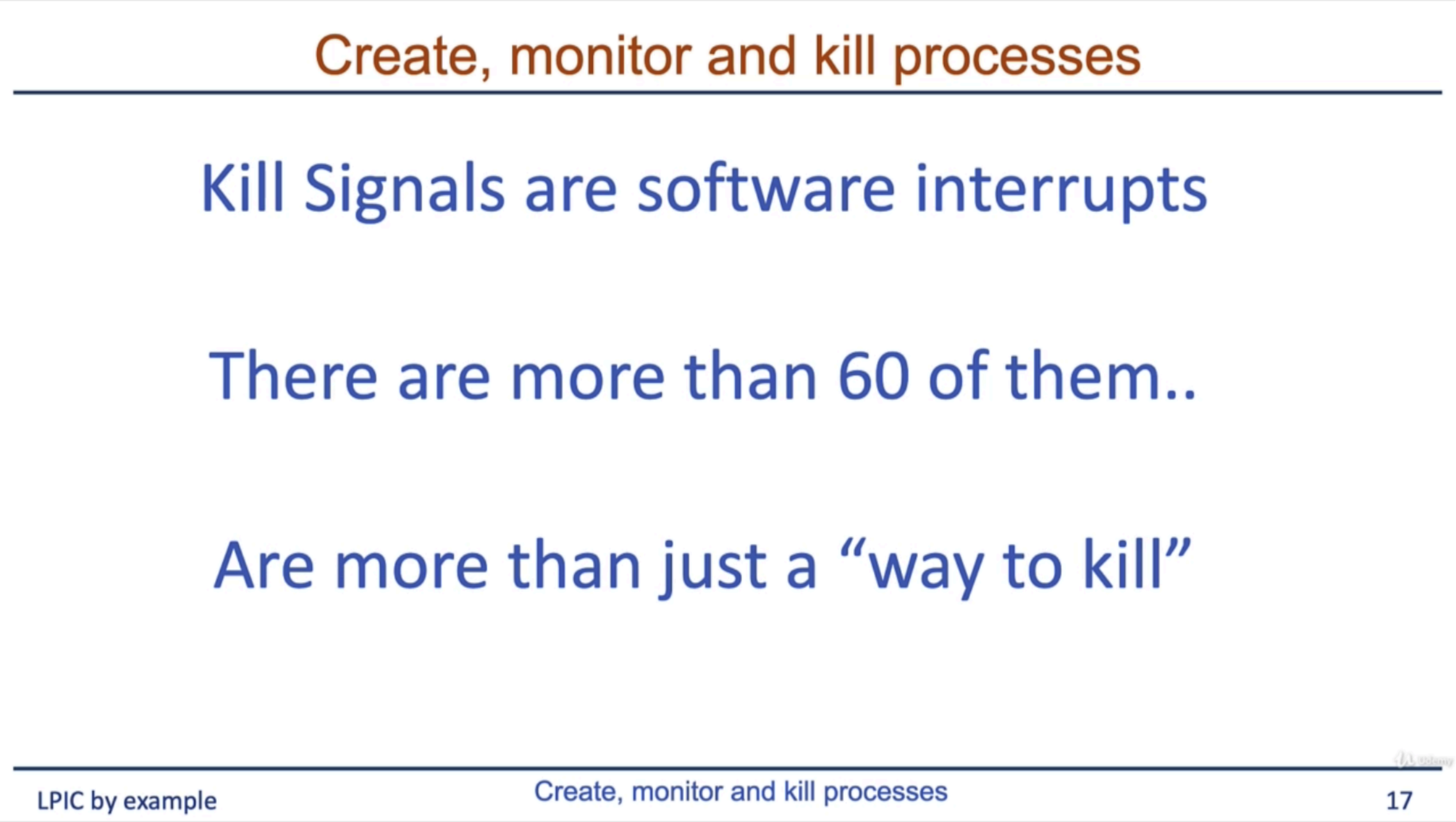


Create, monitor and kill processes

- Th process kill commands: kill and killall

- Syntax:

```
kill  [SIGNAL]   <PID>
killall
```

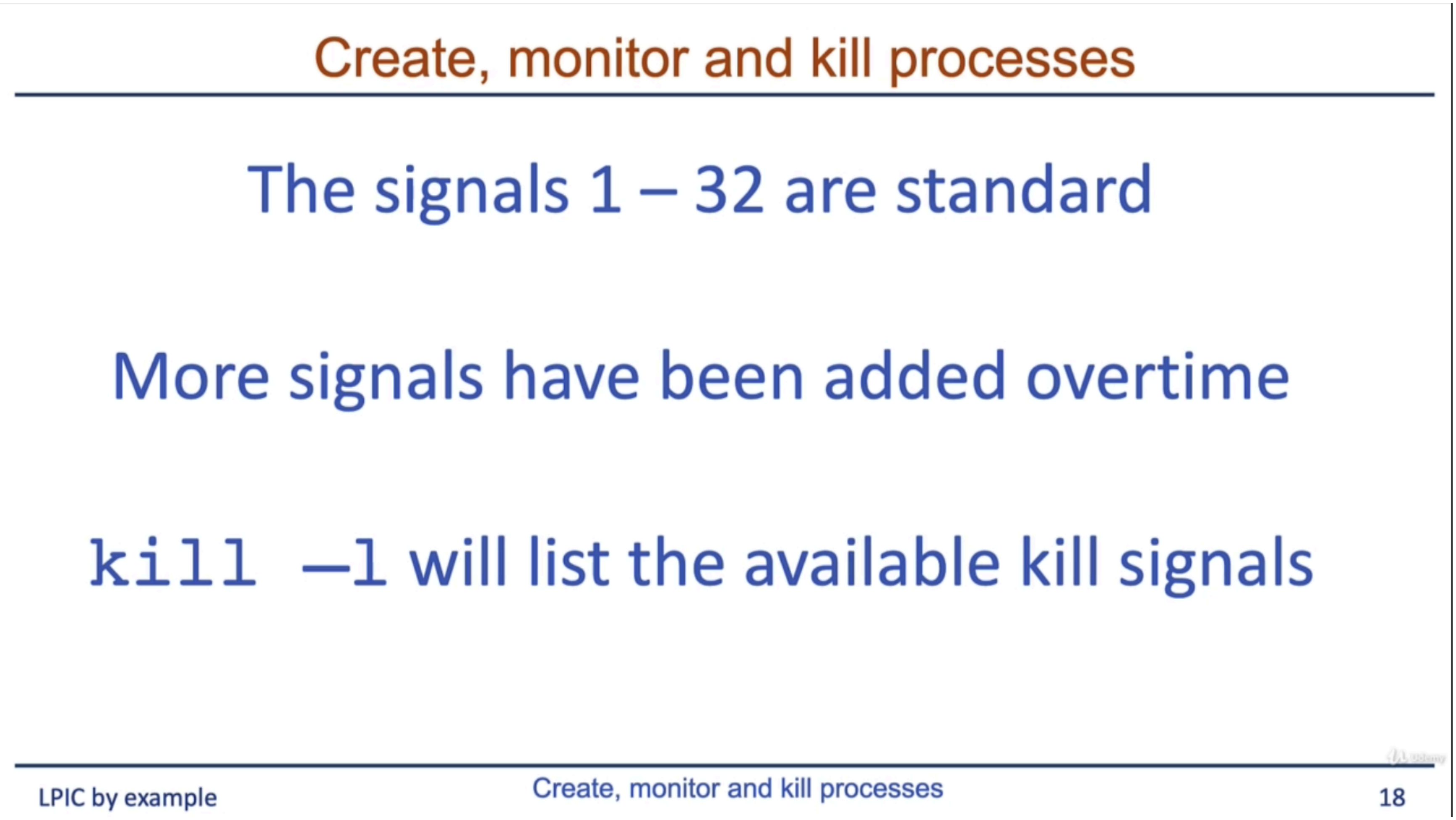


Create, monitor and kill processes

Kill Signals are software interrupts

There are more than 60 of them..

Are more than just a “way to kill”

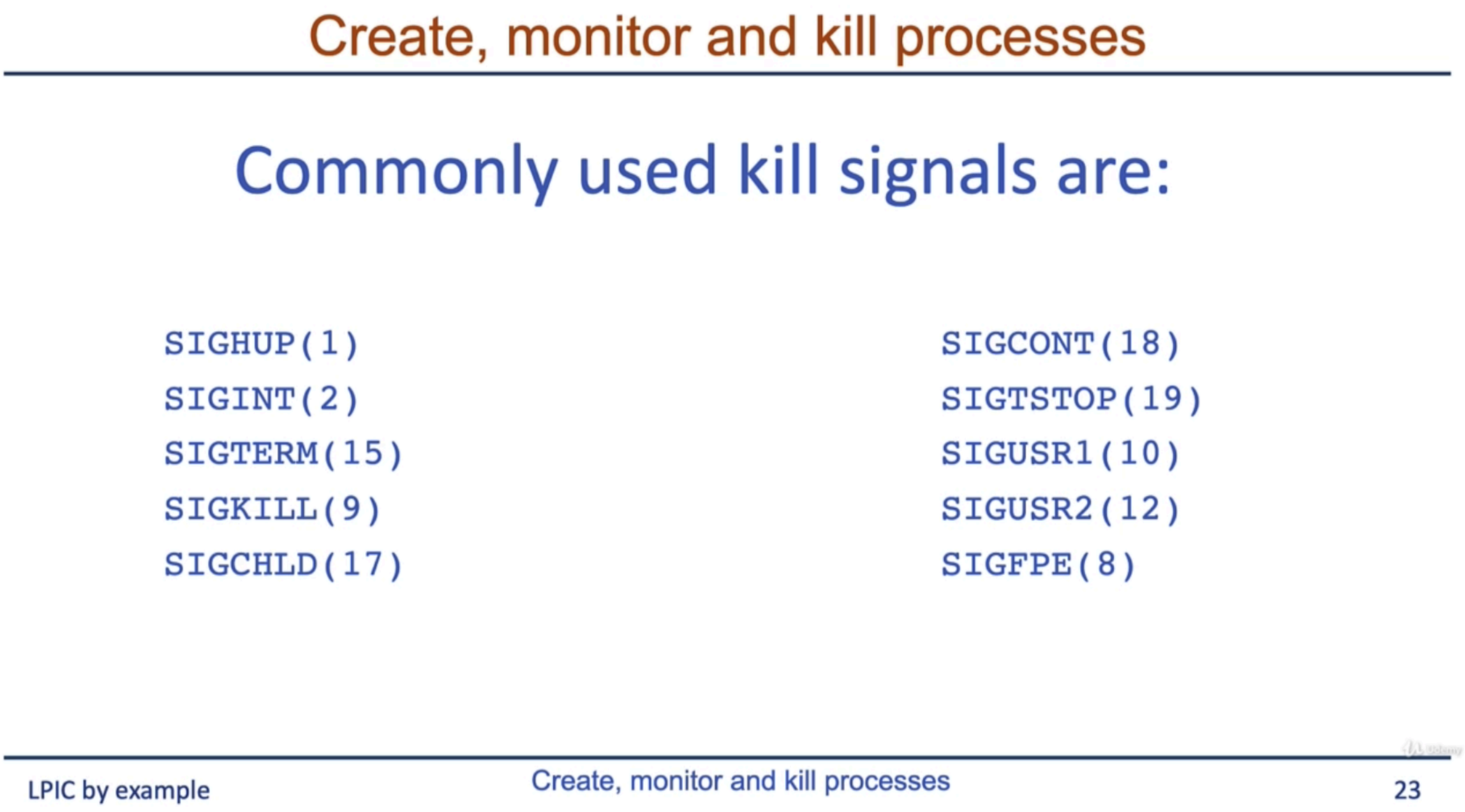


Create, monitor and kill processes

The signals 1 – 32 are standard

More signals have been added overtime

`kill —l` will list the available kill signals

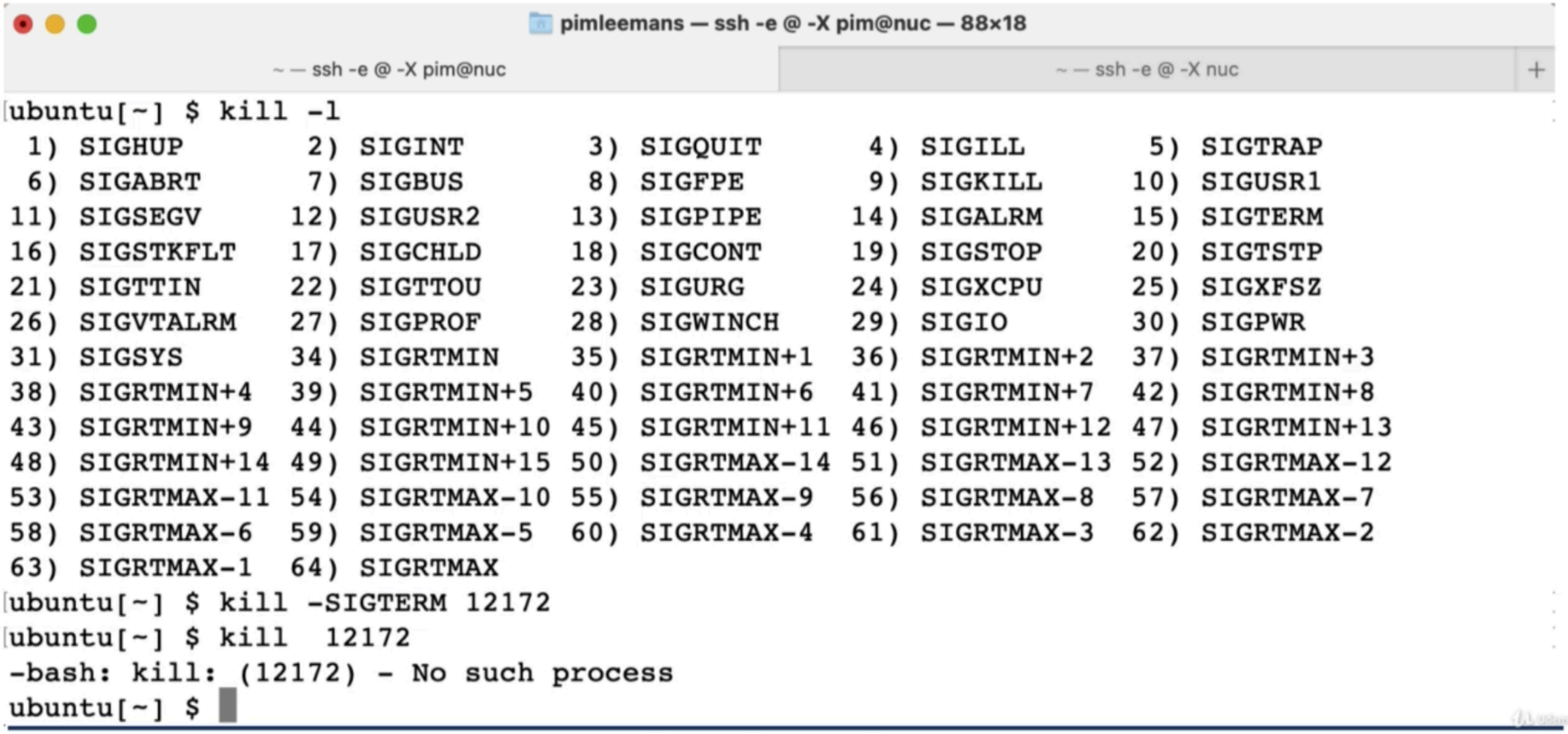


Create, monitor and kill processes

Commonly used kill signals are:

```
SIGHUP(1)
SIGINT(2)
SIGTERM(15)
SIGKILL(9)
SIGCHLD(17)
SIGCONT(18)
SIGTSTOP(19)
SIGUSR1(10)
SIGUSR2(12)
SIGFPE(8)
```

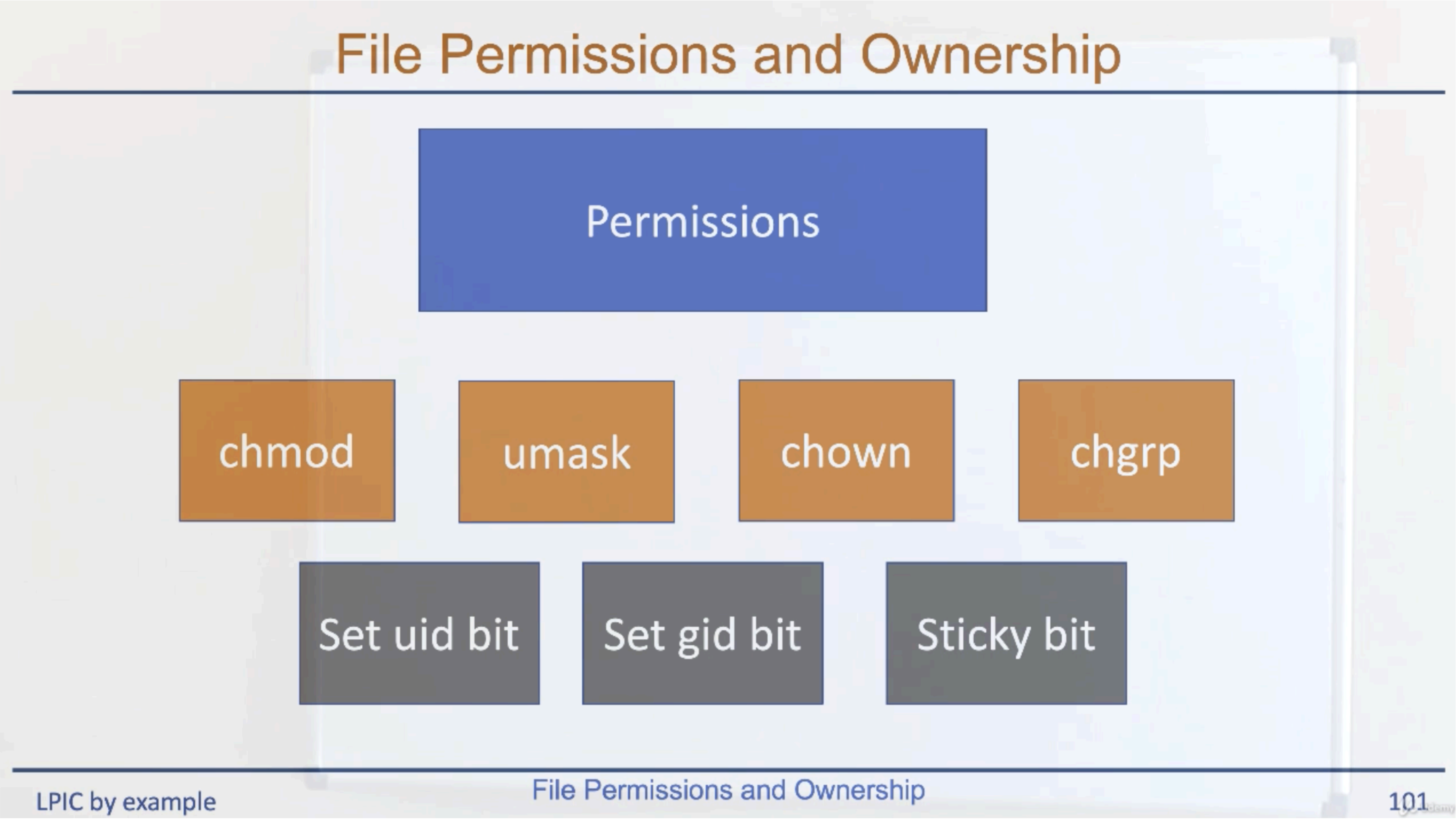
Create, monitor and kill processes



```
[ubuntu[~] $ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
[ubuntu[~] $ kill -SIGTERM 12172
[ubuntu[~] $ kill  12172
-bash: kill: (12172) - No such process
ubuntu[~] $ 
```

File Permissions and Ownership

LPIC by example



File Permissions and Ownership

Permissions

chmod

umask

chown

chgrp

Set uid bit

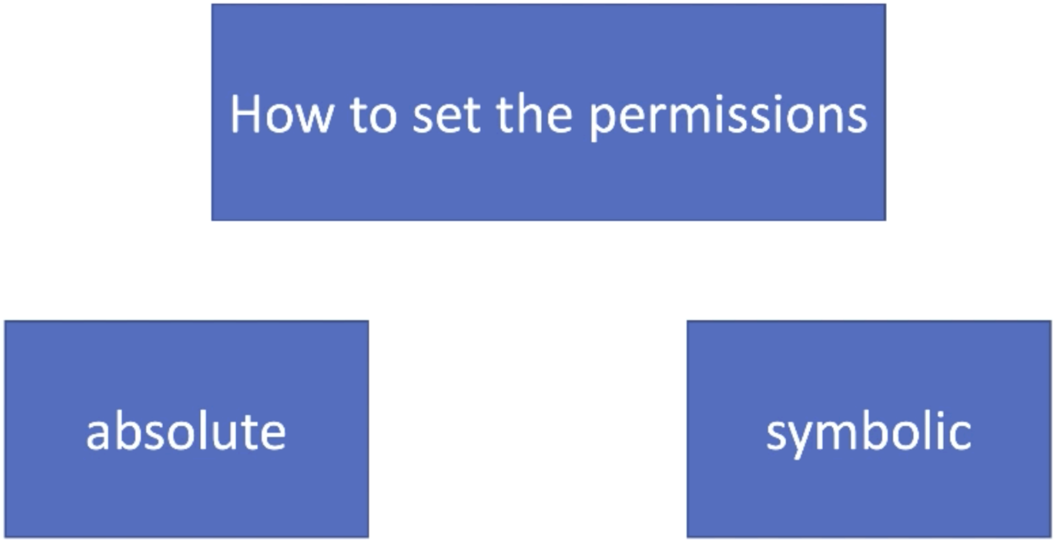
Set gid bit

Sticky bit

File Permissions and Ownership

FILES		DIRECTORIES	
Read	content	Read	list files
Write	content	Write	create, copy, rename files
Execute	run (program/script)	Execute	cd into the directory current working dir

File Permissions and Ownership

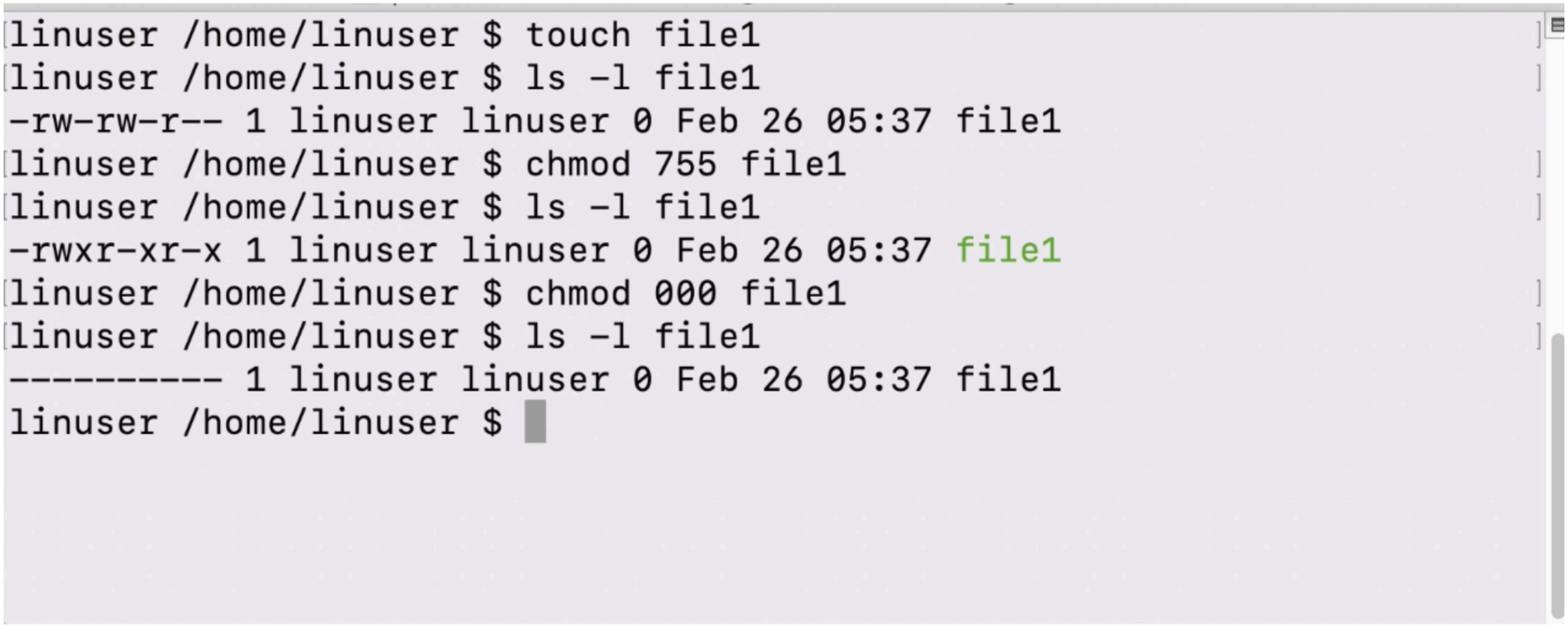


How to set the permissions

absolute

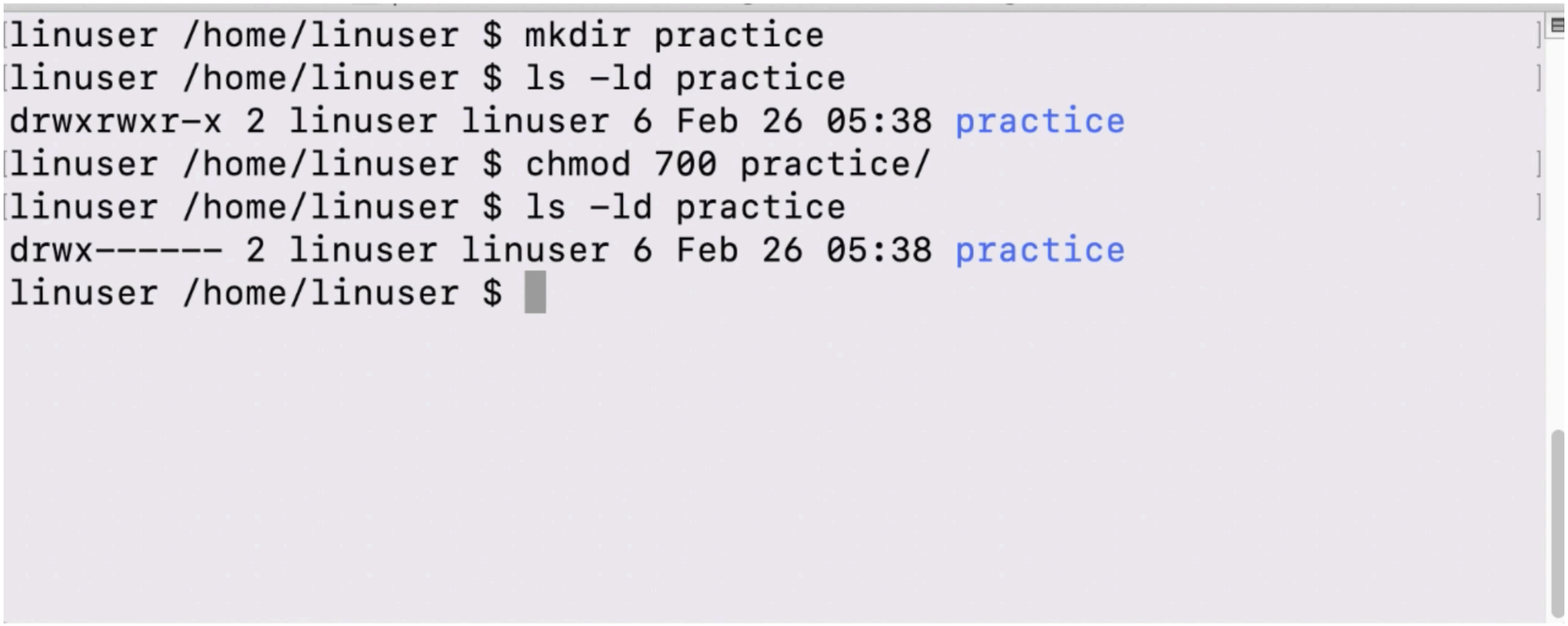
symbolic

File Permissions and Ownership

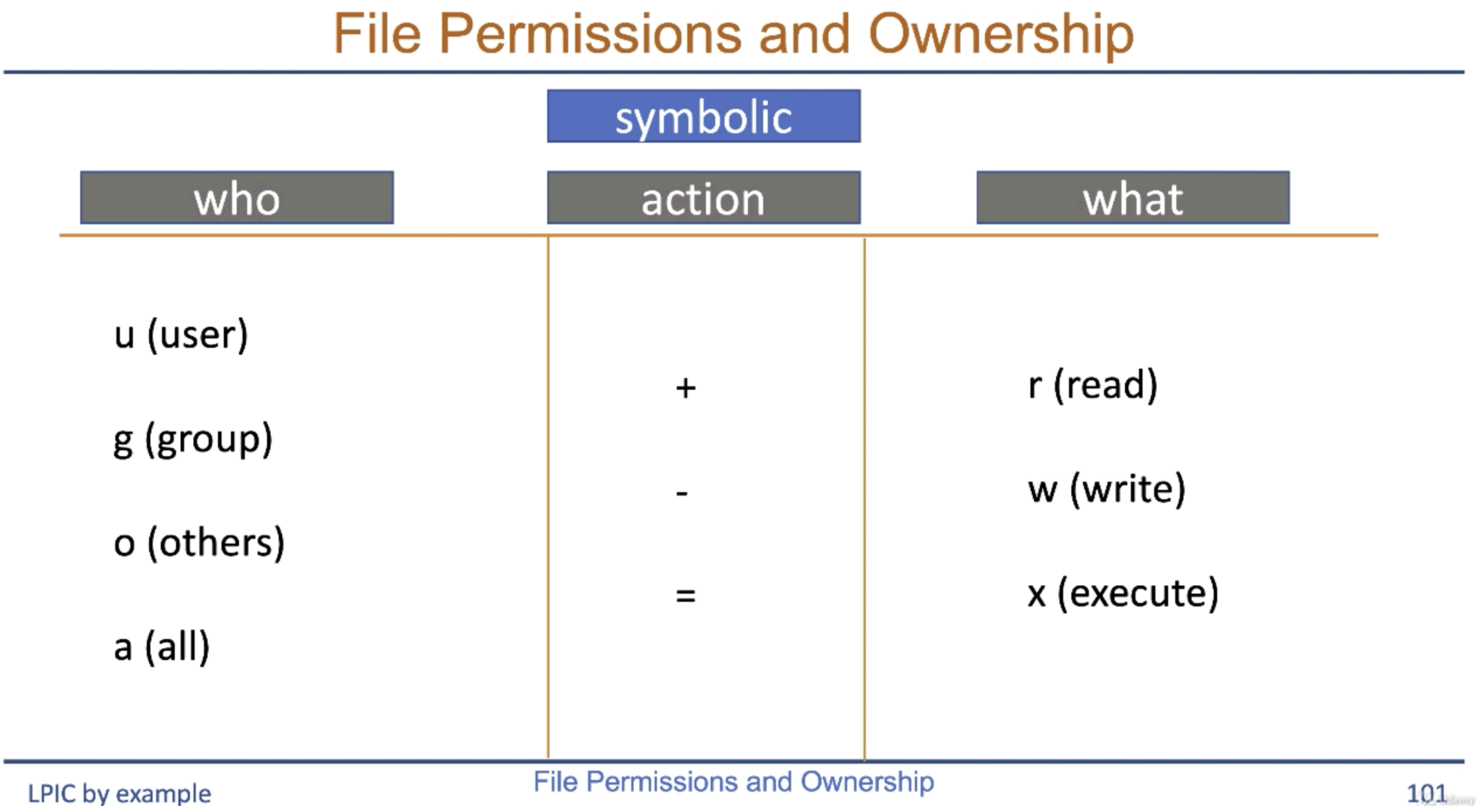


```
linuser /home/linuser $ touch file1
linuser /home/linuser $ ls -l file1
-rw-rw-r-- 1 linuser linuser 0 Feb 26 05:37 file1
linuser /home/linuser $ chmod 755 file1
linuser /home/linuser $ ls -l file1
-rwxr-xr-x 1 linuser linuser 0 Feb 26 05:37 file1
linuser /home/linuser $ chmod 000 file1
linuser /home/linuser $ ls -l file1
---------- 1 linuser linuser 0 Feb 26 05:37 file1
linuser /home/linuser $ 
```

File Permissions and Ownership



```
linuser /home/linuser $ mkdir practice
linuser /home/linuser $ ls -ld practice
drwxrwxr-x 2 linuser linuser 6 Feb 26 05:38 practice
linuser /home/linuser $ chmod 700 practice/
linuser /home/linuser $ ls -ld practice
drwx------ 2 linuser linuser 6 Feb 26 05:38 practice
linuser /home/linuser $ 
```

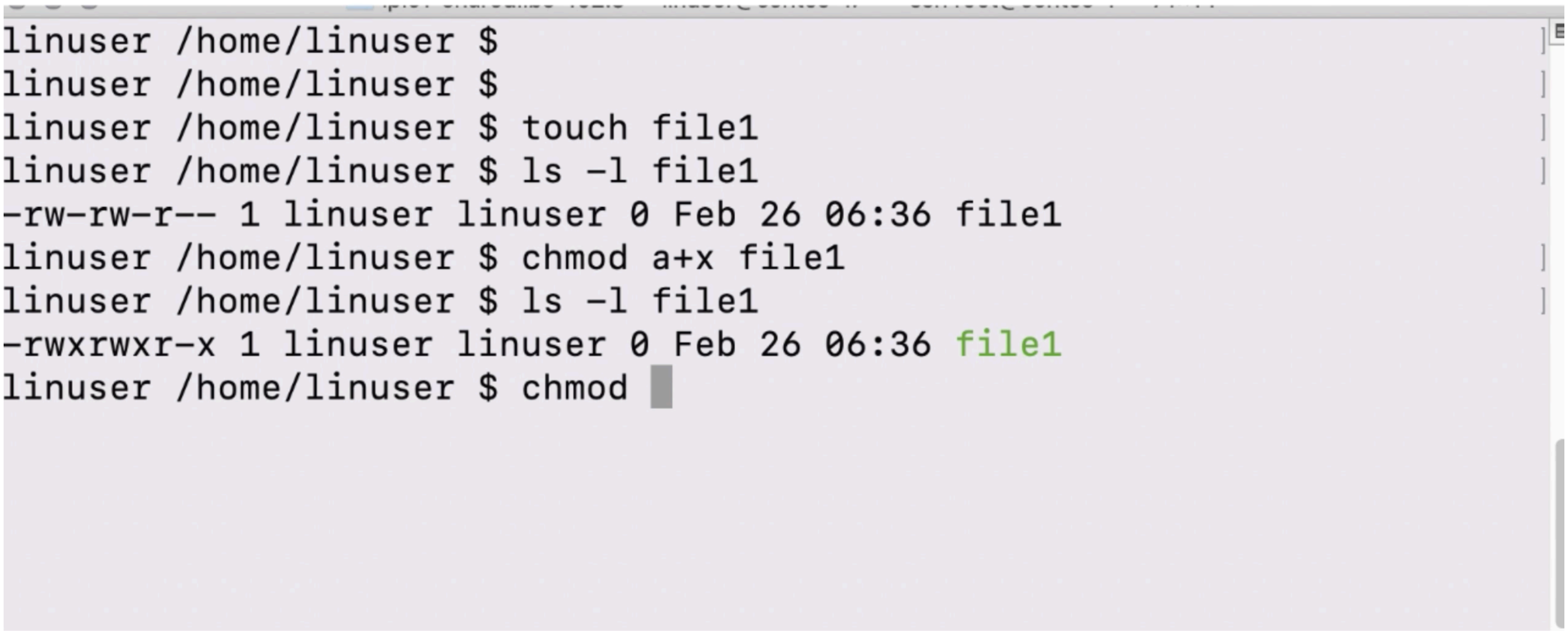


File Permissions and Ownership

symbolic

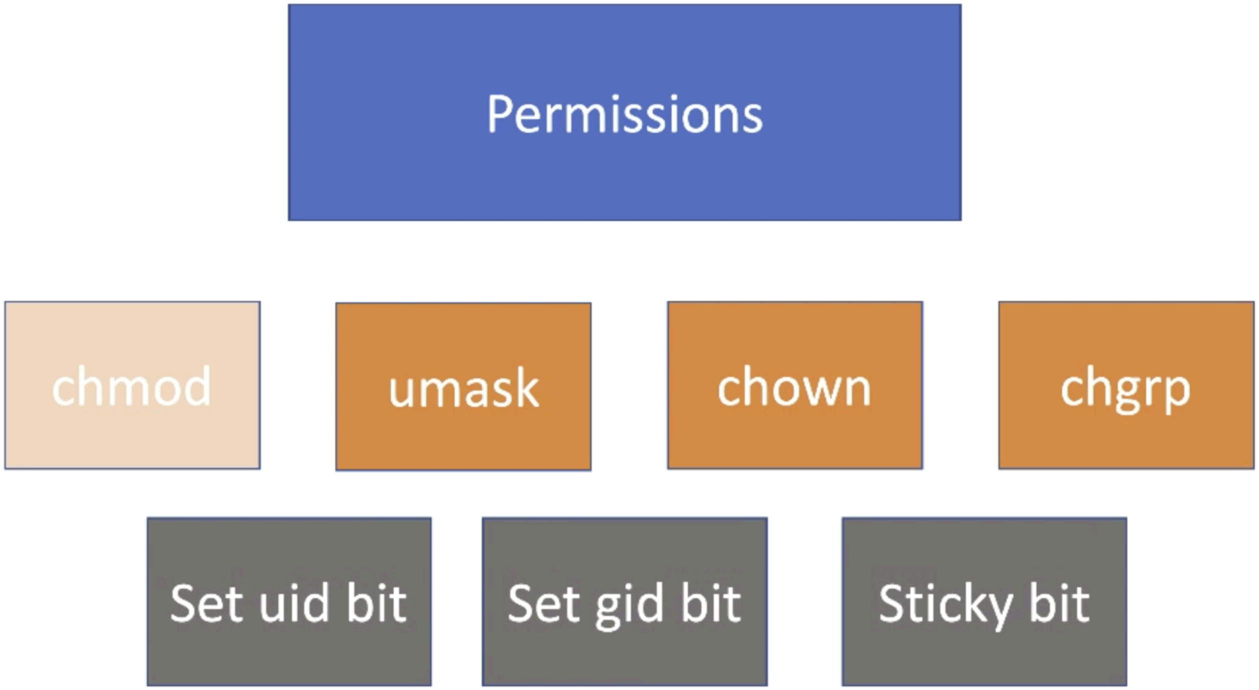
who	action	what
u (user)	+	r (read)
g (group)	-	w (write)
o (others)	=	x (execute)
a (all)		

File Permissions and Ownership



```
linuser /home/linuser $
linuser /home/linuser $
linuser /home/linuser $ touch file1
linuser /home/linuser $ ls -l file1
-rw-rw-r-- 1 linuser linuser 0 Feb 26 06:36 file1
linuser /home/linuser $ chmod a+x file1
linuser /home/linuser $ ls -l file1
-rwxrwxr-x 1 linuser linuser 0 Feb 26 06:36 file1
linuser /home/linuser $ chmod
```

File Permissions and Ownership (part 2)



Permissions

chmod

umask

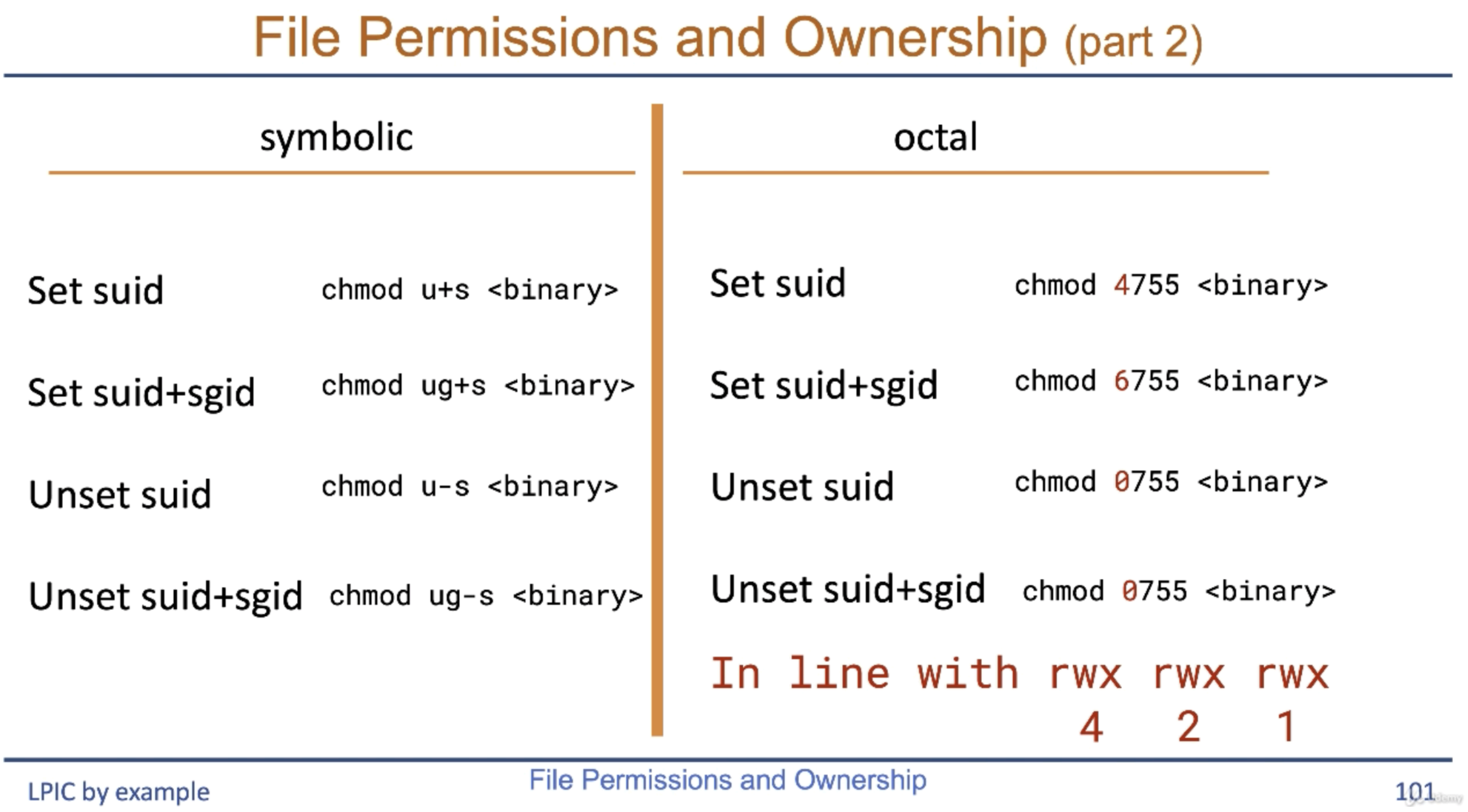
chown

chgrp

Set uid bit

Set gid bit

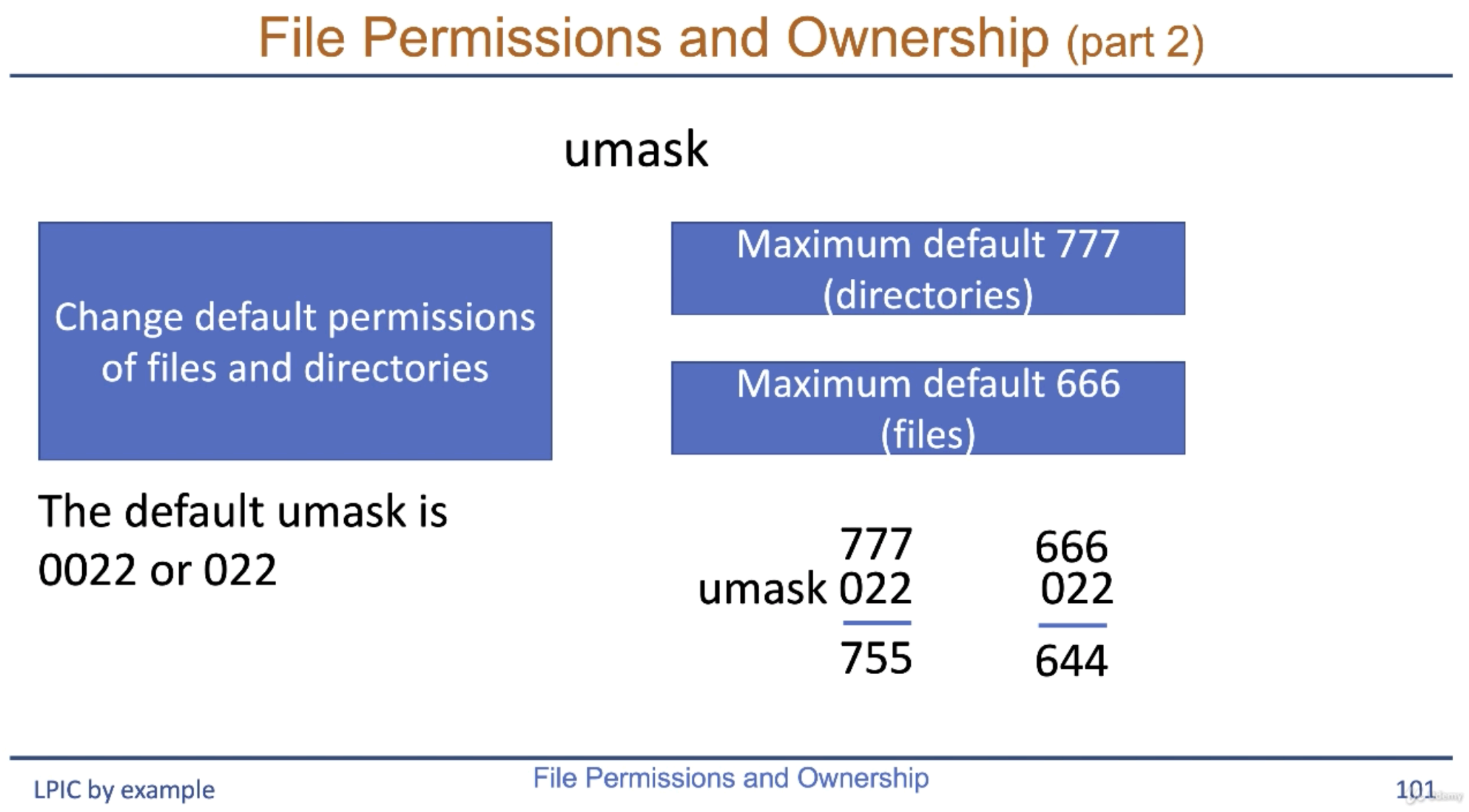
Sticky bit



File Permissions and Ownership (part 2)

symbolic		octal	
Set suid	`chmod u+s <binary>`	Set suid	`chmod 4755 <binary>`
Set suid+sgid	`chmod ug+s <binary>`	Set suid+sgid	`chmod 6755 <binary>`
Unset suid	`chmod u-s <binary>`	Unset suid	`chmod 0755 <binary>`
Unset suid+sgid	`chmod ug-s <binary>`	Unset suid+sgid	`chmod 0755 <binary>`

```
In line with rwx rwx rwx
              4   2   1
```



File Permissions and Ownership (part 2)

umask

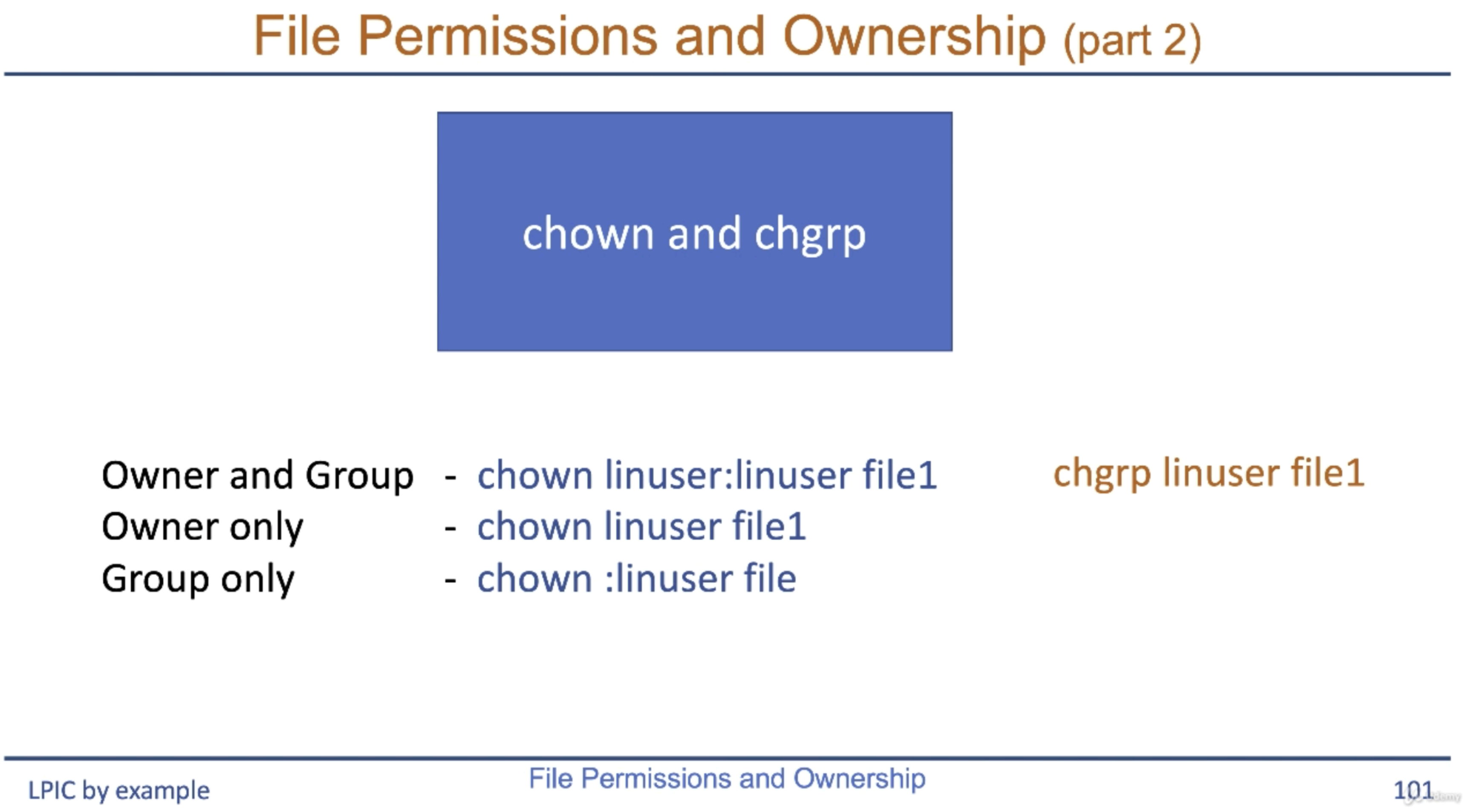
Change default permissions of files and directories

Maximum default 777 (directories)

Maximum default 666 (files)

The default umask is 0022 or 022

```
         777      666
  umask  022      022
         ___      ___
         755      644
```



File Permissions and Ownership (part 2)

chown and chgrp

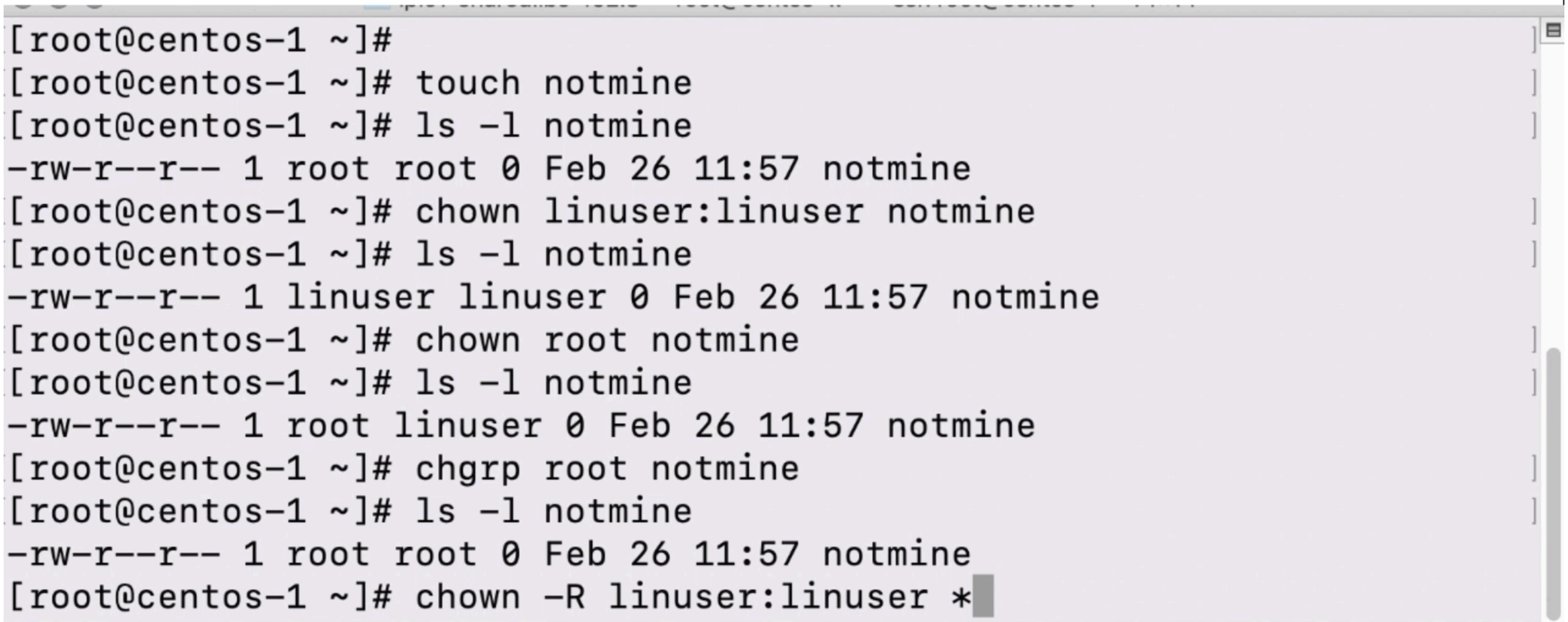
Owner and Group - chown linuser:linuser file1

Owner only - chown linuser file1

Group only - chown :linuser file

chgrp linuser file1

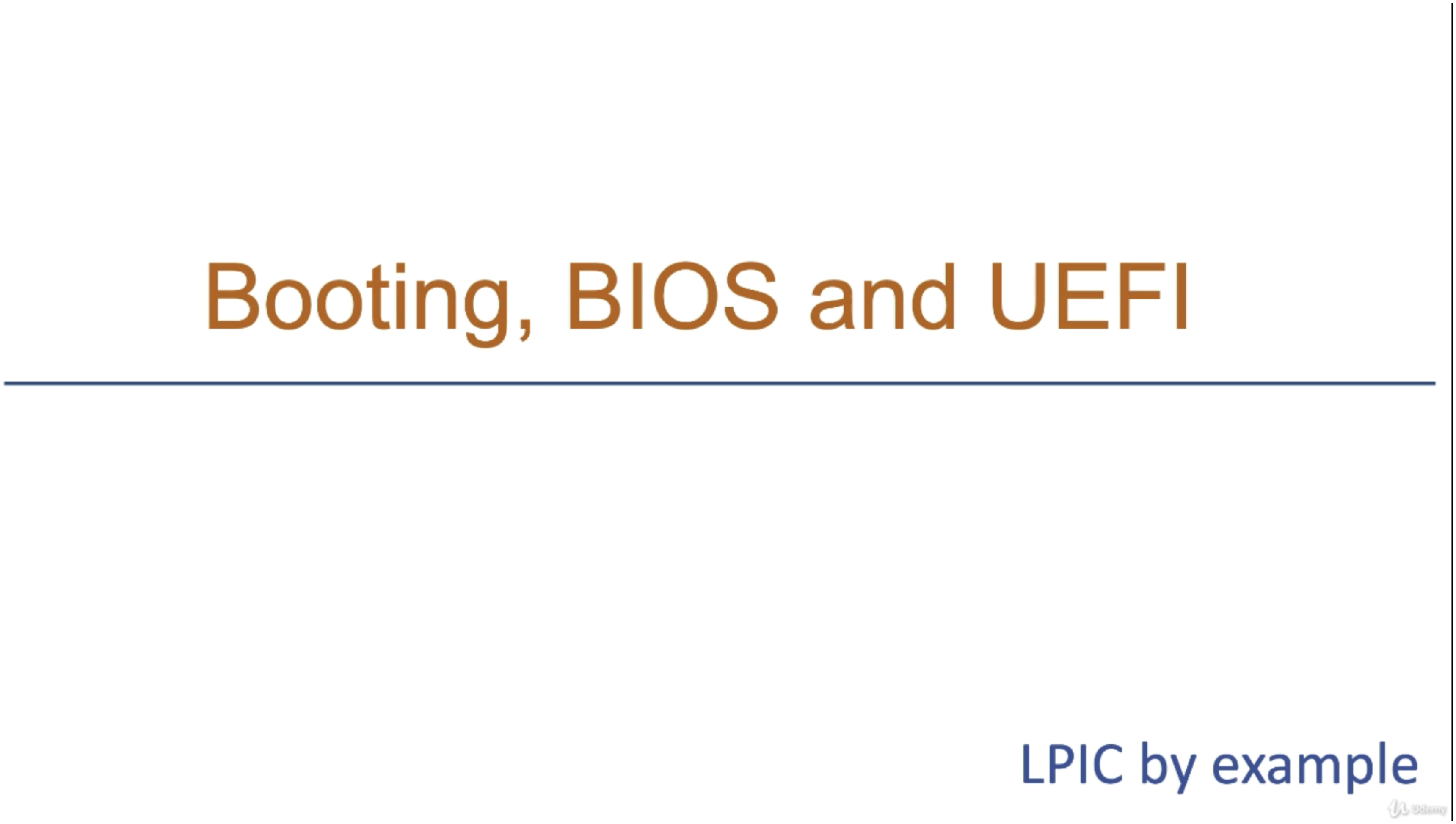
File Permissions and Ownership (part 2)



```
[root@centos-1 ~]#
[root@centos-1 ~]# touch notmine
[root@centos-1 ~]# ls -l notmine
-rw-r--r-- 1 root root 0 Feb 26 11:57 notmine
[root@centos-1 ~]# chown linuser:linuser notmine
[root@centos-1 ~]# ls -l notmine
-rw-r--r-- 1 linuser linuser 0 Feb 26 11:57 notmine
[root@centos-1 ~]# chown root notmine
[root@centos-1 ~]# ls -l notmine
-rw-r--r-- 1 root linuser 0 Feb 26 11:57 notmine
[root@centos-1 ~]# chgrp root notmine
[root@centos-1 ~]# ls -l notmine
-rw-r--r-- 1 root root 0 Feb 26 11:57 notmine
[root@centos-1 ~]# chown -R linuser:linuser *
```

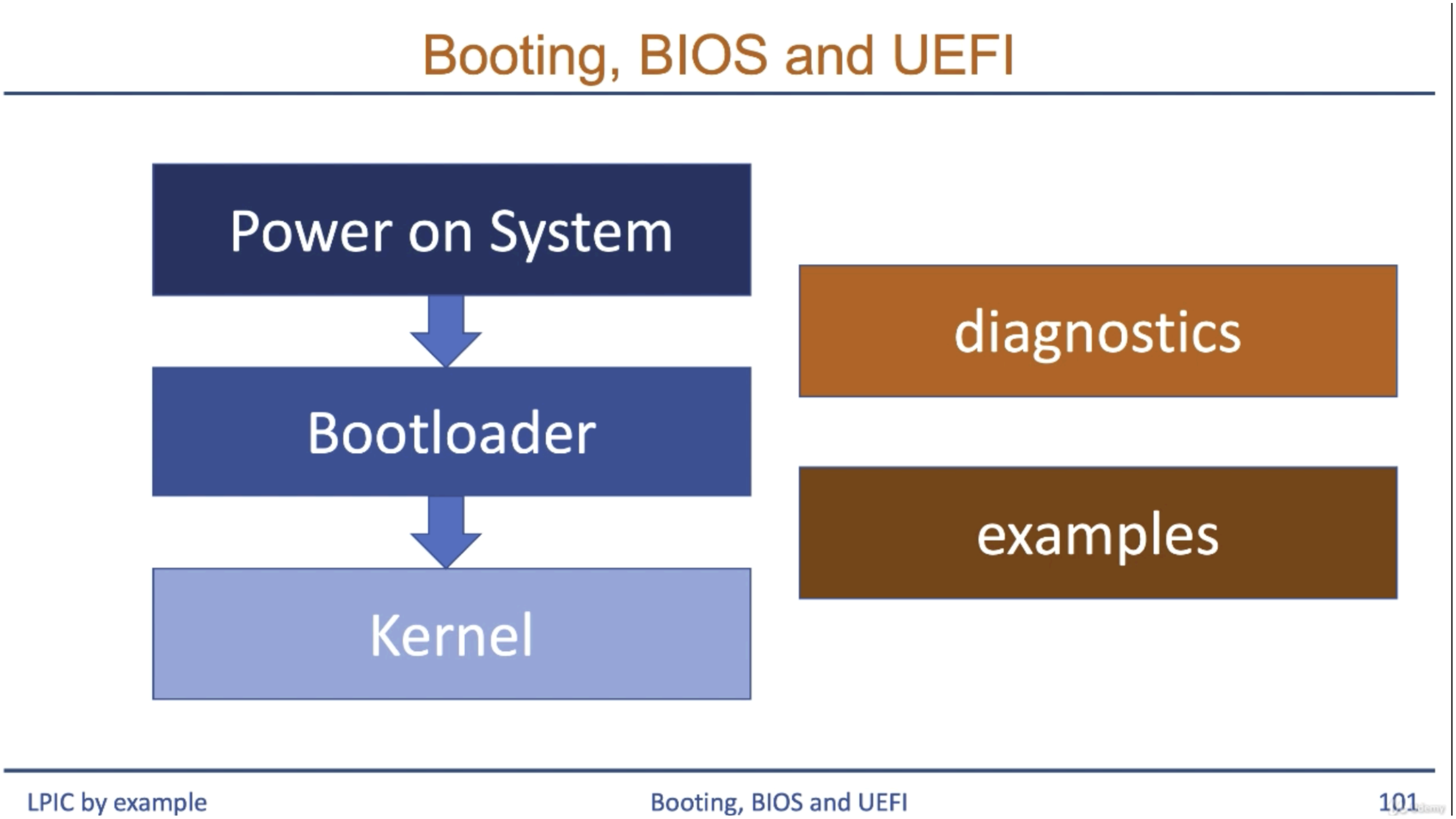
File Permissions and Ownership (part 2)

- Different types of permissions
- Different ways of setting permissions using chmod
- suid sgid sticky bit
- umask command
- Change ownership with chown and chgrp



Booting, BIOS and UEFI

LPIC by example



Booting, BIOS and UEFI

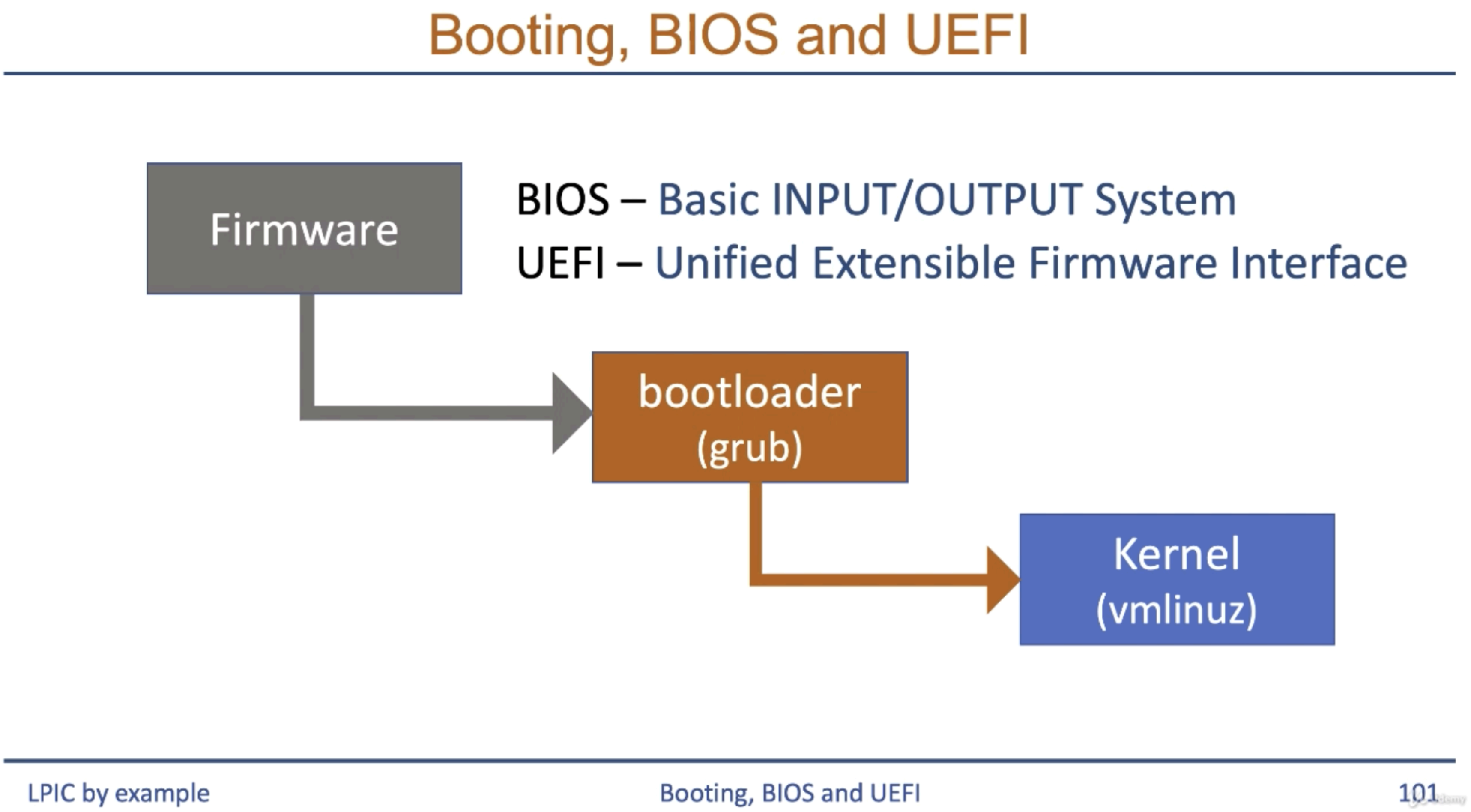
Power on System

Bootloader

Kernel

diagnostics

examples



Booting, BIOS and UEFI

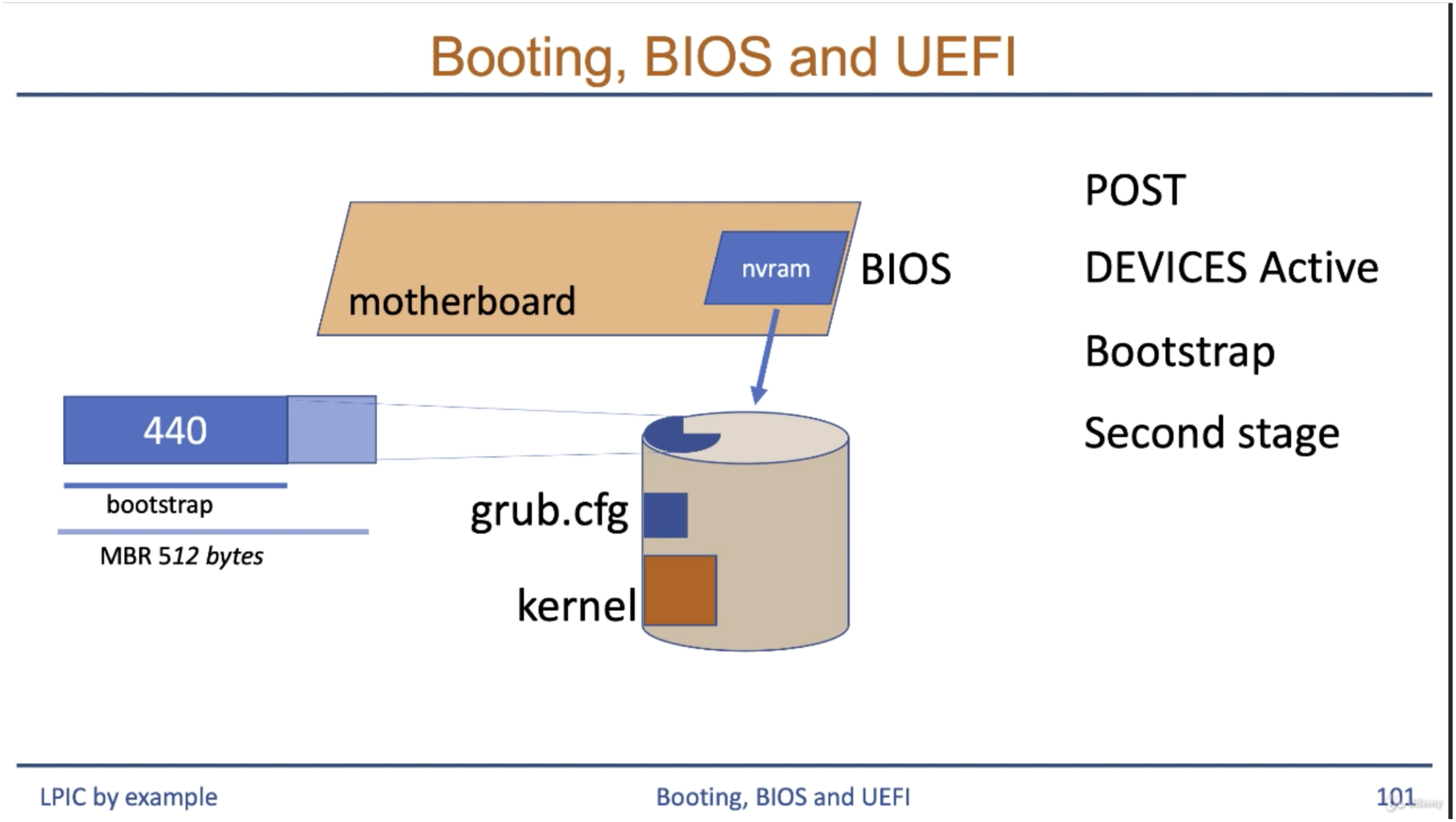
BIOS – Basic INPUT/OUTPUT System

UEFI – Unified Extensible Firmware Interface

Firmware

bootloader (grub)

Kernel (vmlinuz)



Booting, BIOS and UEFI

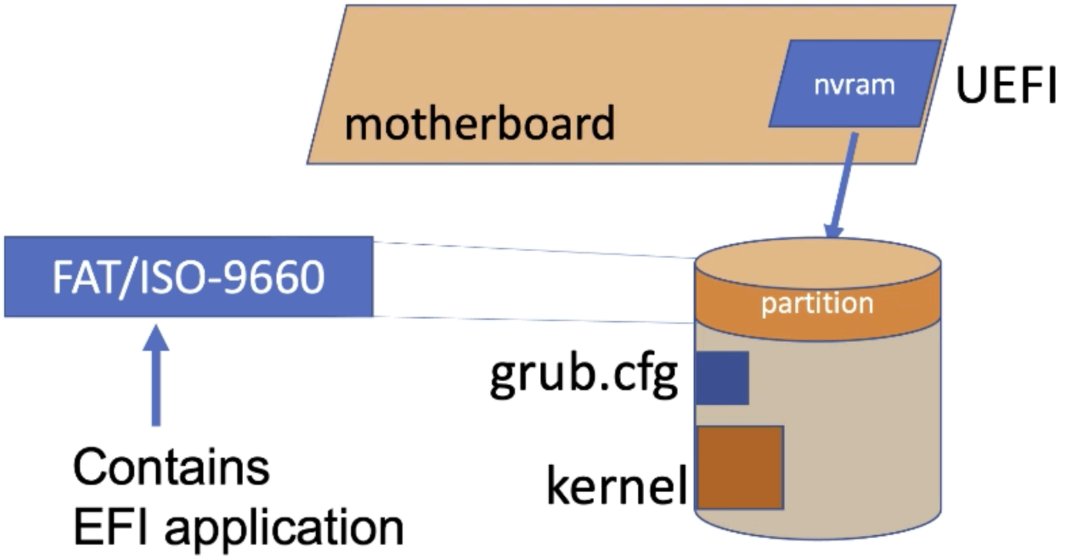
POST

DEVICES Active

Bootstrap

Second stage

Booting, BIOS and UEFI



POST

DEVICES Active

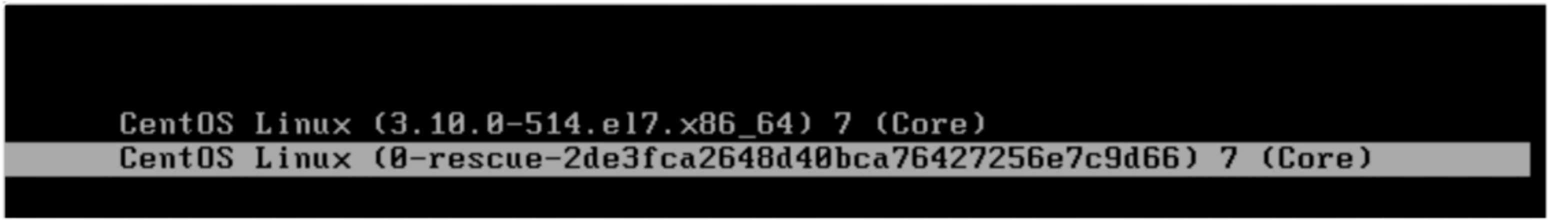
EFI Application

Reads grub.cfg

Booting, BIOS and UEFI

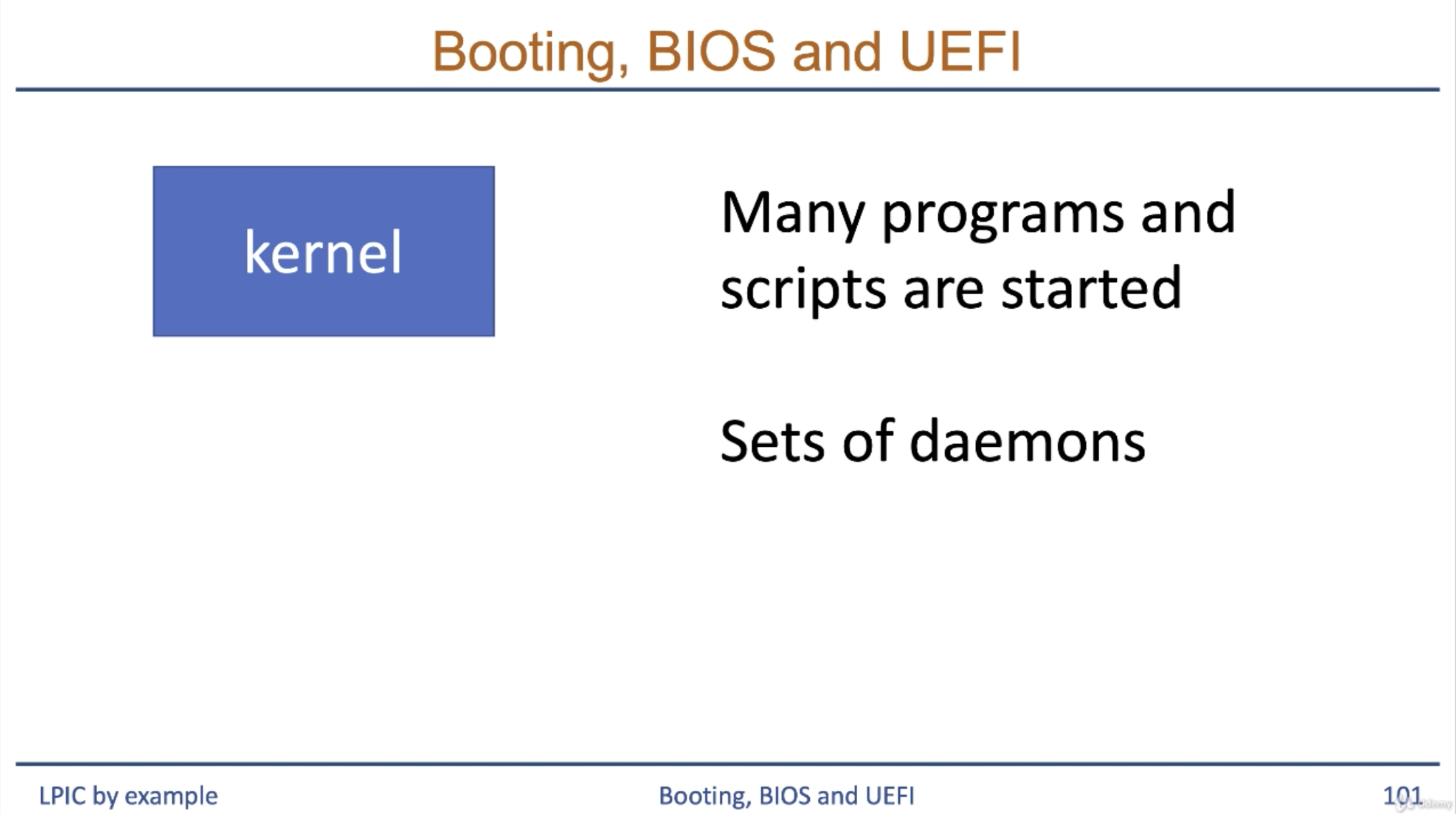
GRUB (GRand Unified Bootloader)

GRUB / GRUB2



```
CentOS Linux (3.10.0-514.el7.x86_64) 7 (Core)
CentOS Linux (0-rescue-2de3fca2648d40bca76427256e7c9d66) 7 (Core)
```

set parameters maxcpus



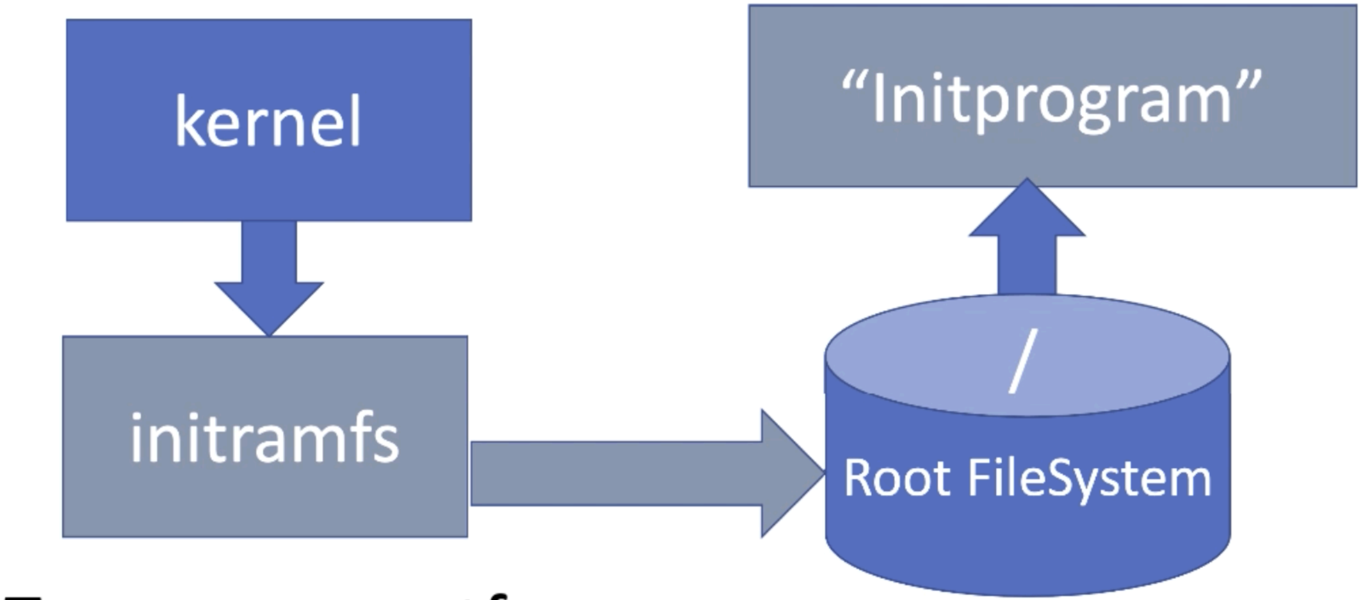
Booting, BIOS and UEFI

kernel

Many programs and scripts are started

Sets of daemons

Booting, BIOS and UEFI



kernel

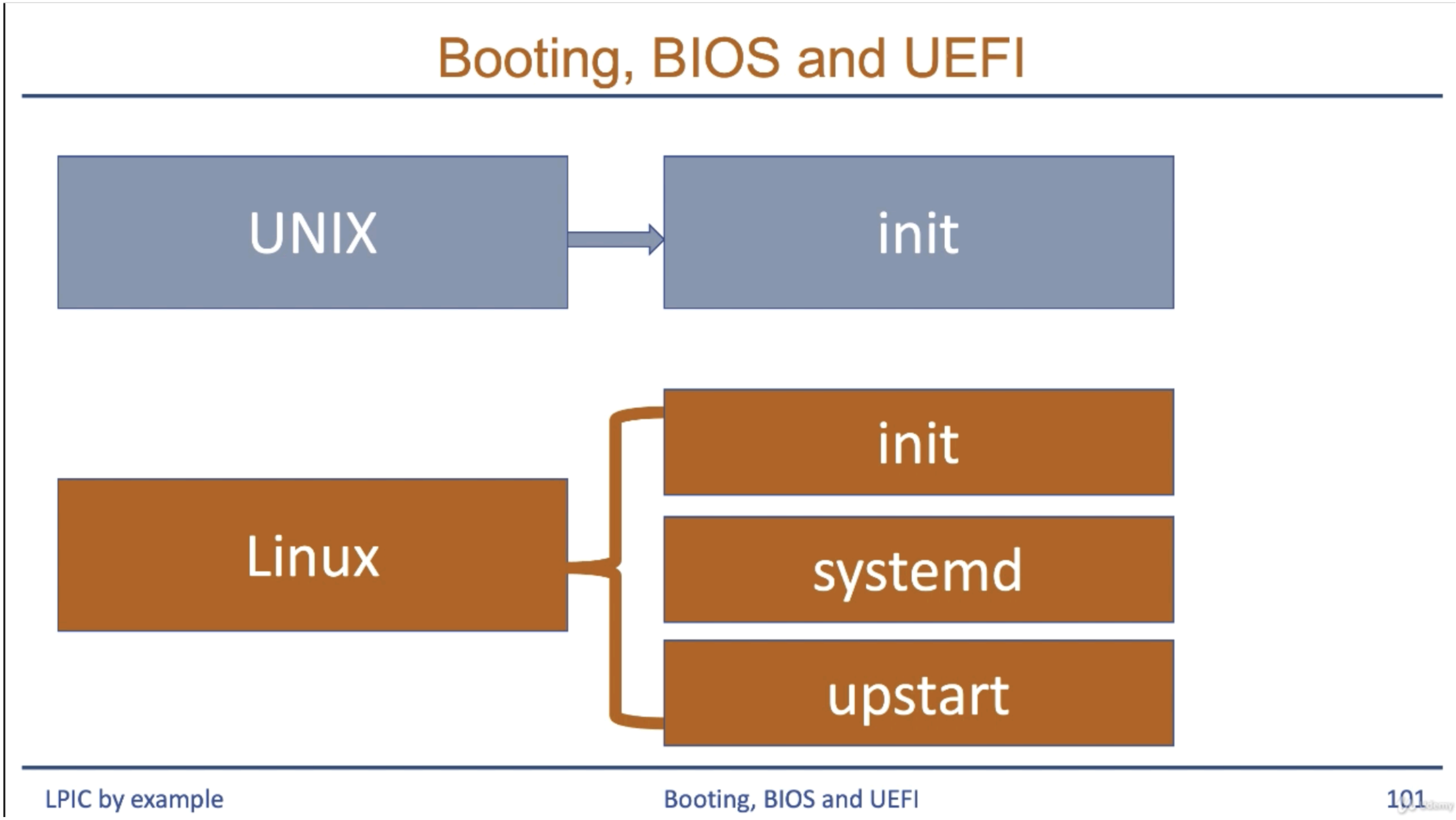
initramfs

“Initprogram”

/

Root FileSystem

Temporary rootfs



Booting, BIOS and UEFI

UNIX

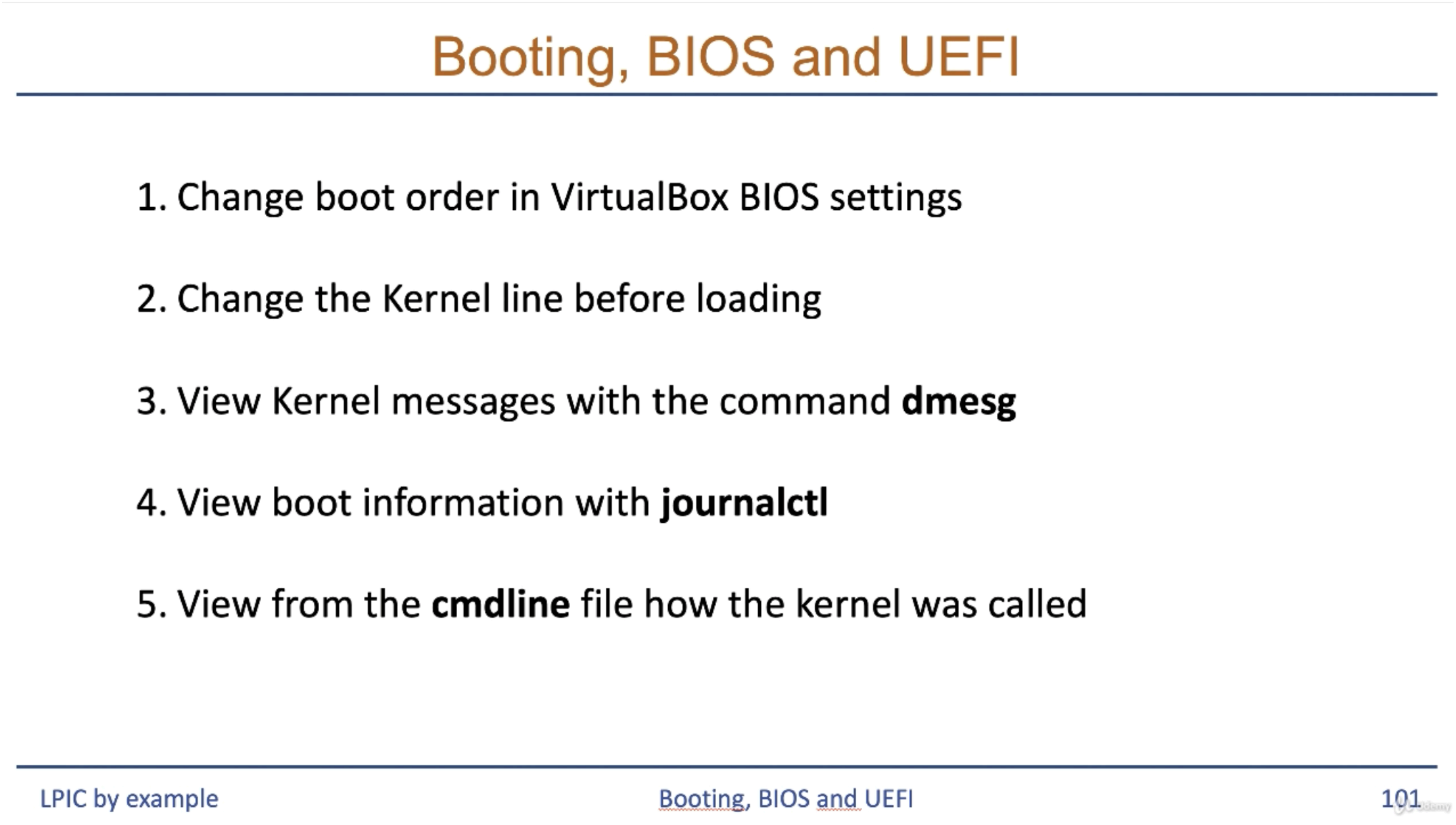
init

Linux

init

systemd

upstart



Booting, BIOS and UEFI

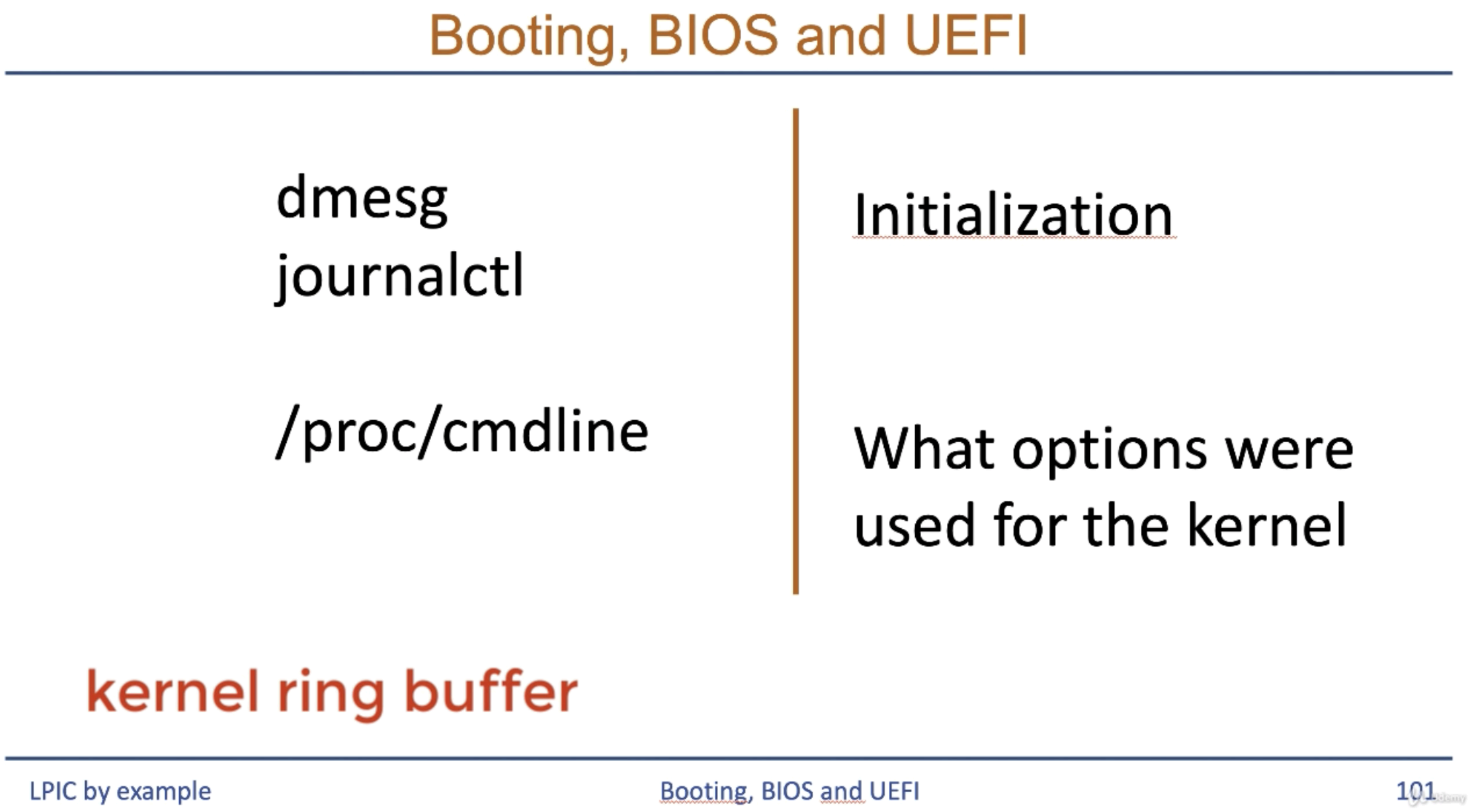
1. Change boot order in VirtualBox BIOS settings

2. Change the Kernel line before loading

3. View Kernel messages with the command **dmesg**

4. View boot information with **journalctl**

5. View from the **cmdline** file how the kernel was called



Booting, BIOS and UEFI

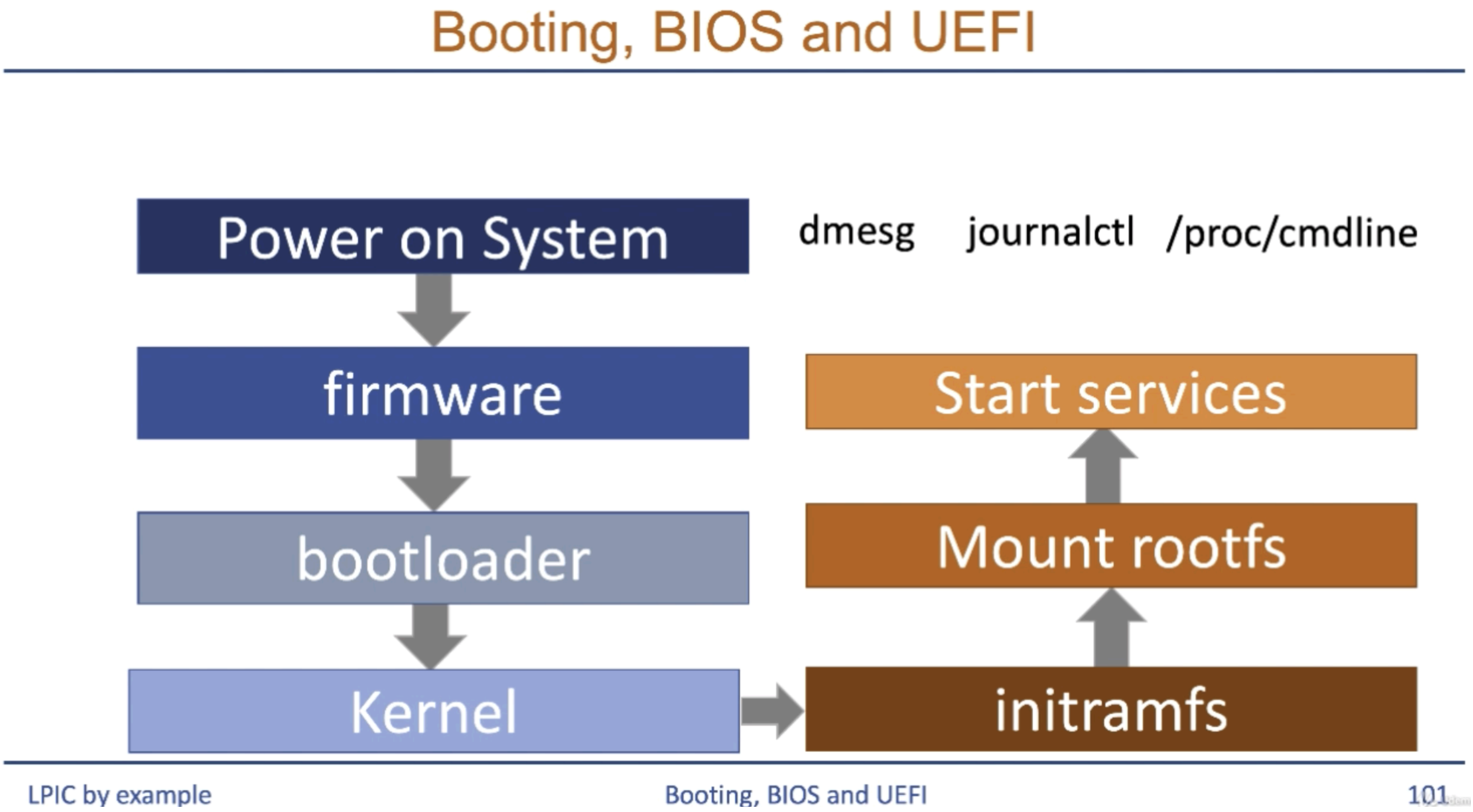
dmesg
journalctl

Initialization

/proc/cmdline

What options were used for the kernel

kernel ring buffer



Booting, BIOS and UEFI

Power on System

firmware

bootloader

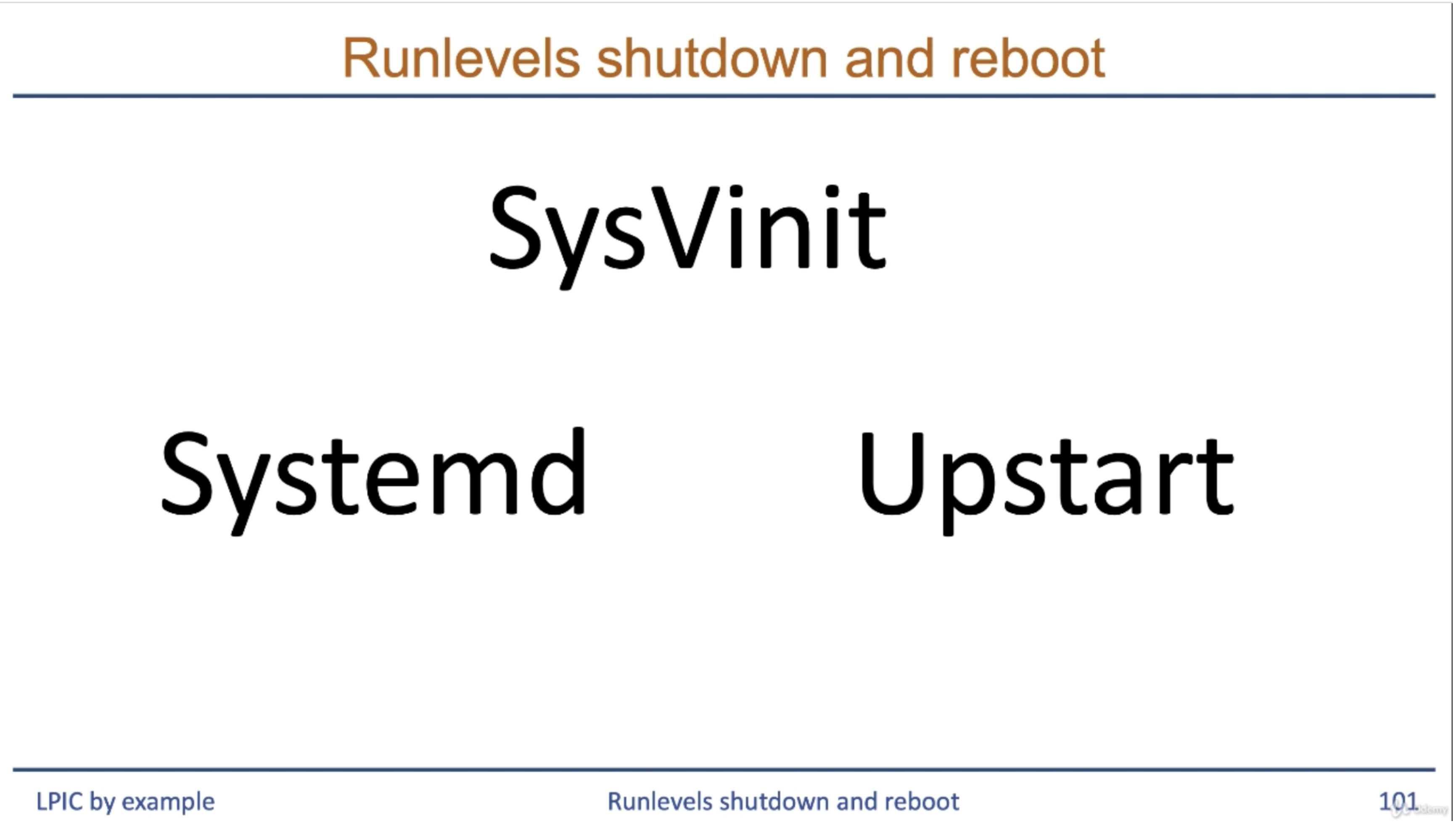
Kernel

initramfs

Mount rootfs

Start services

dmesg journalctl /proc/cmdline



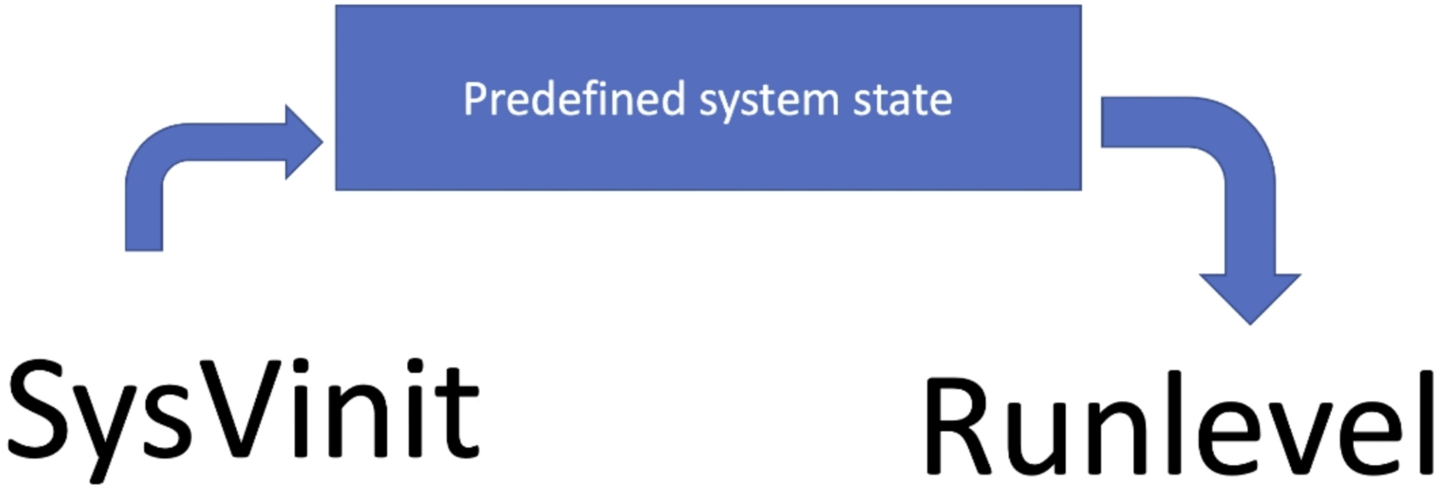
Runlevels shutdown and reboot

SysVinit

Systemd

Upstart

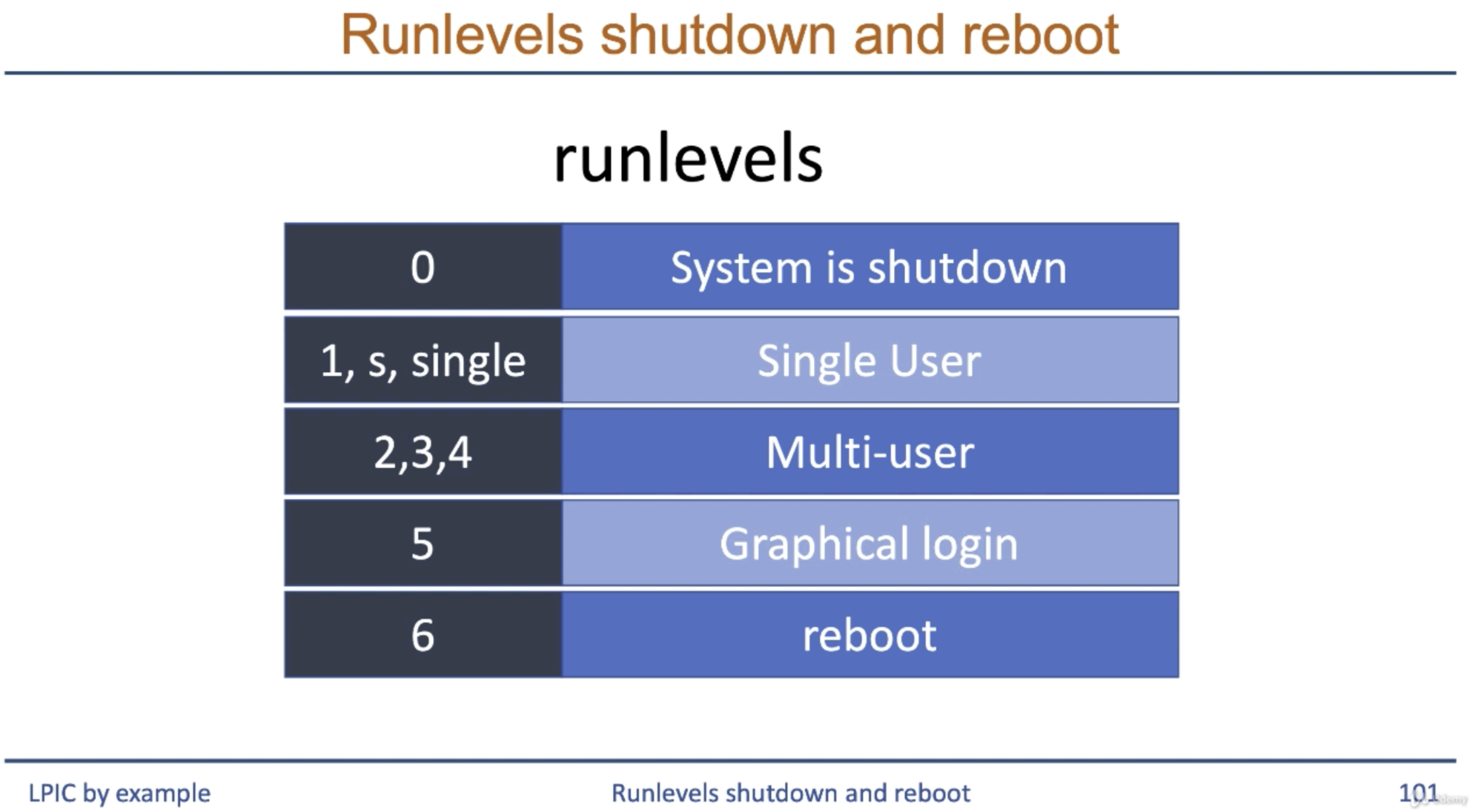
Runlevels shutdown and reboot



Predefined system state

SysVinit

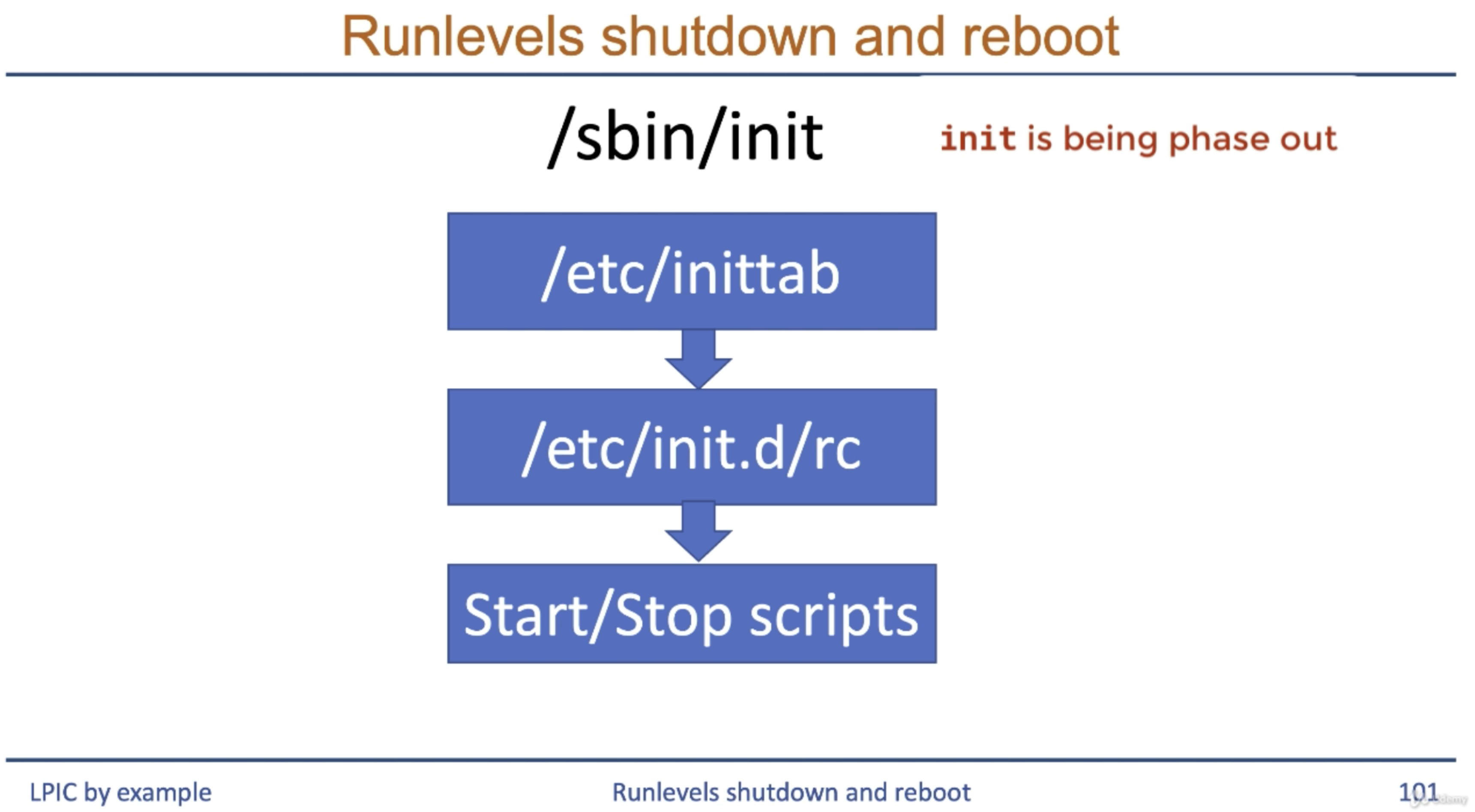
Runlevel



Runlevels shutdown and reboot

runlevels

0	System is shutdown
1, s, single	Single User
2,3,4	Multi-user
5	Graphical login
6	reboot



Runlevels shutdown and reboot

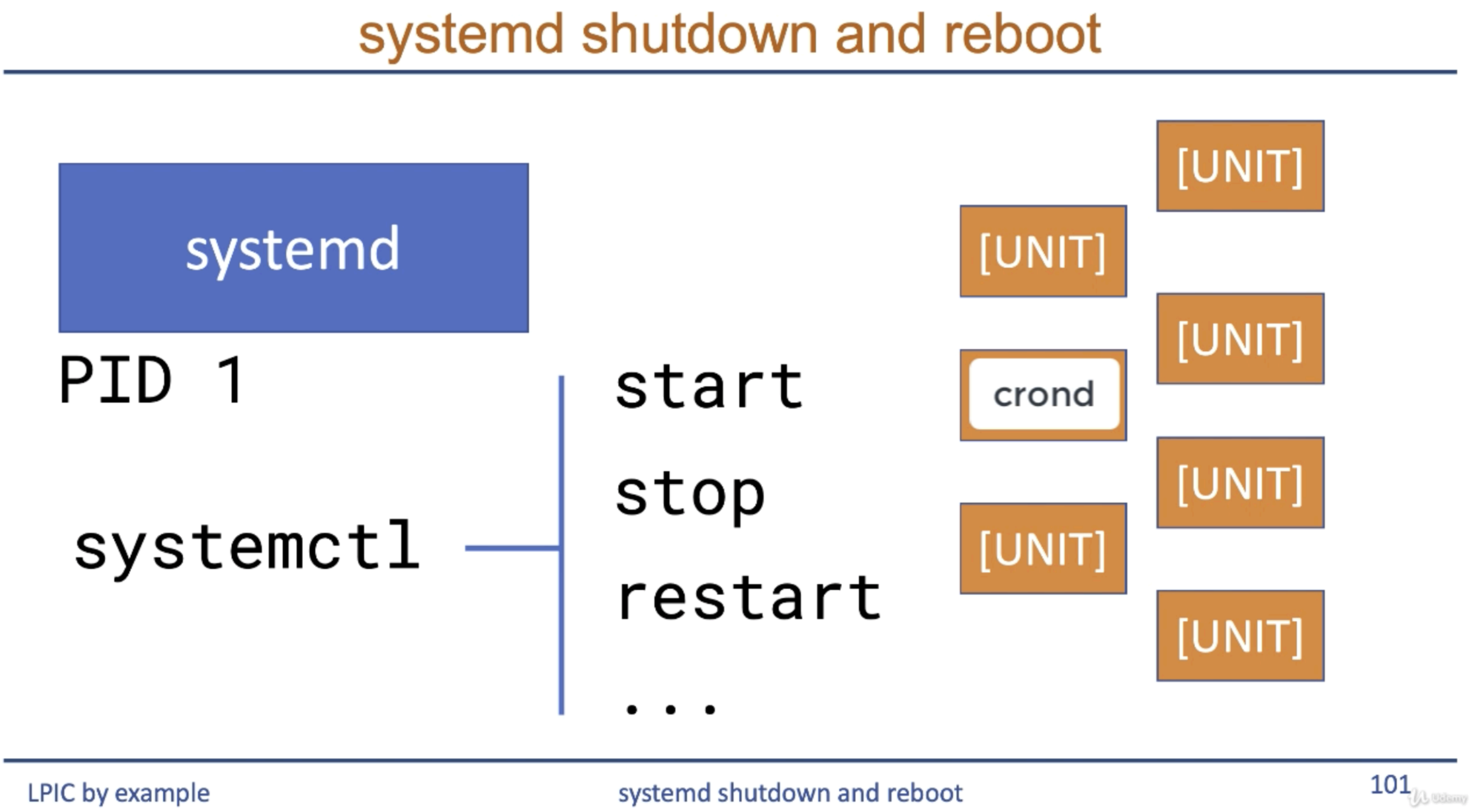
/sbin/init

init is being phase out

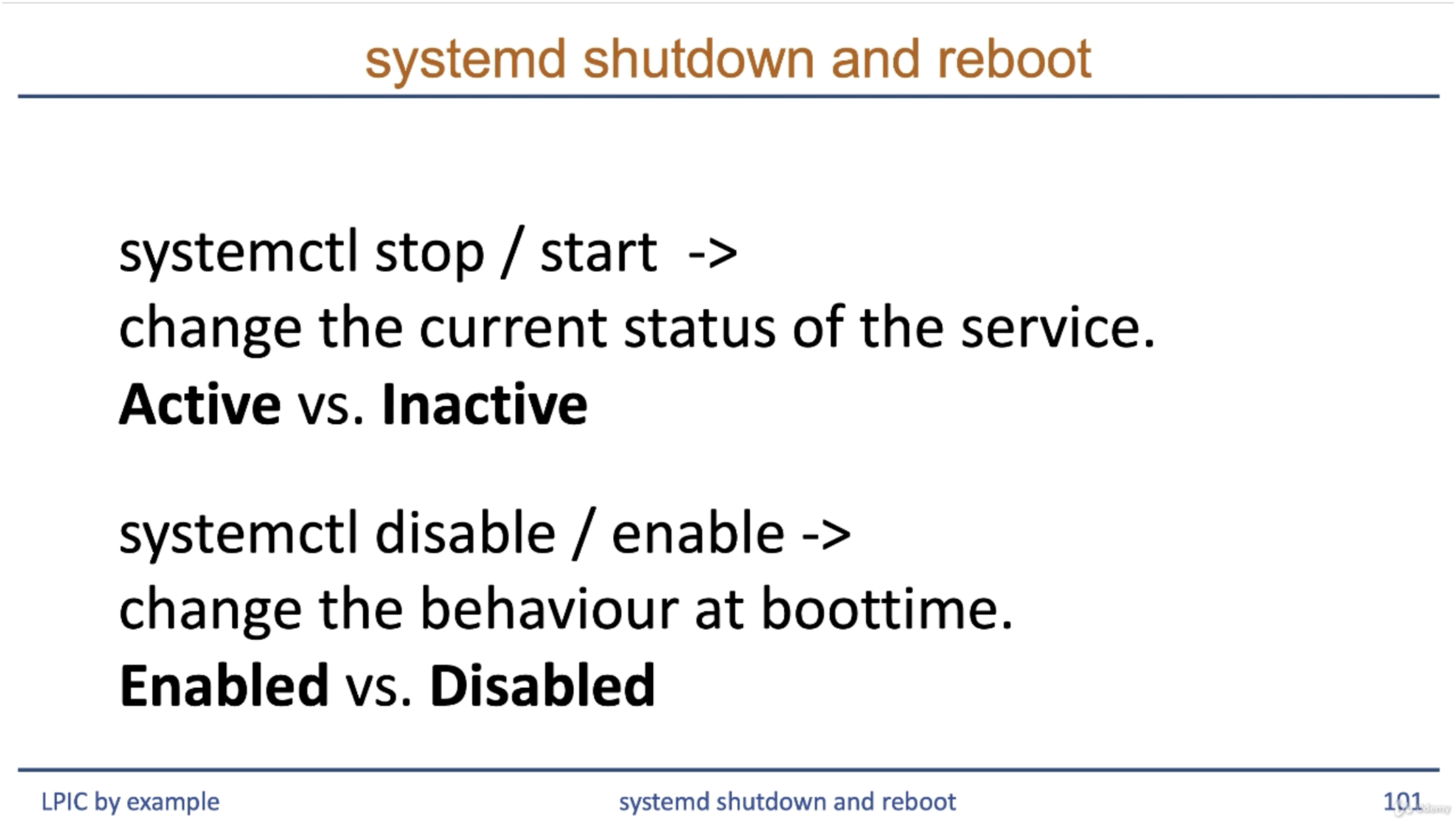
/etc/inittab

/etc/init.d/rc

Start/Stop scripts



systemd shutdown and reboot

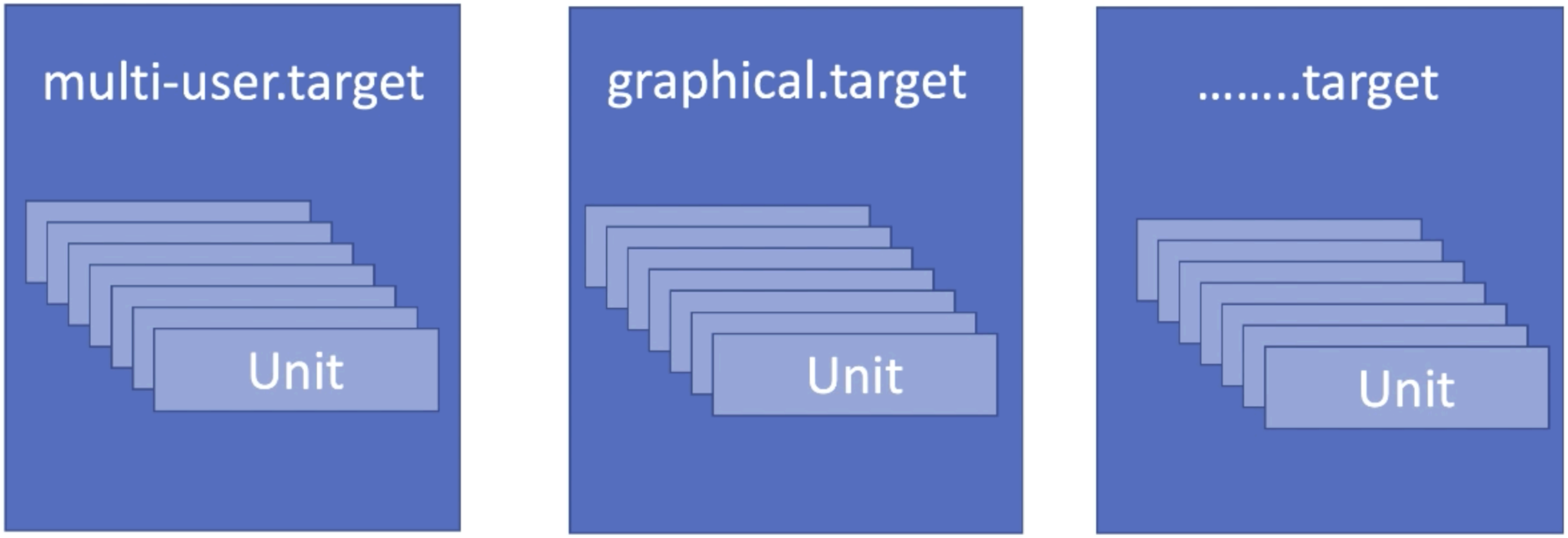


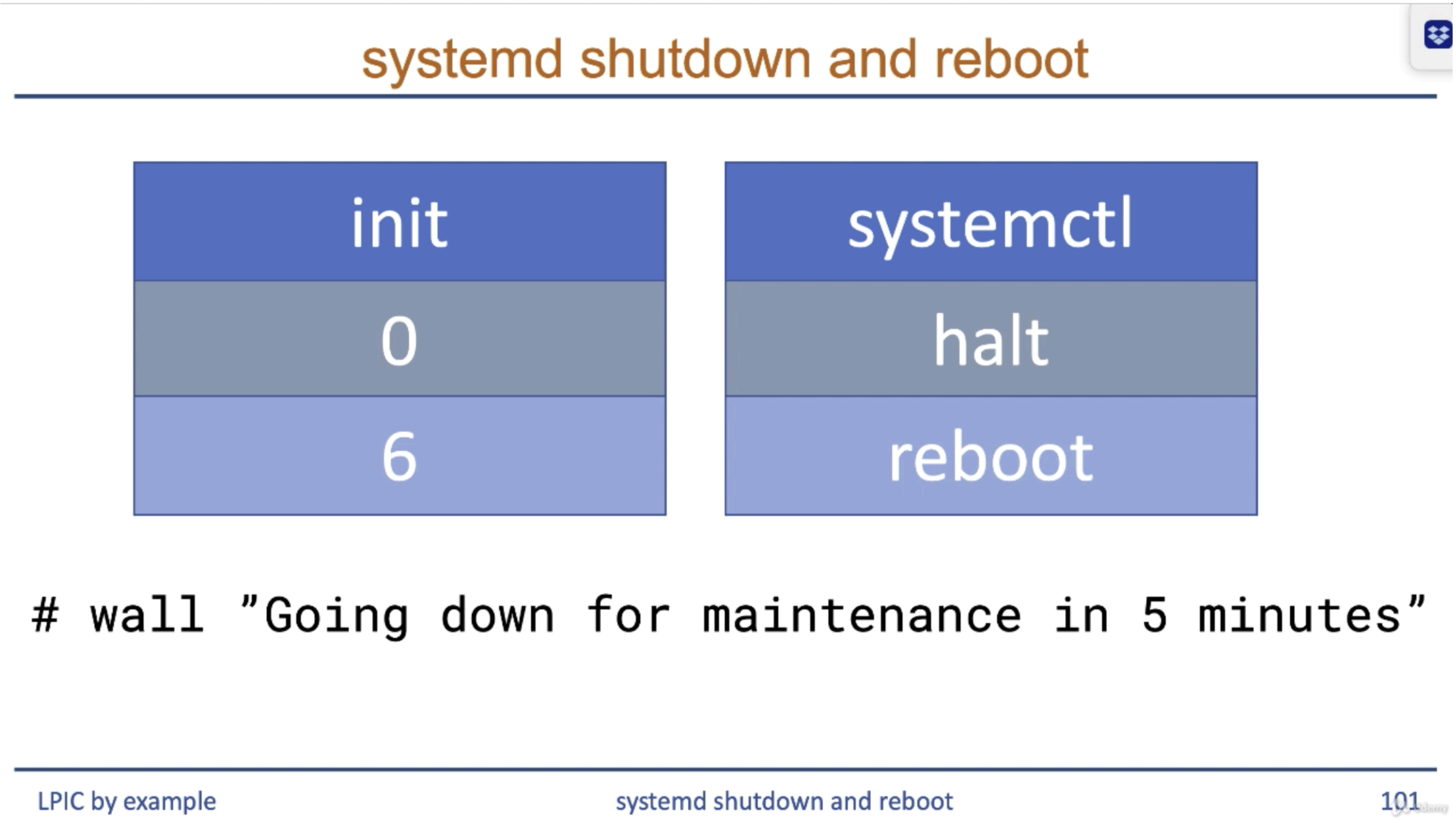
systemd shutdown and reboot

systemctl stop / start ->
change the current status of the service.
Active vs. **Inactive**

systemctl disable / enable ->
change the behaviour at boottime.
Enabled vs. **Disabled**

systemd shutdown and reboot



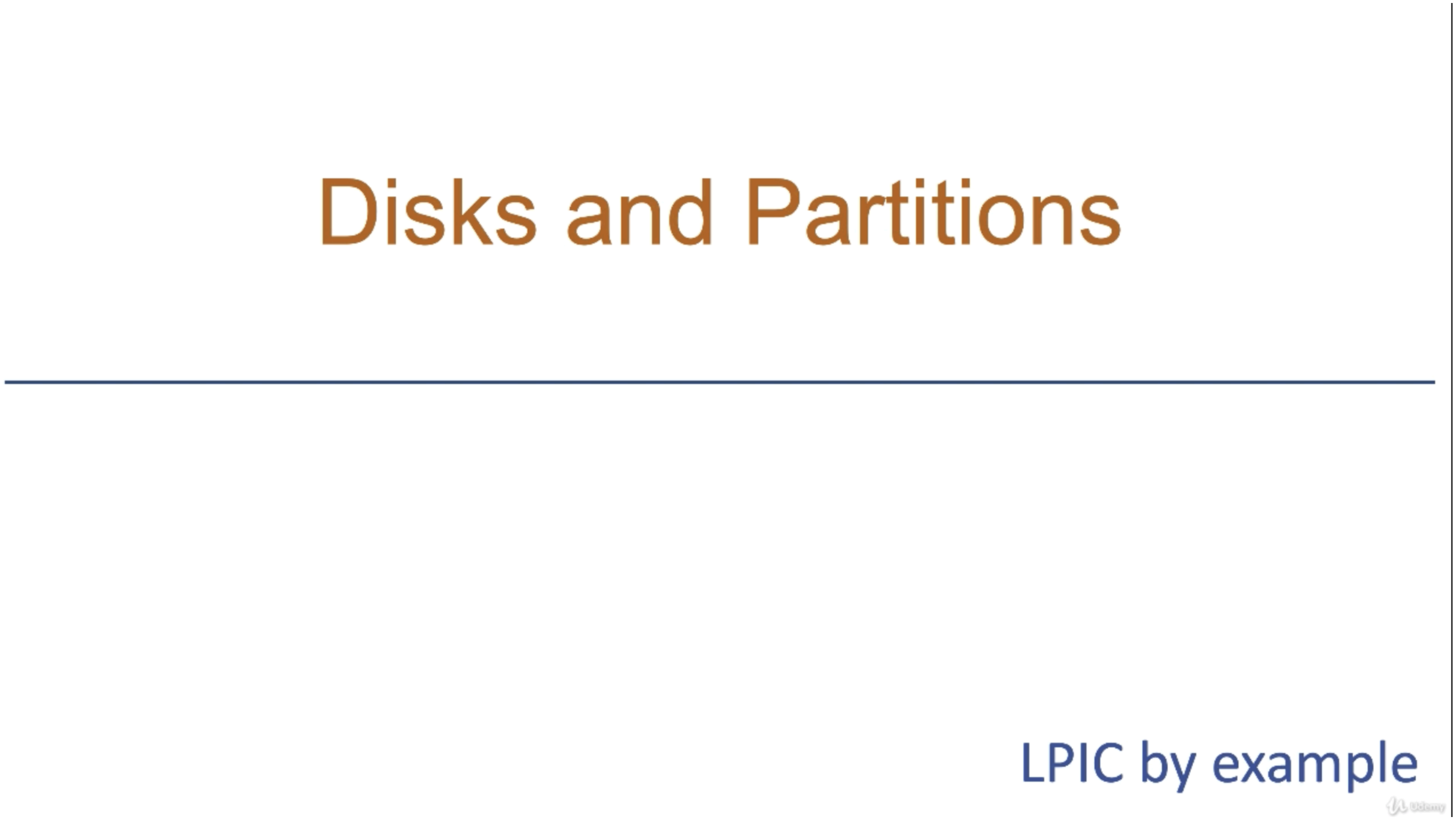


systemd shutdown and reboot

init
0
6

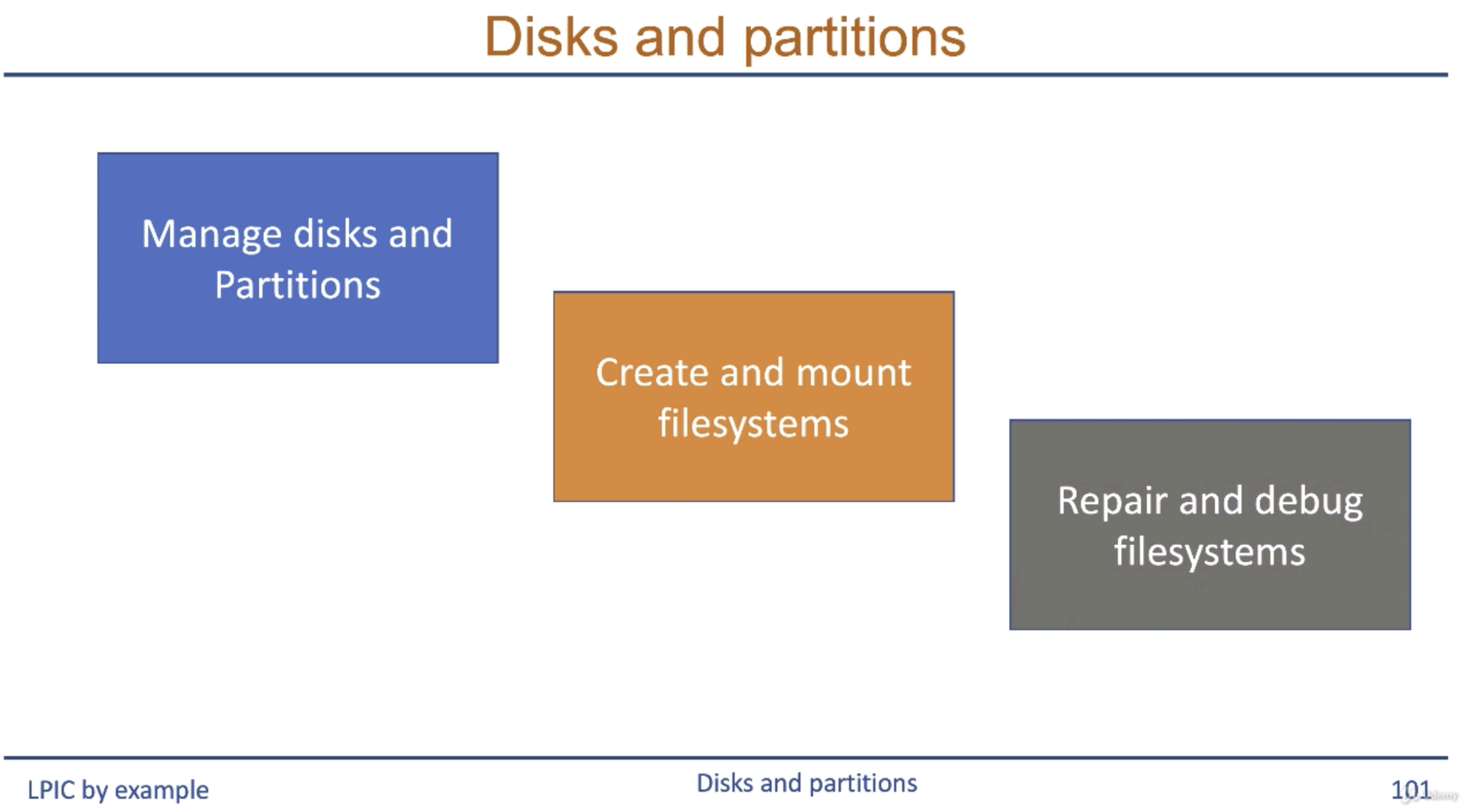
systemctl
halt
reboot

```
# wall "Going down for maintenance in 5 minutes"
```



Disks and Partitions

LPIC by example



Disks and partitions

Manage disks and Partitions

Create and mount filesystems

Repair and debug filesystems

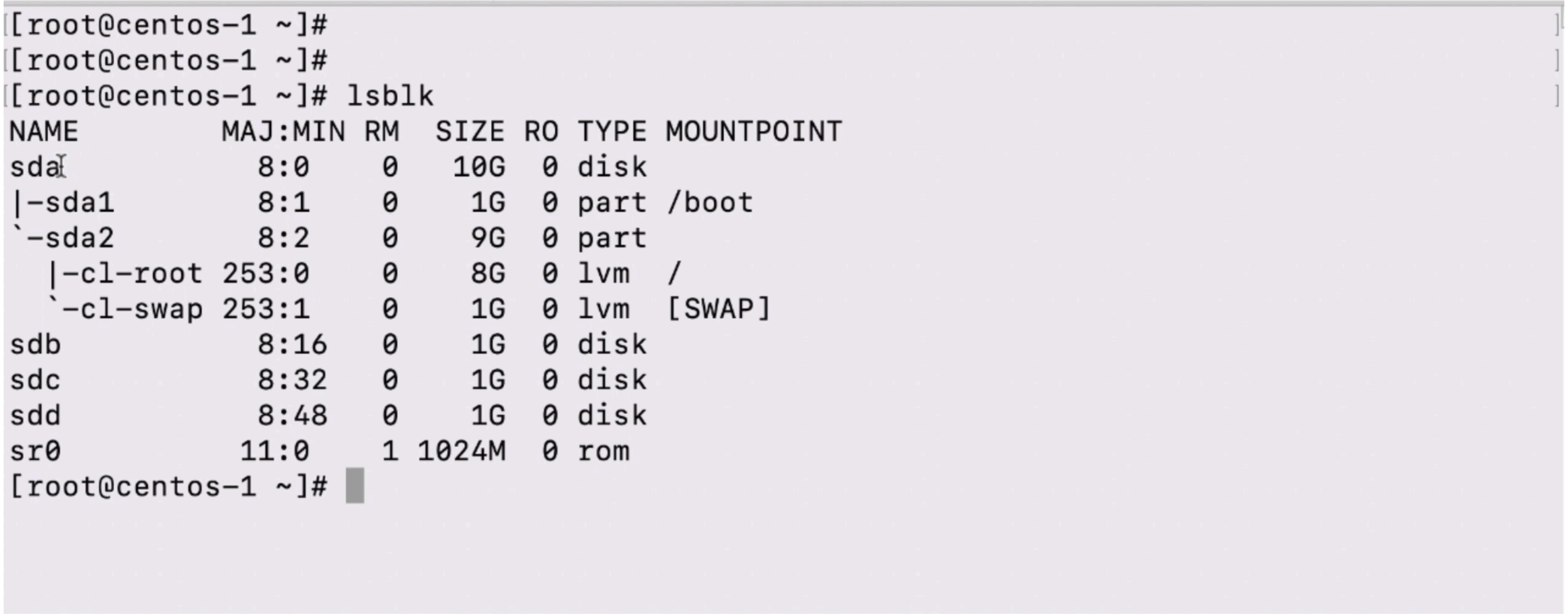
Disks and partitions

List the disks

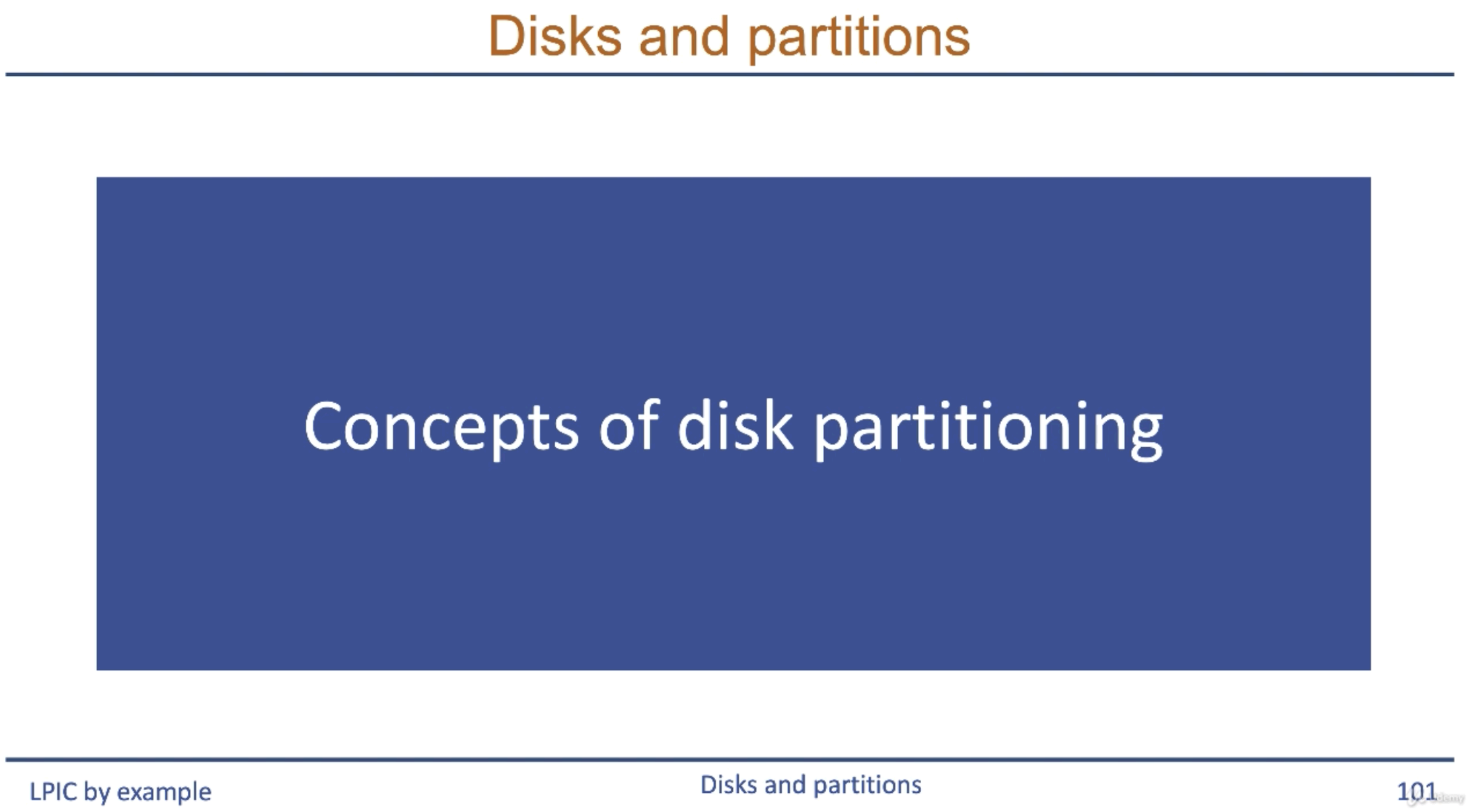
lsblk

fdisk

Disks and partitions

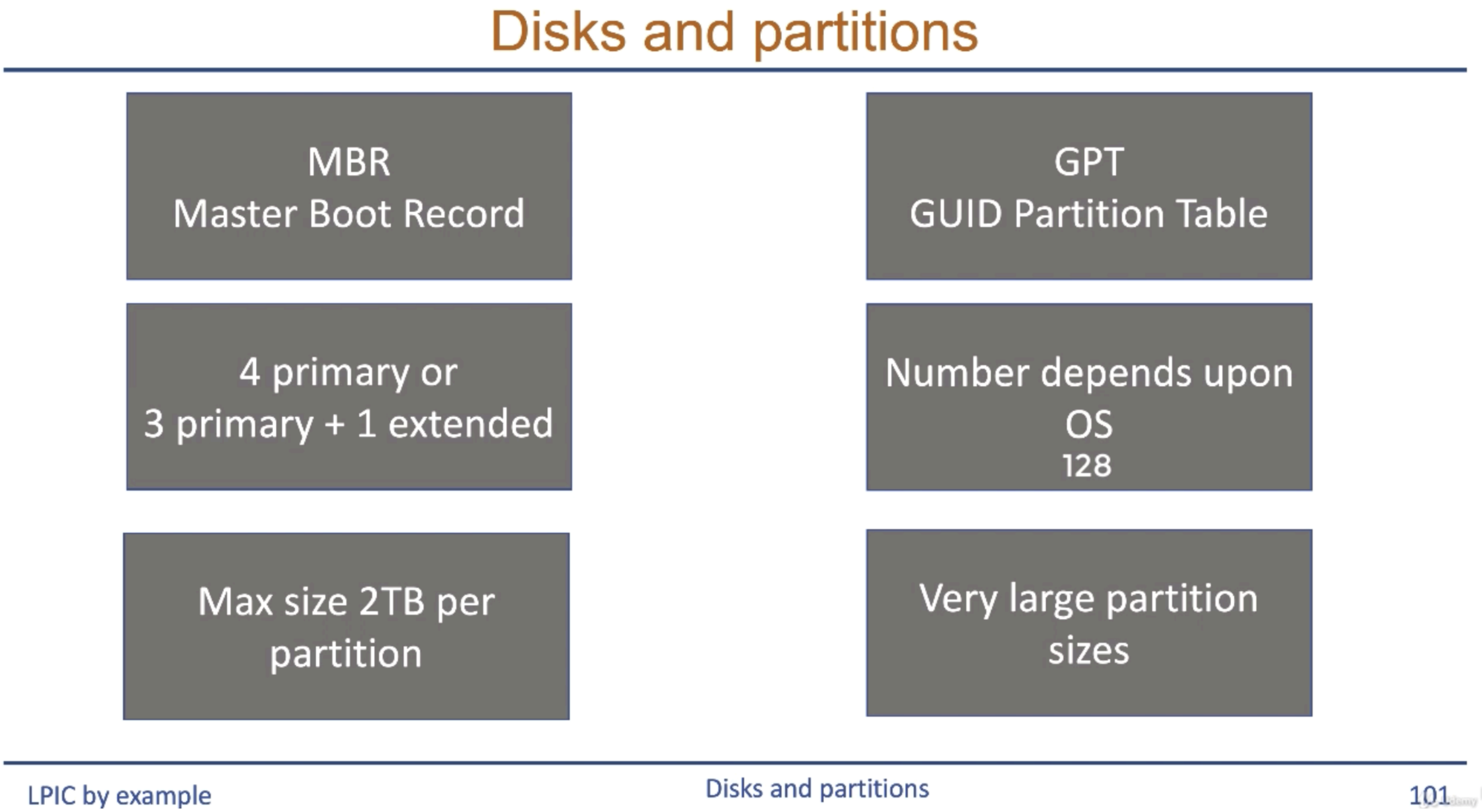


```
[root@centos-1 ~]#
[root@centos-1 ~]#
[root@centos-1 ~]# lsblk
NAME         MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda            8:0    0   10G  0 disk
|-sda1         8:1    0    1G  0 part /boot
`-sda2         8:2    0    9G  0 part
  |-cl-root 253:0    0    8G  0 lvm  /
  `-cl-swap 253:1    0    1G  0 lvm  [SWAP]
sdb            8:16   0    1G  0 disk
sdc            8:32   0    1G  0 disk
sdd            8:48   0    1G  0 disk
sr0           11:0    1 1024M  0 rom
[root@centos-1 ~]#
```



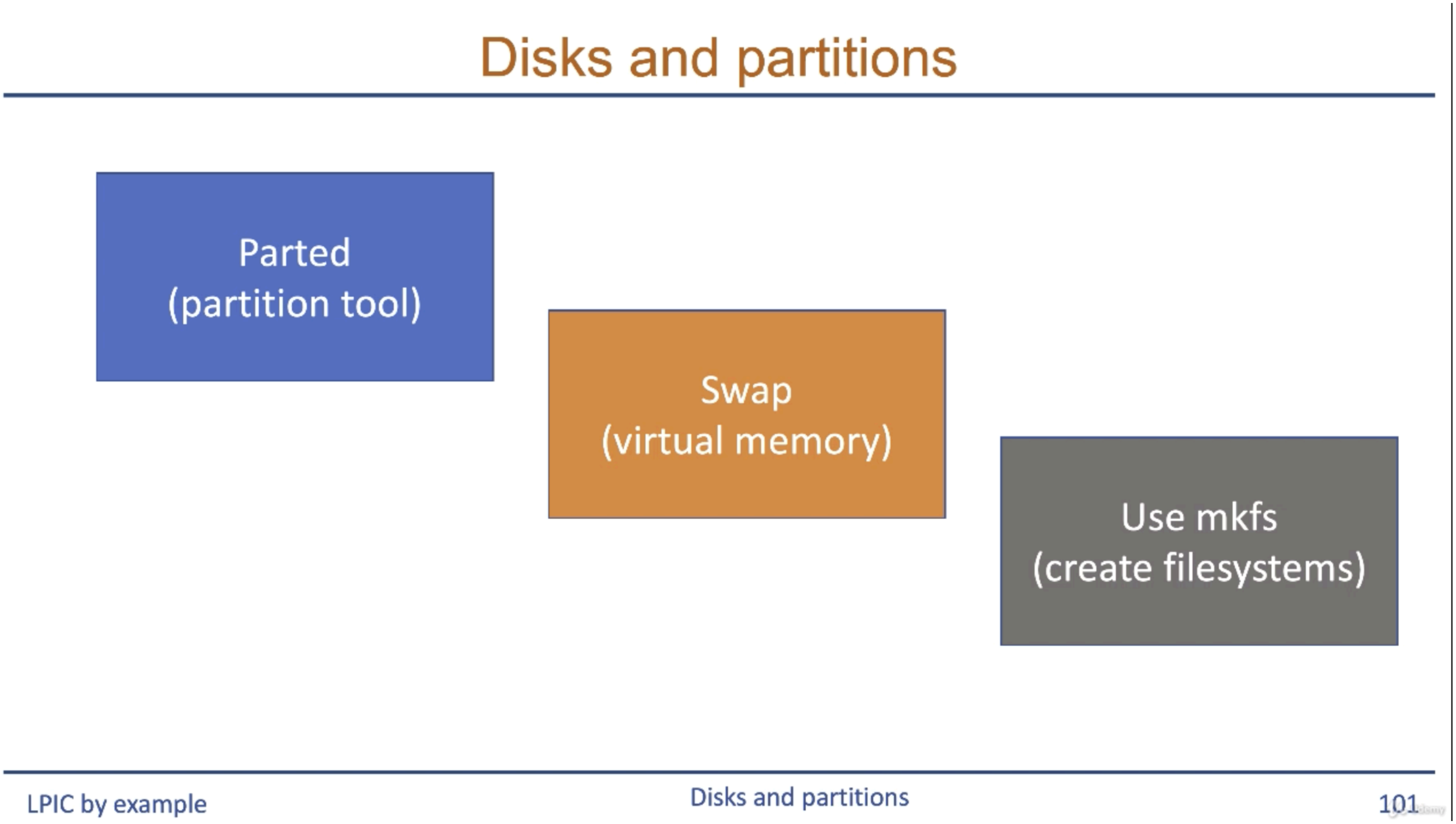
Disks and partitions

Concepts of disk partitioning



Disks and partitions

MBR Master Boot Record	GPT GUID Partition Table
4 primary or 3 primary + 1 extended	Number depends upon OS 128
Max size 2TB per partition	Very large partition sizes



Disks and partitions

Parted
(partition tool)

Swap
(virtual memory)

Use mkfs
(create filesystems)

Disks and partitions

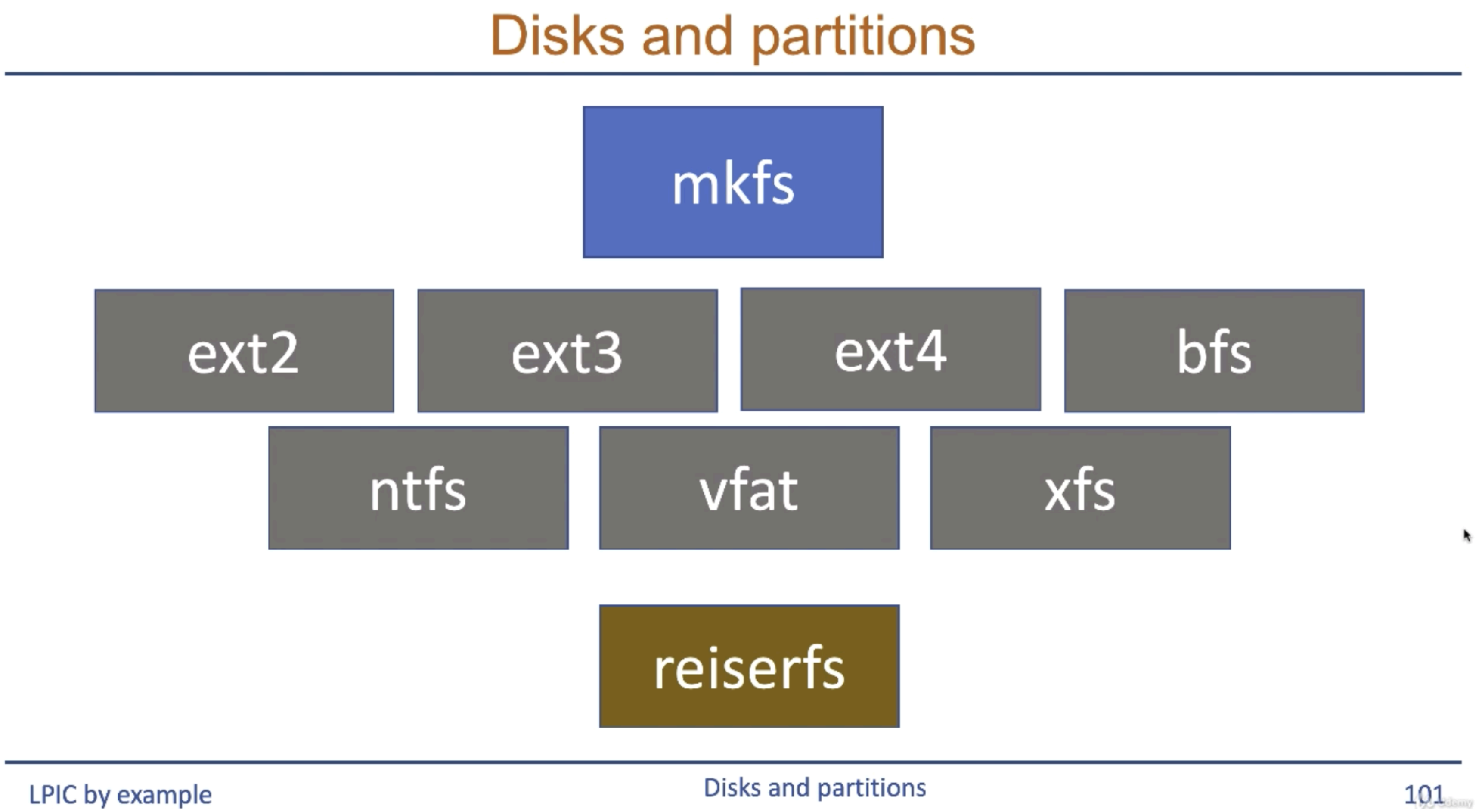
Swap

What is it?

Swap device or file

mkswap

swapon



Disks and partitions

mkfs

ext2

ext3

ext4

bfs

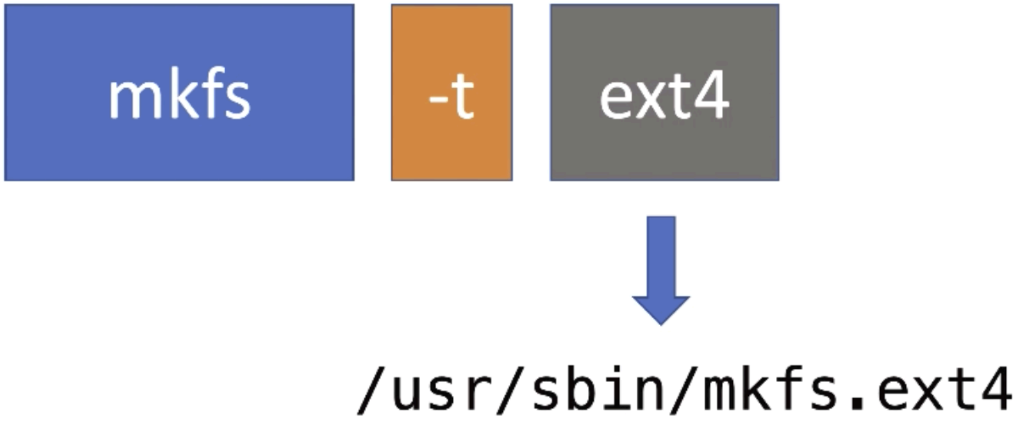
ntfs

vfat

xfs

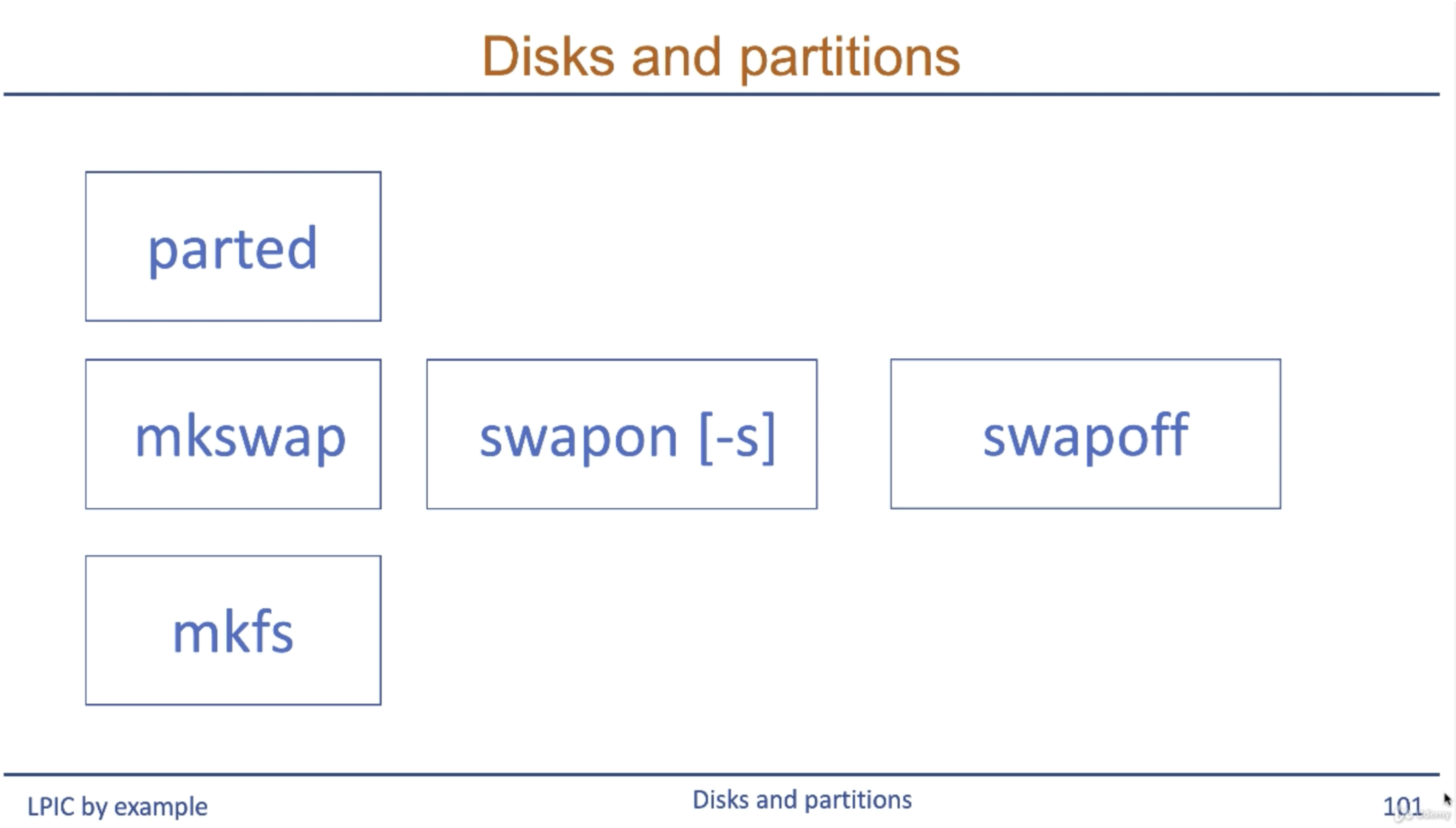
reiserfs

Disks and partitions



mkfs -t ext4

/usr/sbin/mkfs.ext4



Disks and partitions

parted

mkswap

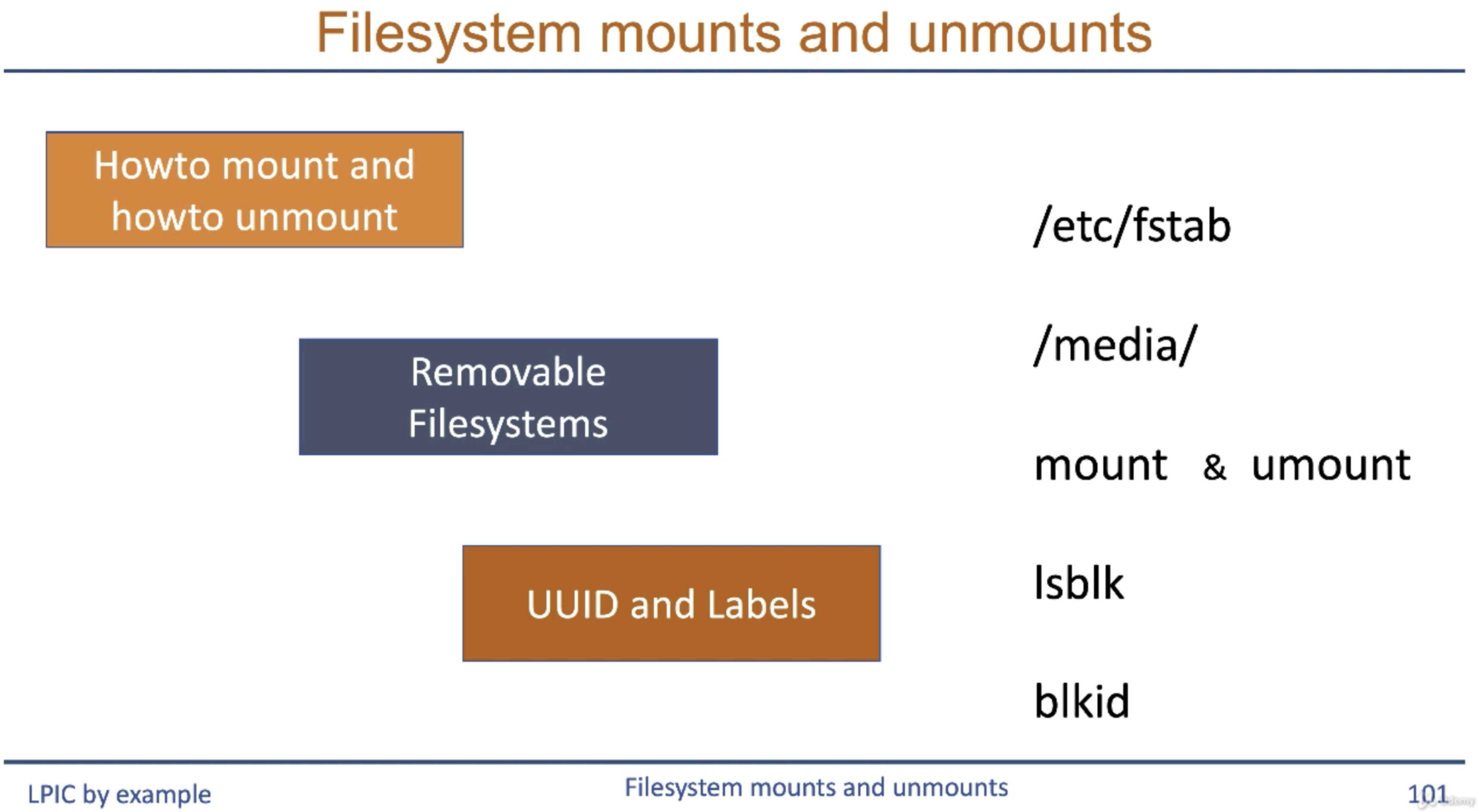
swapon [-s]

swapoff

mkfs

Filesystem mounts and unmounts

LPIC by example



Filesystem mounts and unmounts

Howto mount and howto unmount

Removable Filesystems

UUID and Labels

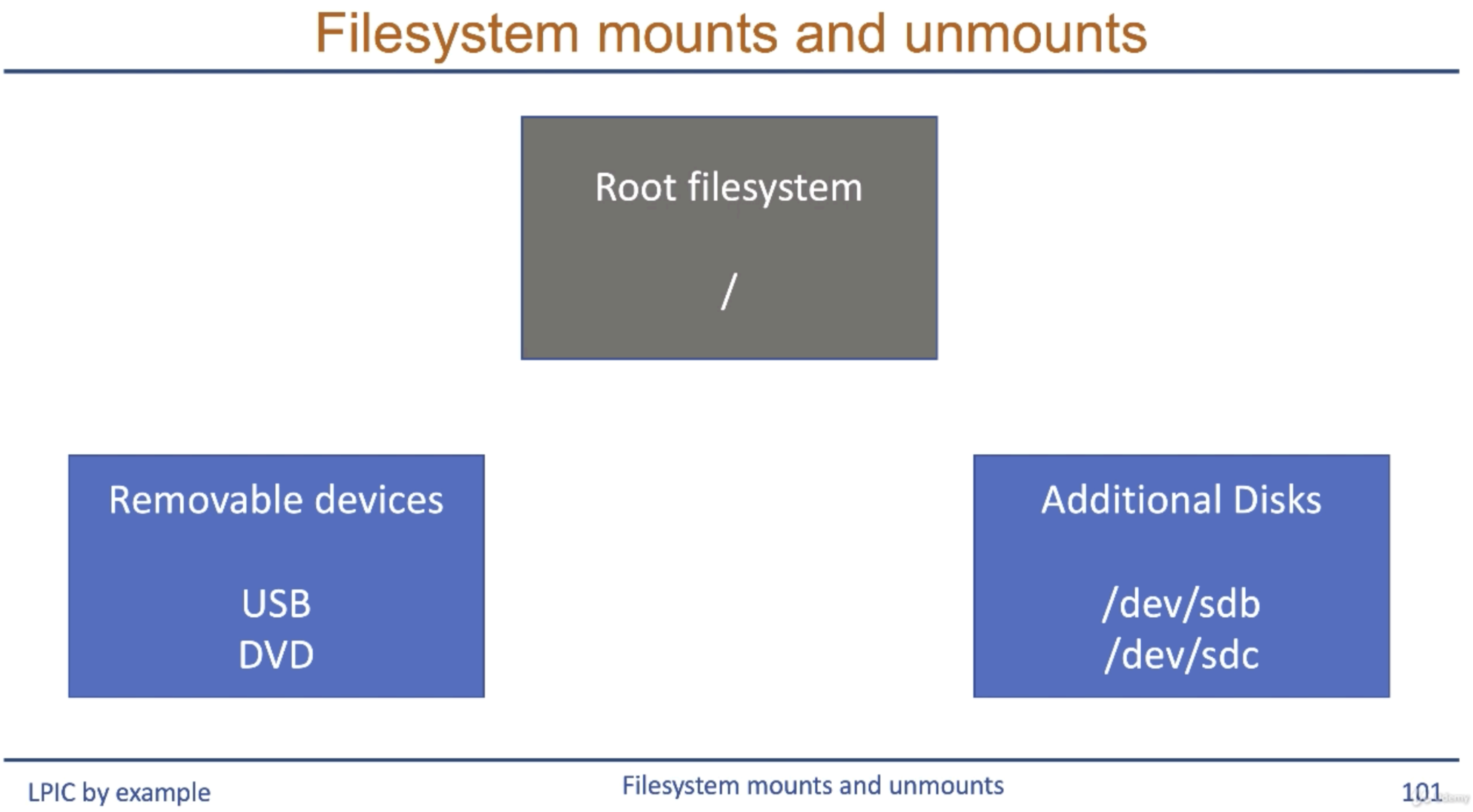
/etc/fstab

/media/

mount & umount

lsblk

blkid



Filesystem mounts and unmounts

Root filesystem

/

Removable devices

USB
DVD

Additional Disks

/dev/sdb
/dev/sdc

Filesystem mounts and unmounts

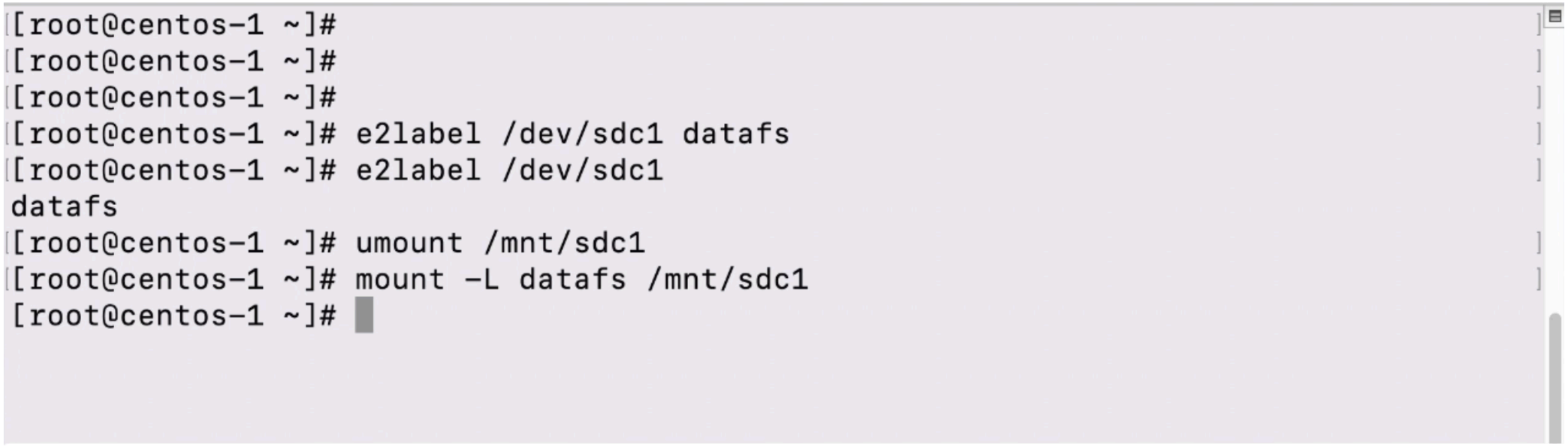
LABEL is a Filesystem property

Filesystems can be mounted using a LABEL instead of a devicename

Command differs per filesystem:

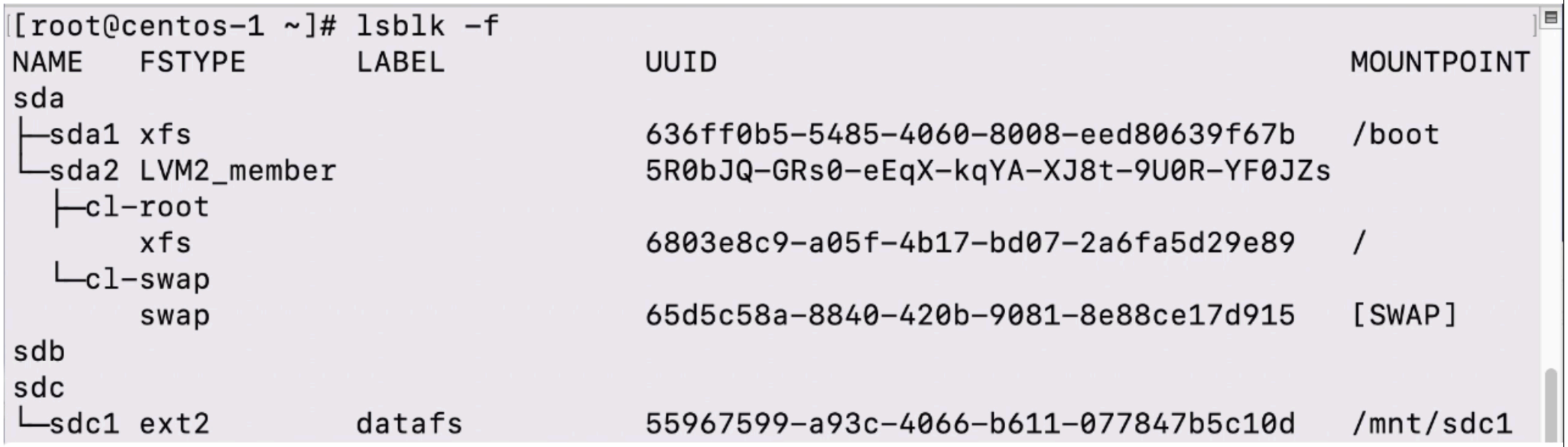
ext2, ext3, ext4 - e2label

Filesystem mounts and unmounts

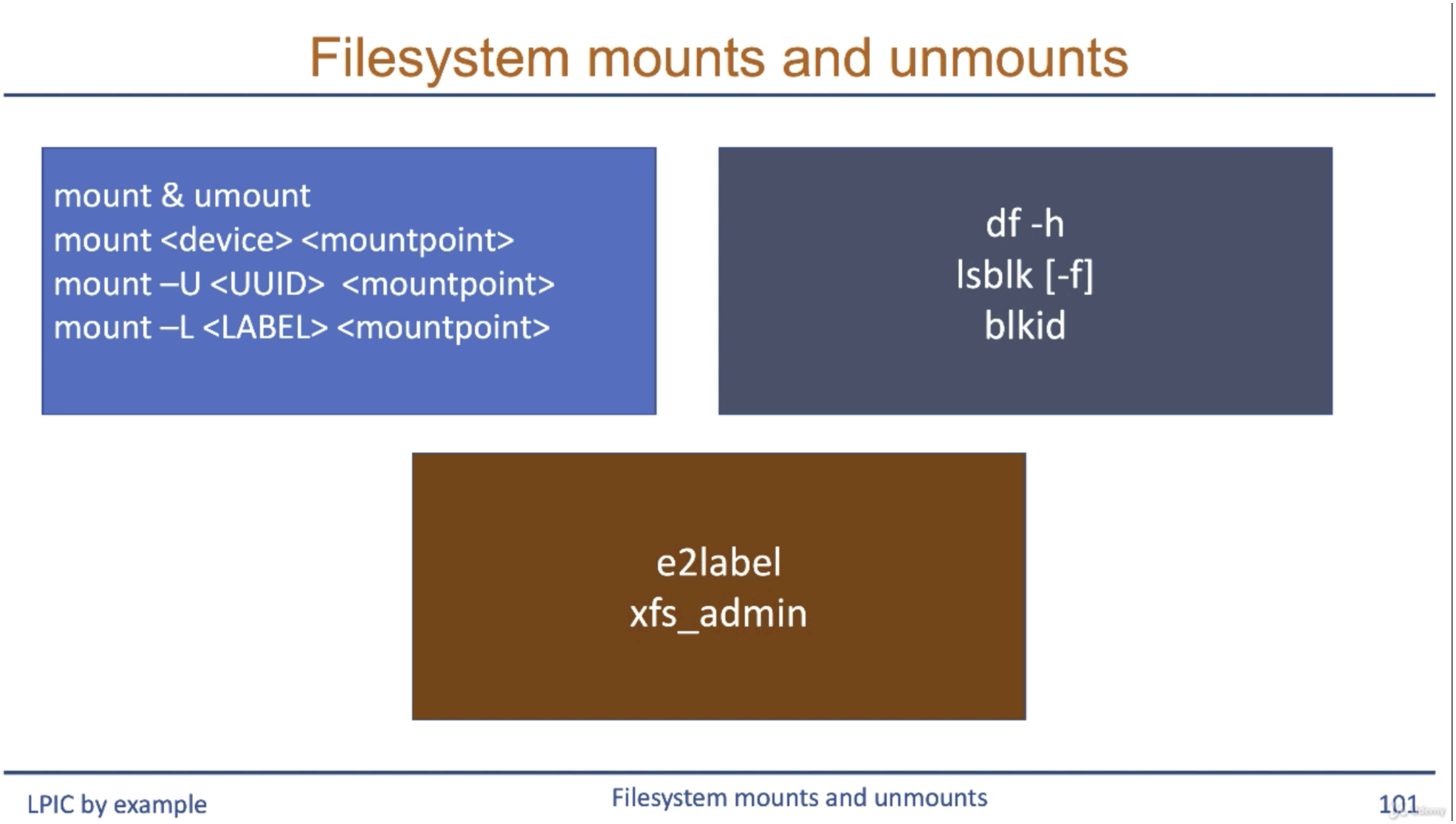


```
[root@centos-1 ~]#
[root@centos-1 ~]#
[root@centos-1 ~]#
[root@centos-1 ~]# e2label /dev/sdc1 datafs
[root@centos-1 ~]# e2label /dev/sdc1
datafs
[root@centos-1 ~]# umount /mnt/sdc1
[root@centos-1 ~]# mount -L datafs /mnt/sdc1
[root@centos-1 ~]# 
```

Filesystem mounts and unmounts



```
[root@centos-1 ~]# lsblk -f
NAME   FSTYPE      LABEL           UUID                                   MOUNTPOINT
sda
├─sda1 xfs                         636ff0b5-5485-4060-8008-eed80639f67b   /boot
└─sda2 LVM2_member                 5R0bJQ-GRs0-eEqX-kqYA-XJ8t-9U0R-YF0JZs
  ├─cl-root
       xfs                         6803e8c9-a05f-4b17-bd07-2a6fa5d29e89   /
  └─cl-swap
       swap                        65d5c58a-8840-420b-9081-8e88ce17d915   [SWAP]
sdb
sdc
└─sdc1 ext2        datafs          55967599-a93c-4066-b611-077847b5c10d   /mnt/sdc1
```

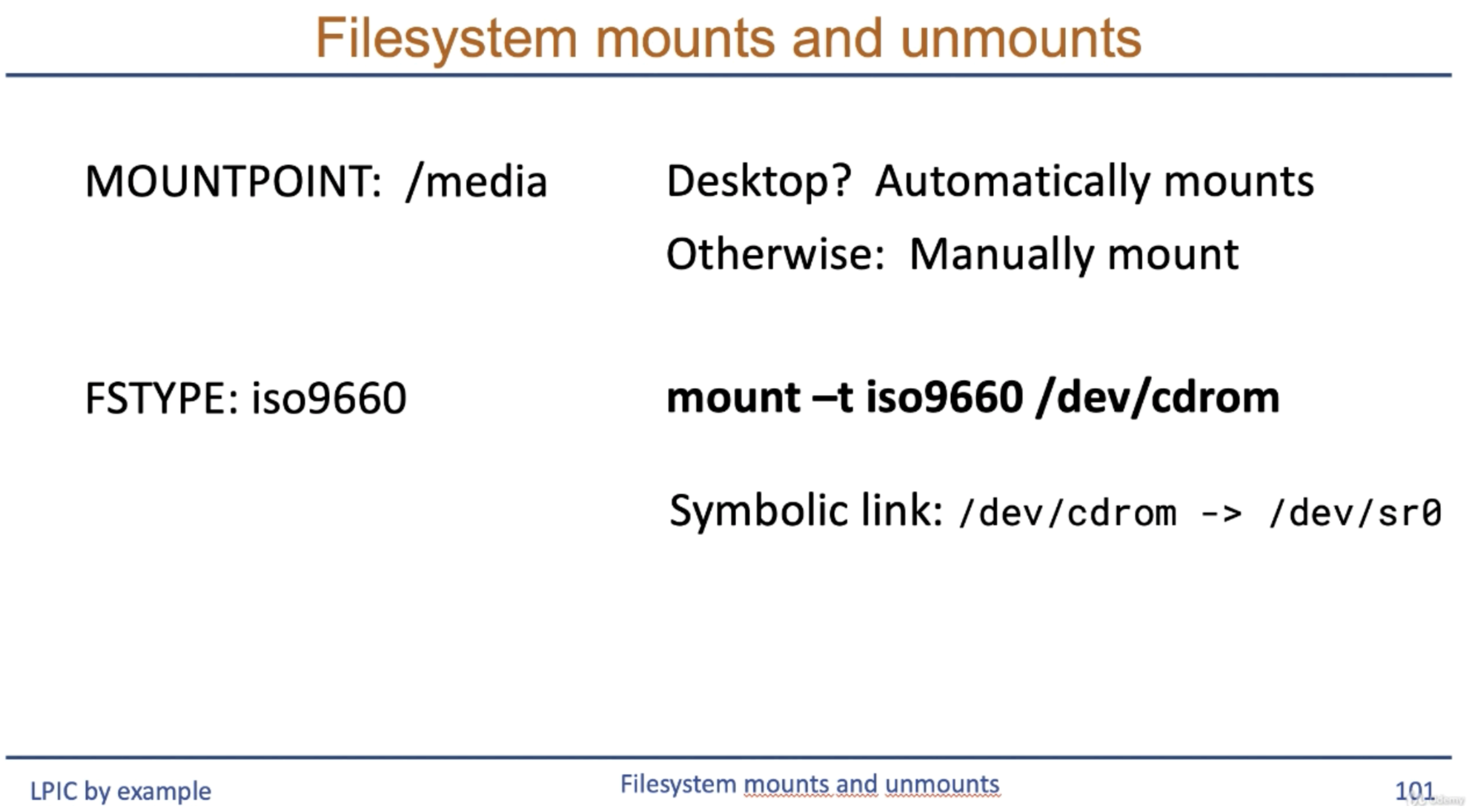


Filesystem mounts and unmounts

mount & umount
mount <device> <mountpoint>
mount –U <UUID> <mountpoint>
mount –L <LABEL> <mountpoint>

df -h
lsblk [-f]
blkid

e2label
xfs_admin



Filesystem mounts and unmounts

MOUNTPOINT: /media

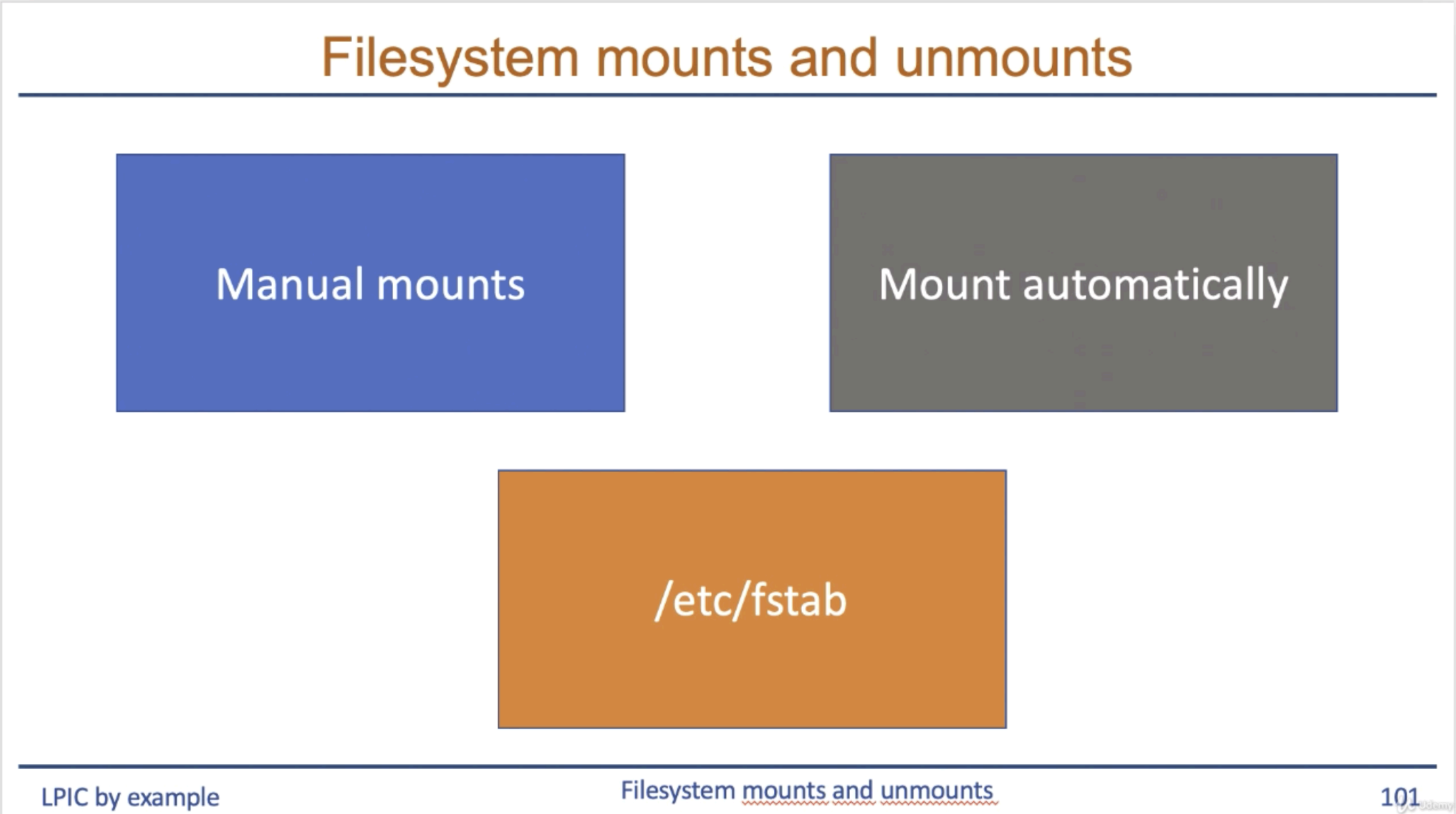
Desktop? Automatically mounts

Otherwise: Manually mount

FSTYPE: iso9660

mount –t iso9660 /dev/cdrom

Symbolic link: `/dev/cdrom -> /dev/sr0`

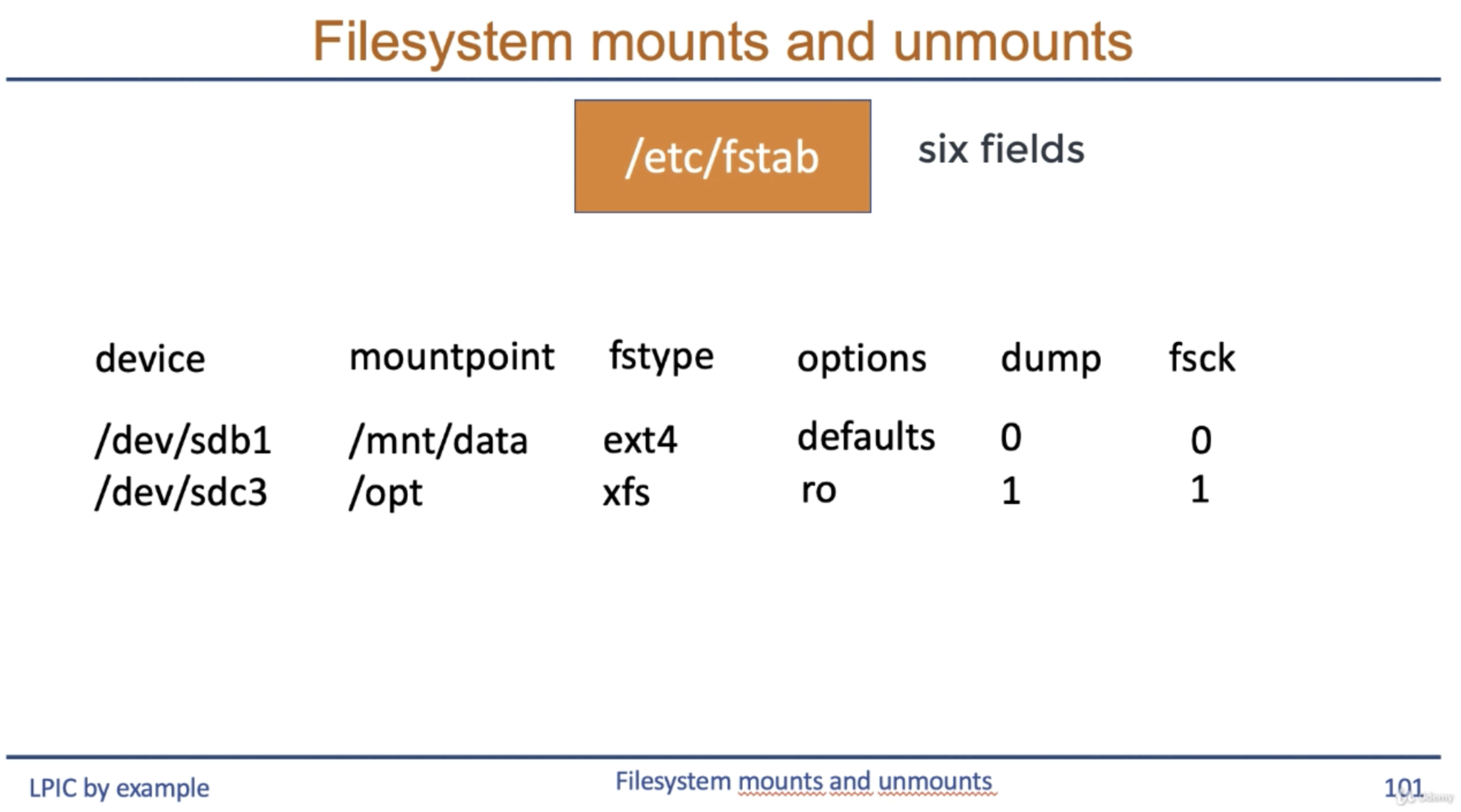


Filesystem mounts and unmounts

Manual mounts

Mount automatically

/etc/fstab

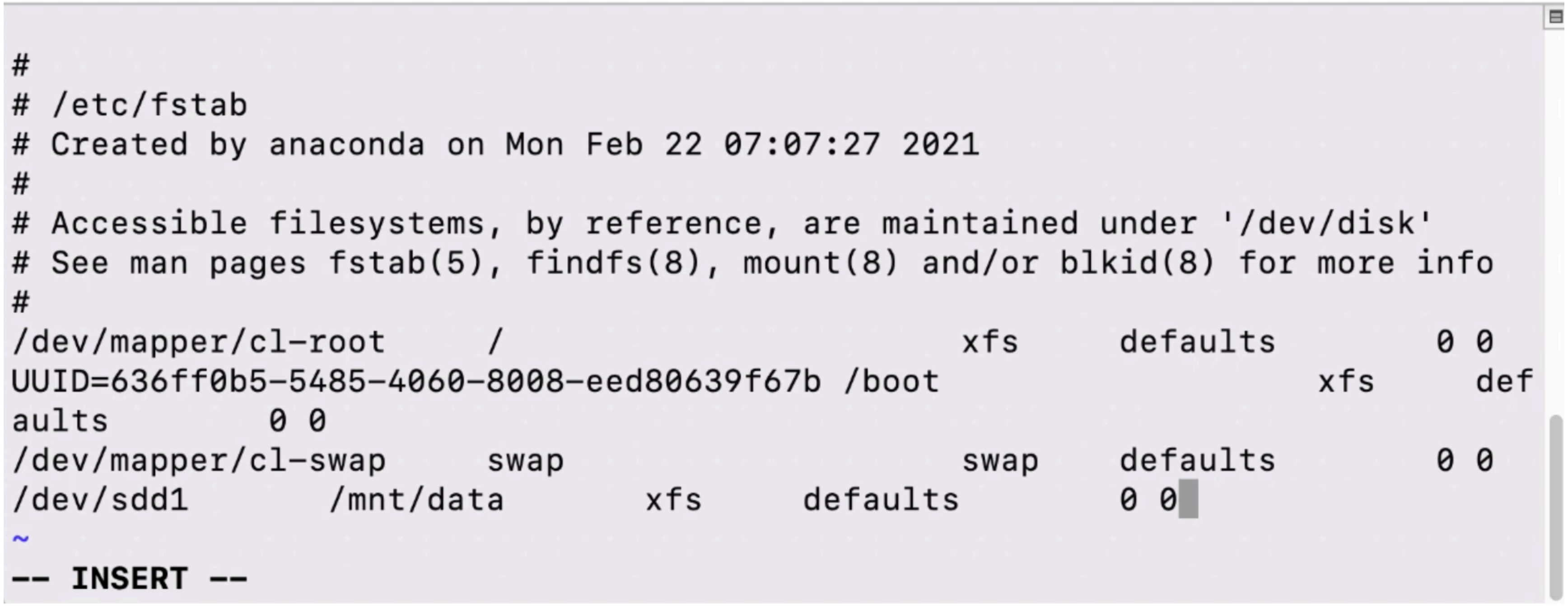


Filesystem mounts and unmounts

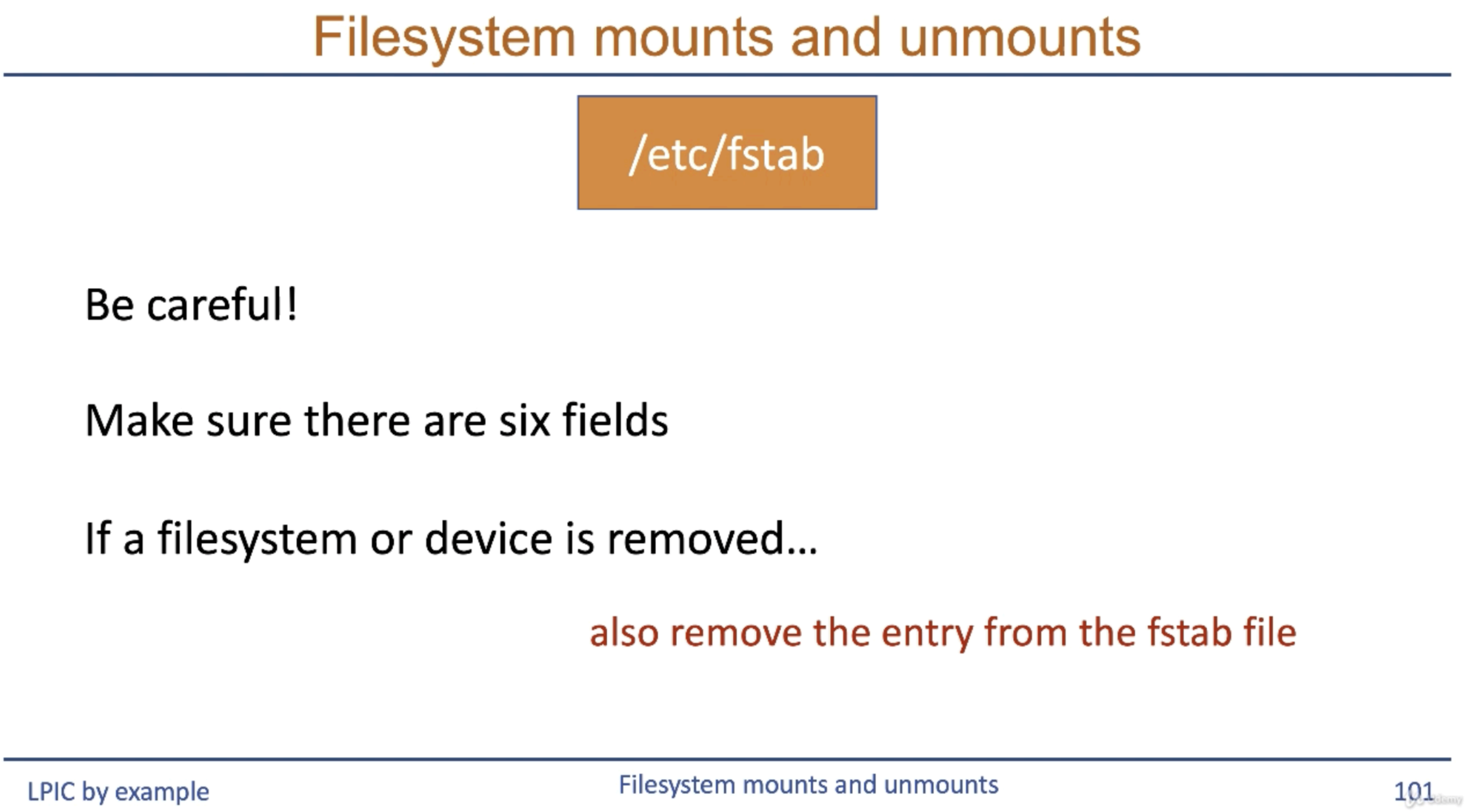
/etc/fstab six fields

device	mountpoint	fstype	options	dump	fsck
/dev/sdb1	/mnt/data	ext4	defaults	0	0
/dev/sdc3	/opt	xfs	ro	1	1

Filesystem mounts and unmounts



```
#
# /etc/fstab
# Created by anaconda on Mon Feb 22 07:07:27 2021
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/cl-root     /                       xfs     defaults        0 0
UUID=636ff0b5-5485-4060-8008-eed80639f67b /boot                   xfs     def
aults        0 0
/dev/mapper/cl-swap     swap                    swap    defaults        0 0
/dev/sdd1       /mnt/data       xfs     defaults        0 0
~
-- INSERT --
```



Filesystem mounts and unmounts

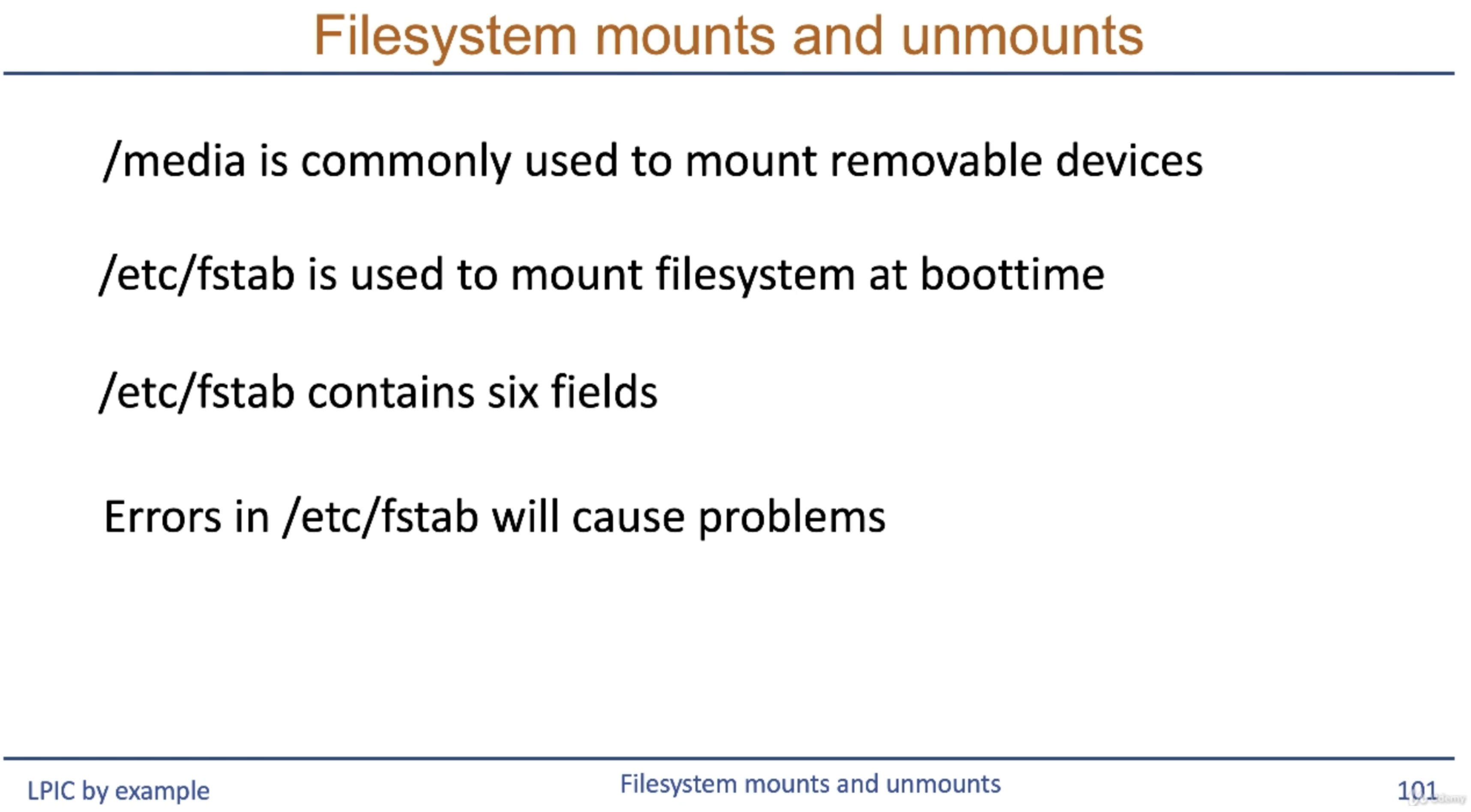
/etc/fstab

Be careful!

Make sure there are six fields

If a filesystem or device is removed…

also remove the entry from the fstab file



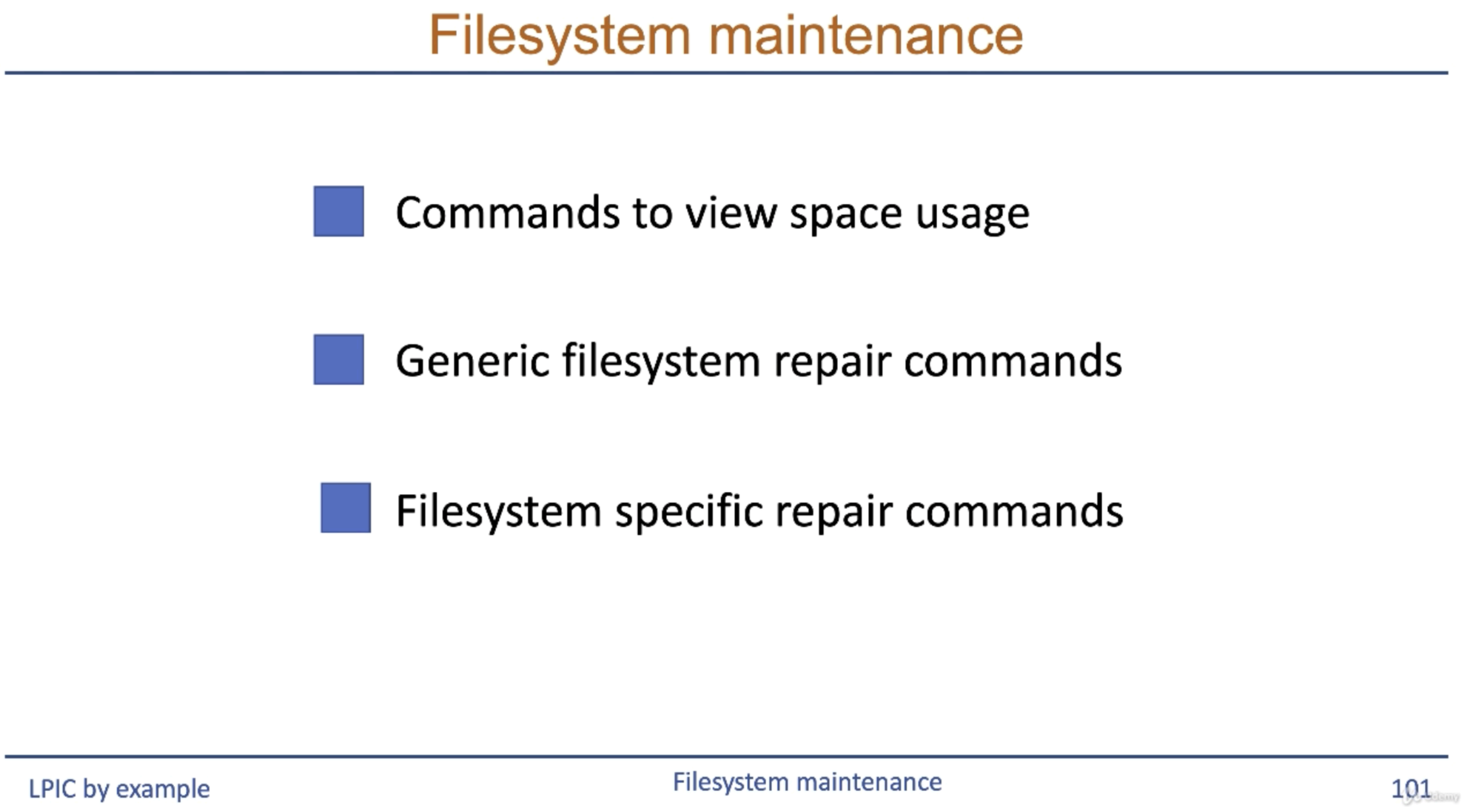
Filesystem mounts and unmounts

/media is commonly used to mount removable devices

/etc/fstab is used to mount filesystem at boottime

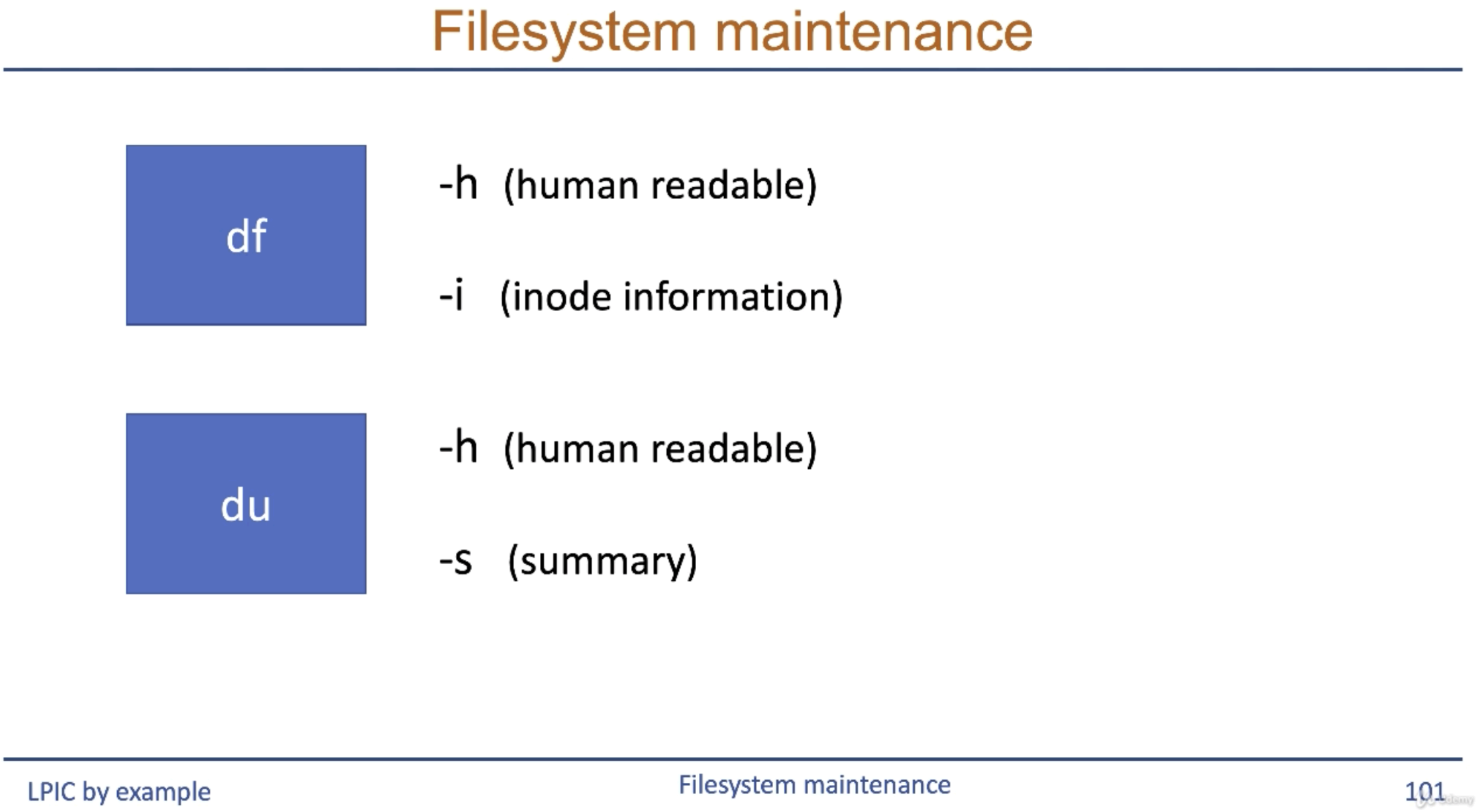
/etc/fstab contains six fields

Errors in /etc/fstab will cause problems



Filesystem maintenance

- Commands to view space usage
- Generic filesystem repair commands
- Filesystem specific repair commands



Filesystem maintenance

df

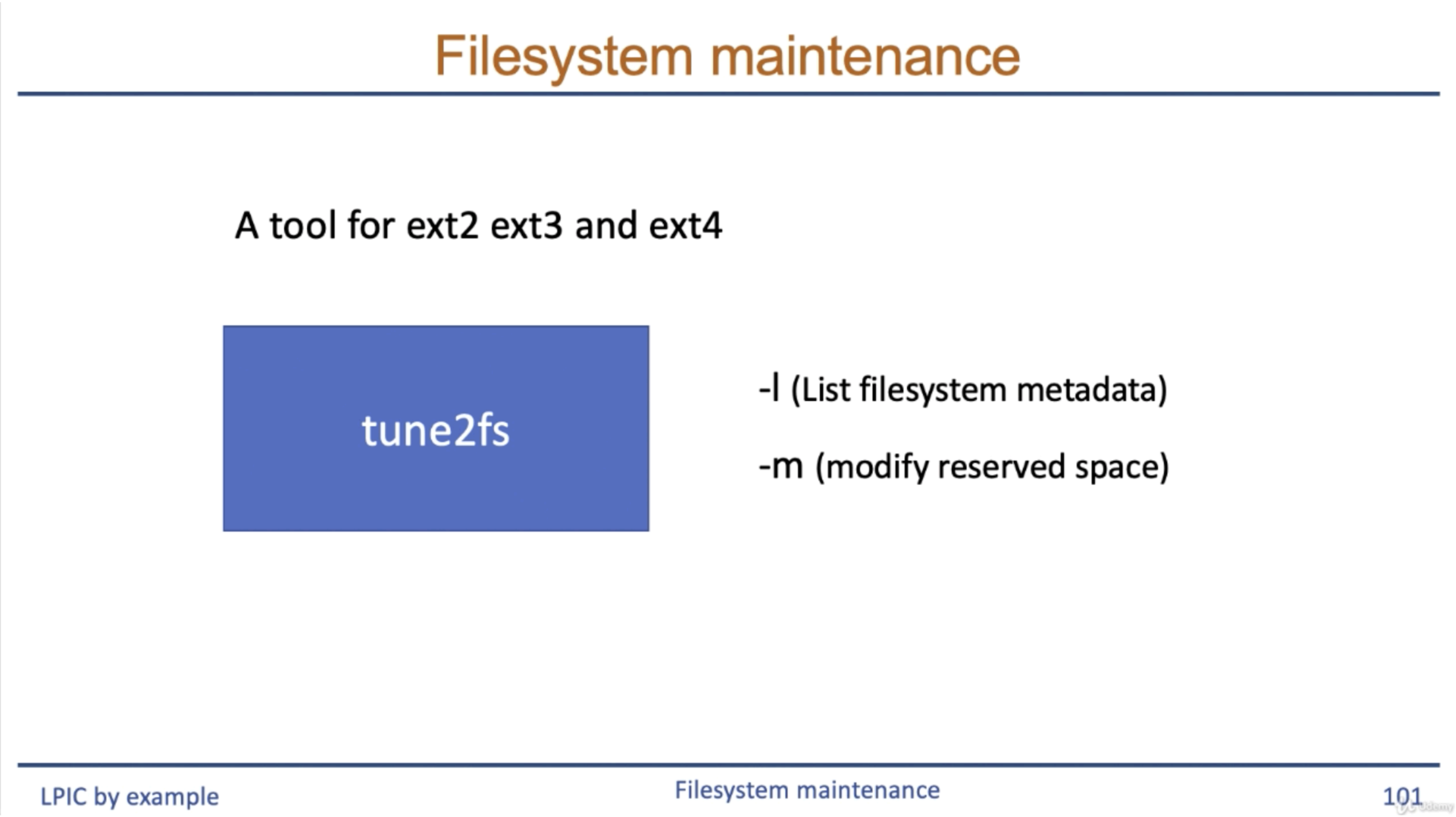
-h (human readable)

-i (inode information)

du

-h (human readable)

-s (summary)



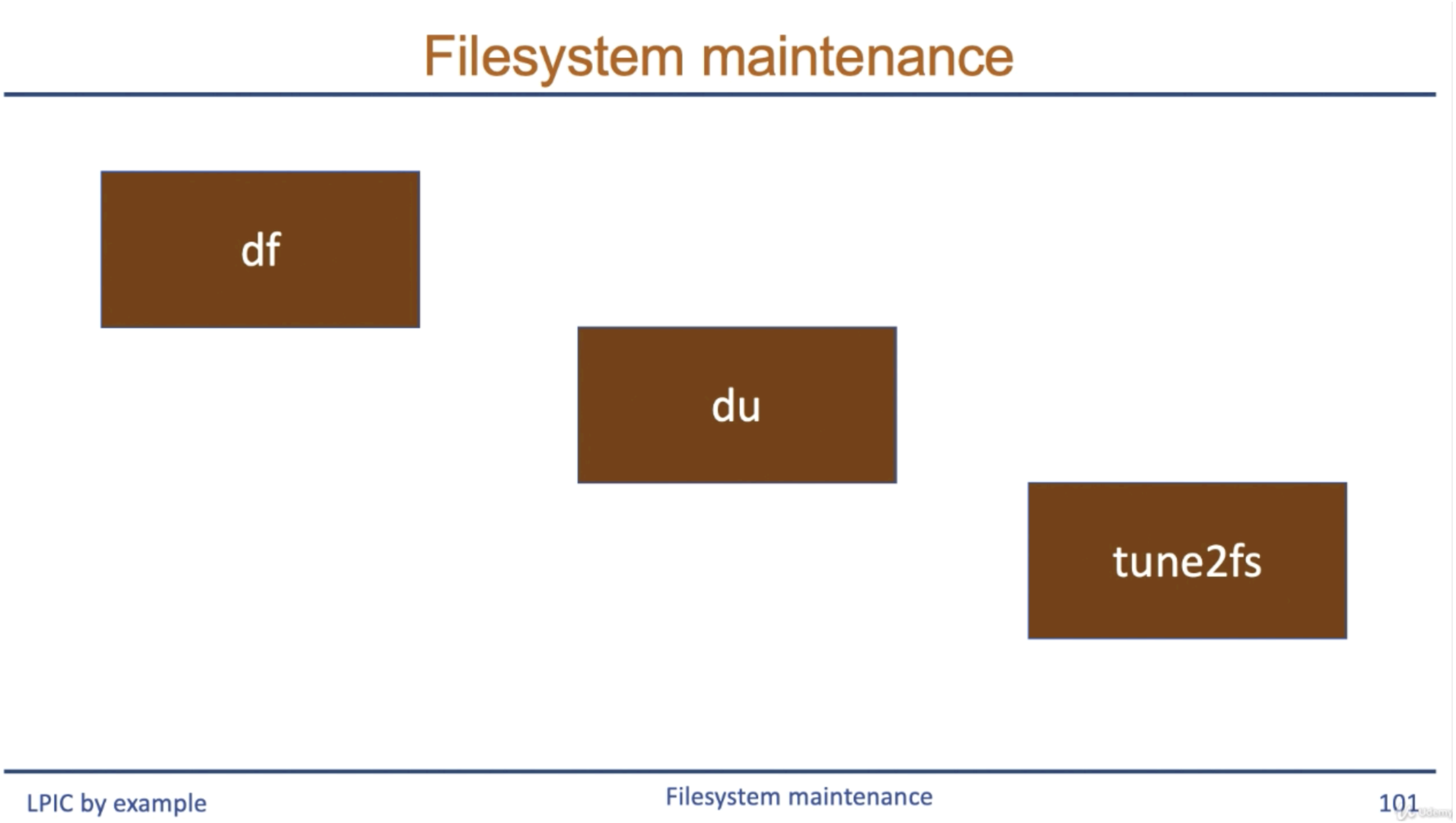
Filesystem maintenance

A tool for ext2 ext3 and ext4

tune2fs

-l (List filesystem metadata)

-m (modify reserved space)

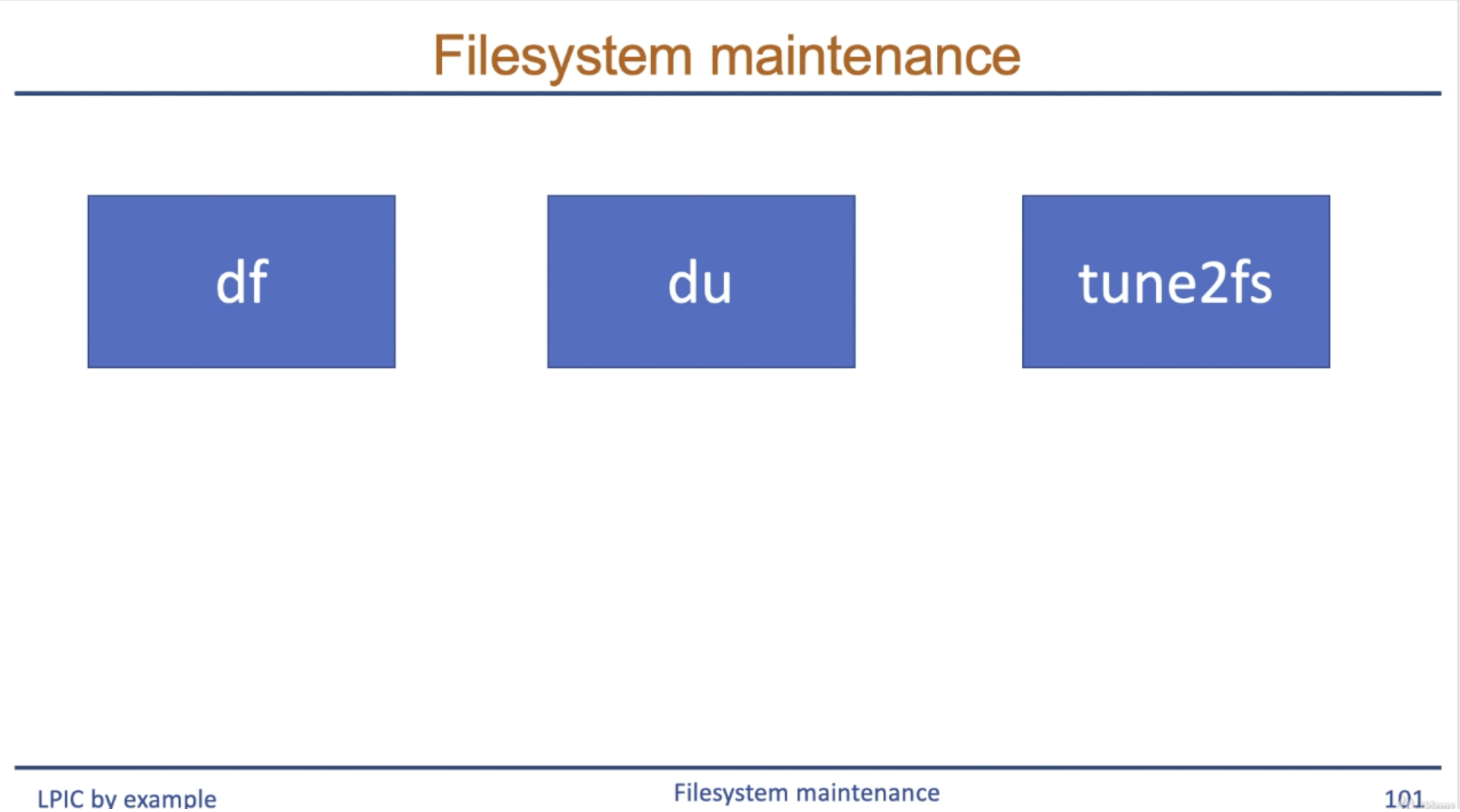


Filesystem maintenance

df

du

tune2fs

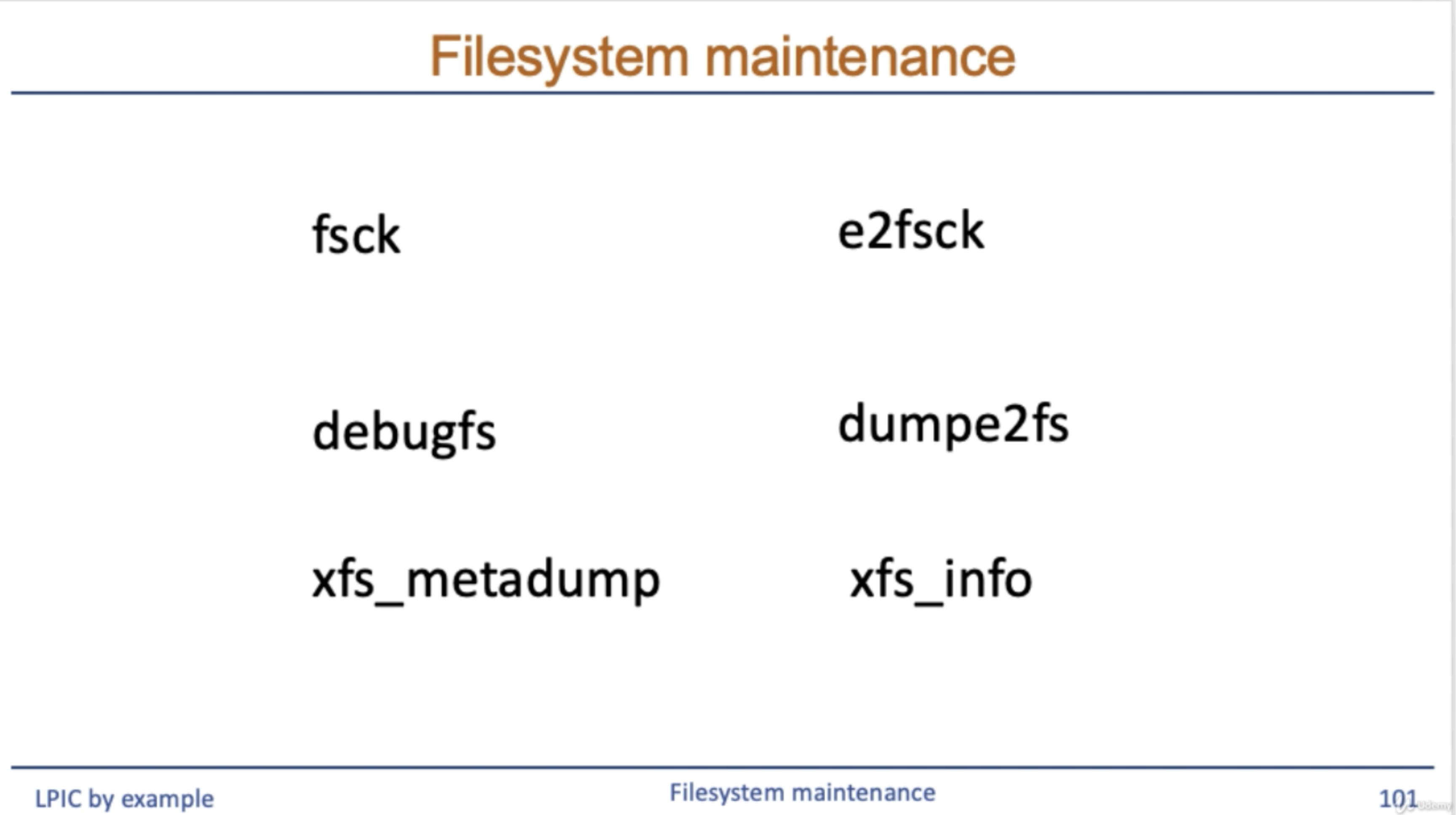


Filesystem maintenance

df

du

tune2fs



Filesystem maintenance

fsck

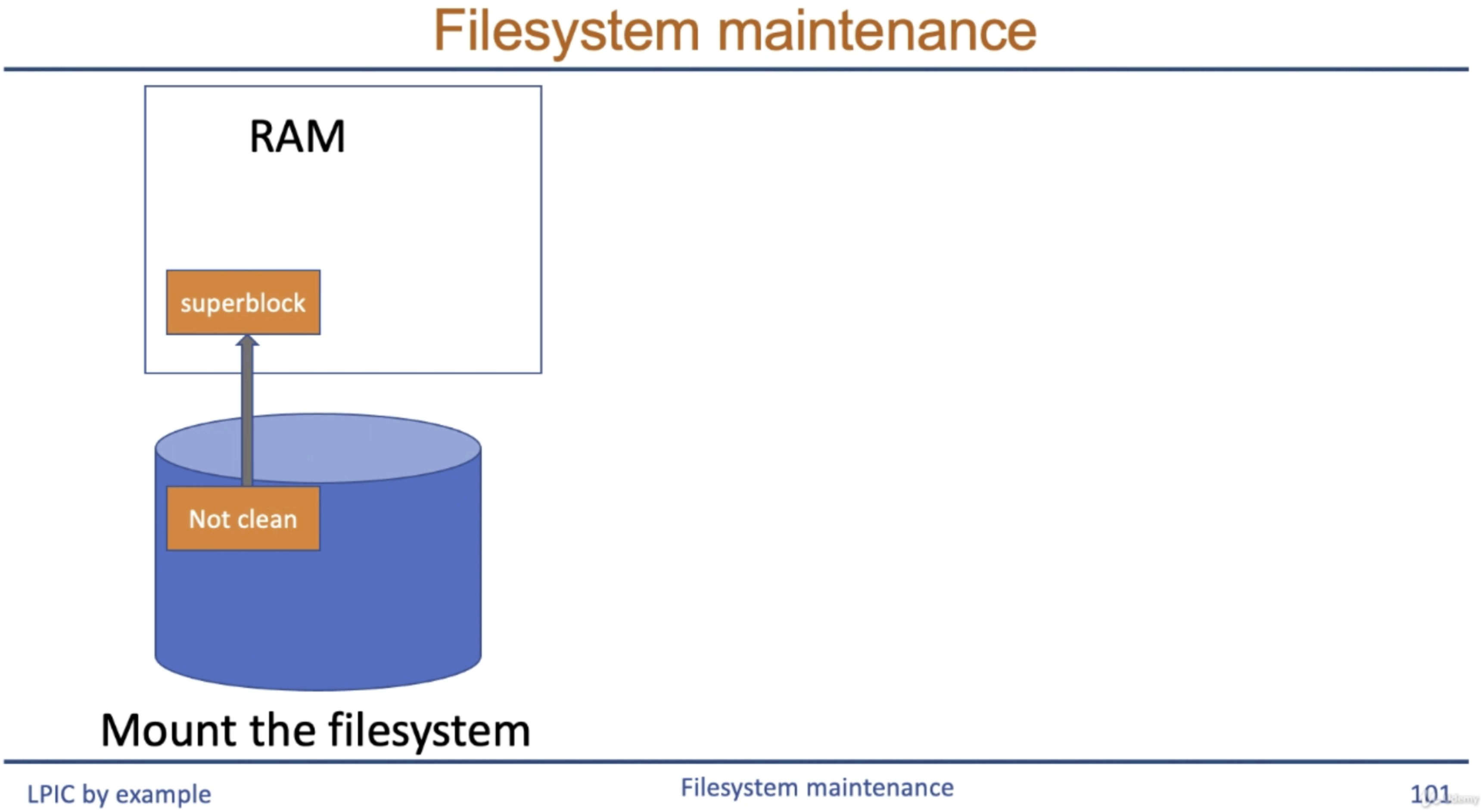
e2fsck

debugfs

dumpe2fs

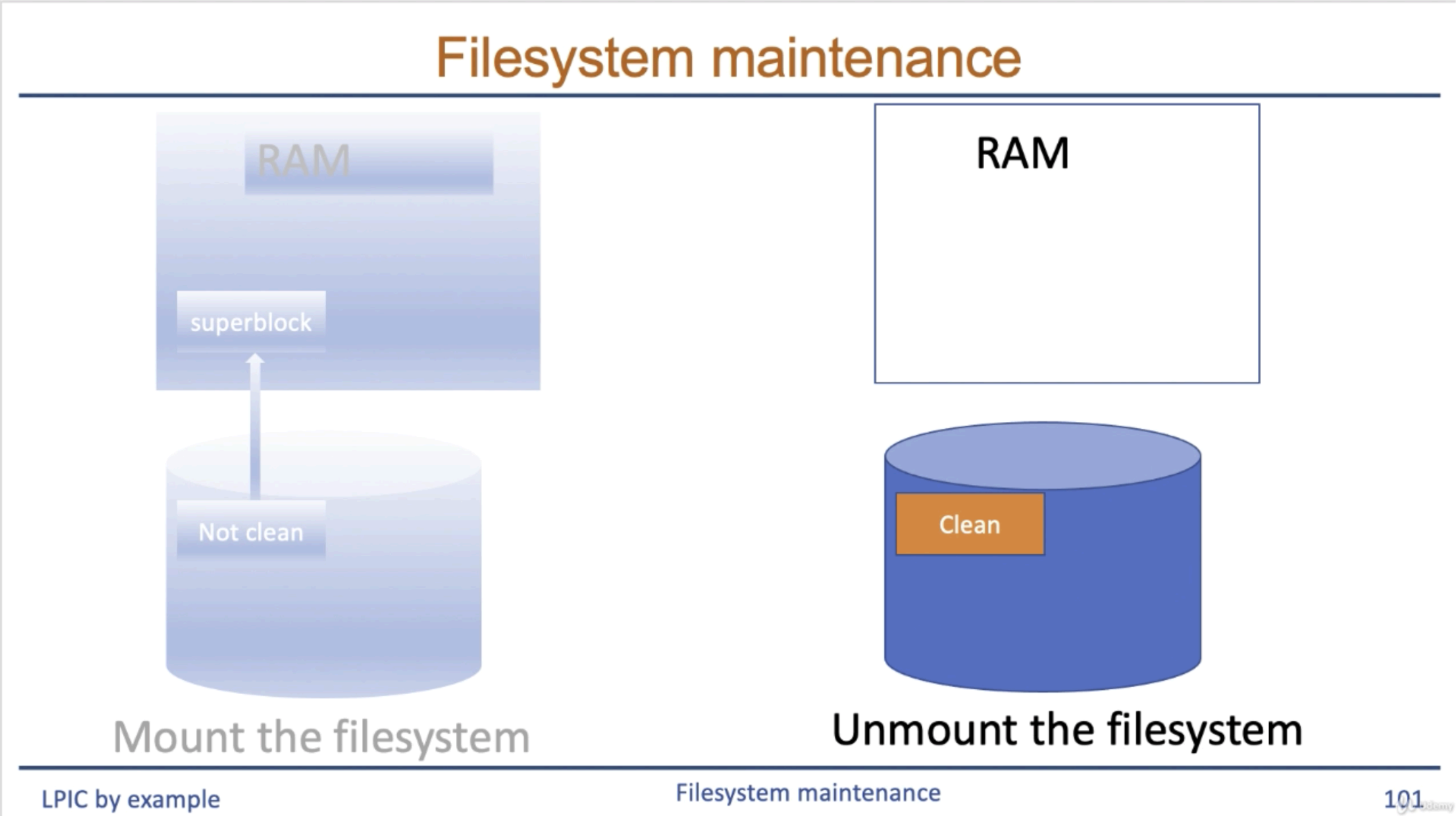
xfs_metadump

xfs_info



Filesystem maintenance

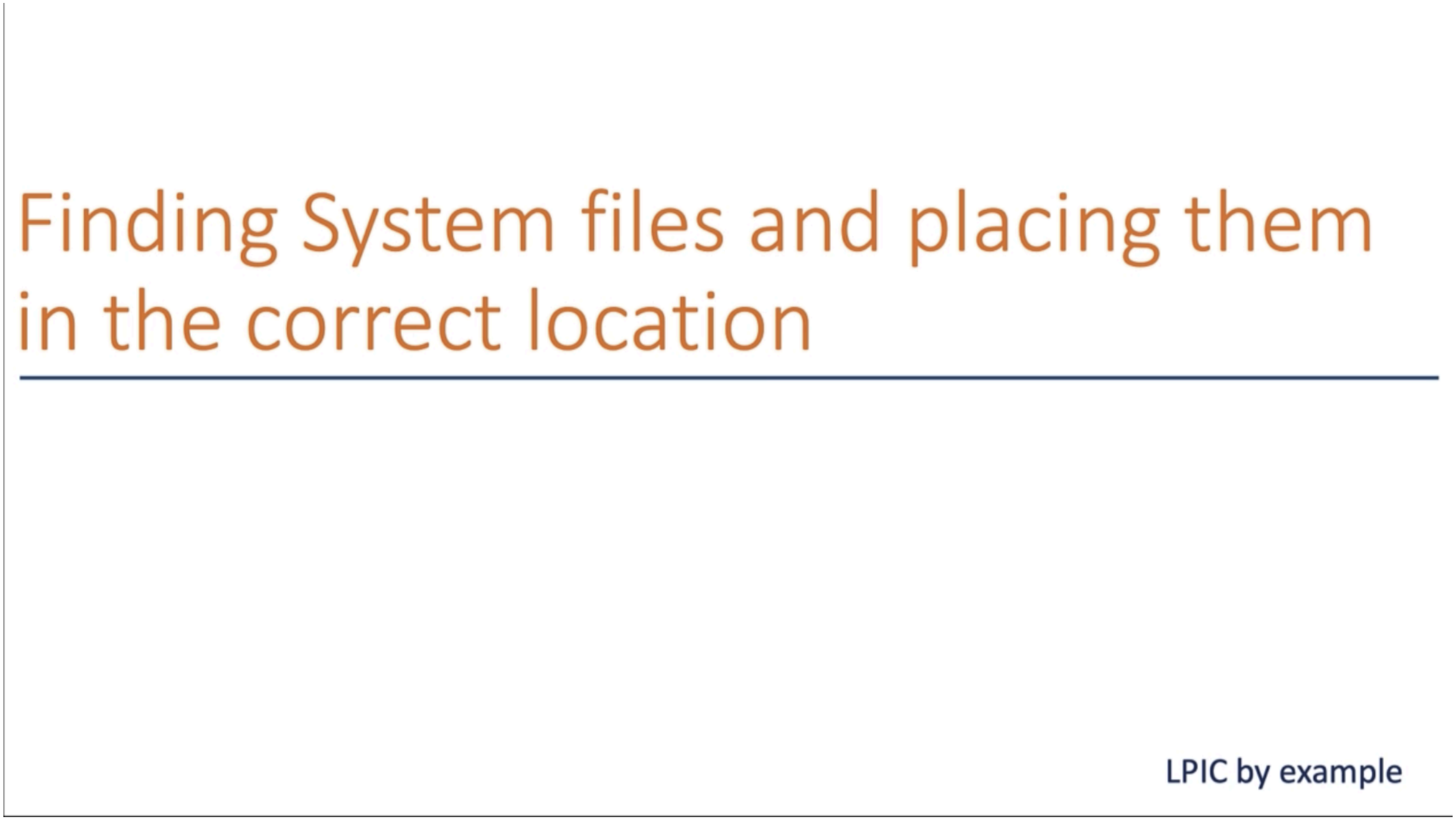
Mount the filesystem



Filesystem maintenance

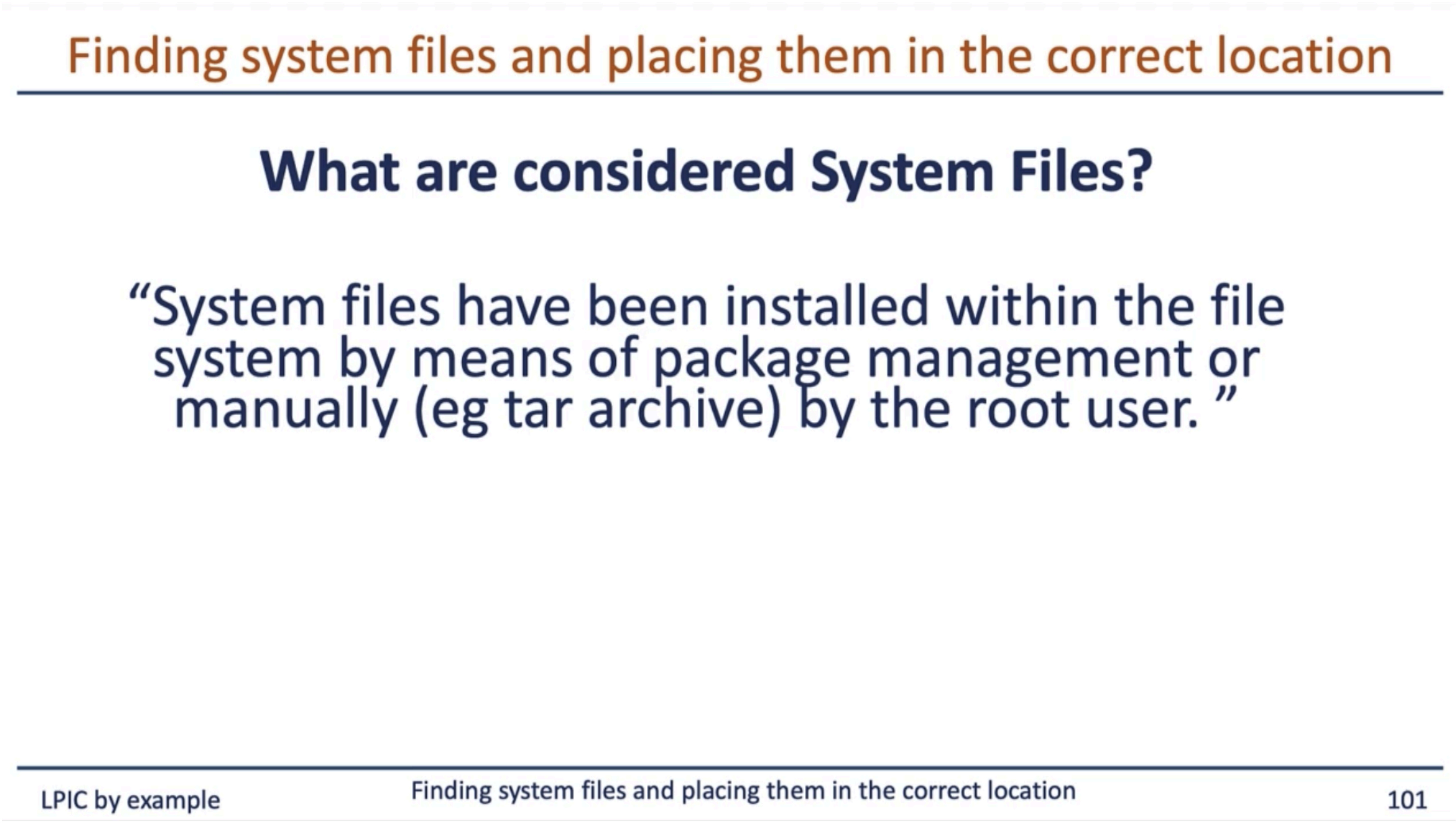
Mount the filesystem

Unmount the filesystem



Finding System files and placing them in the correct location

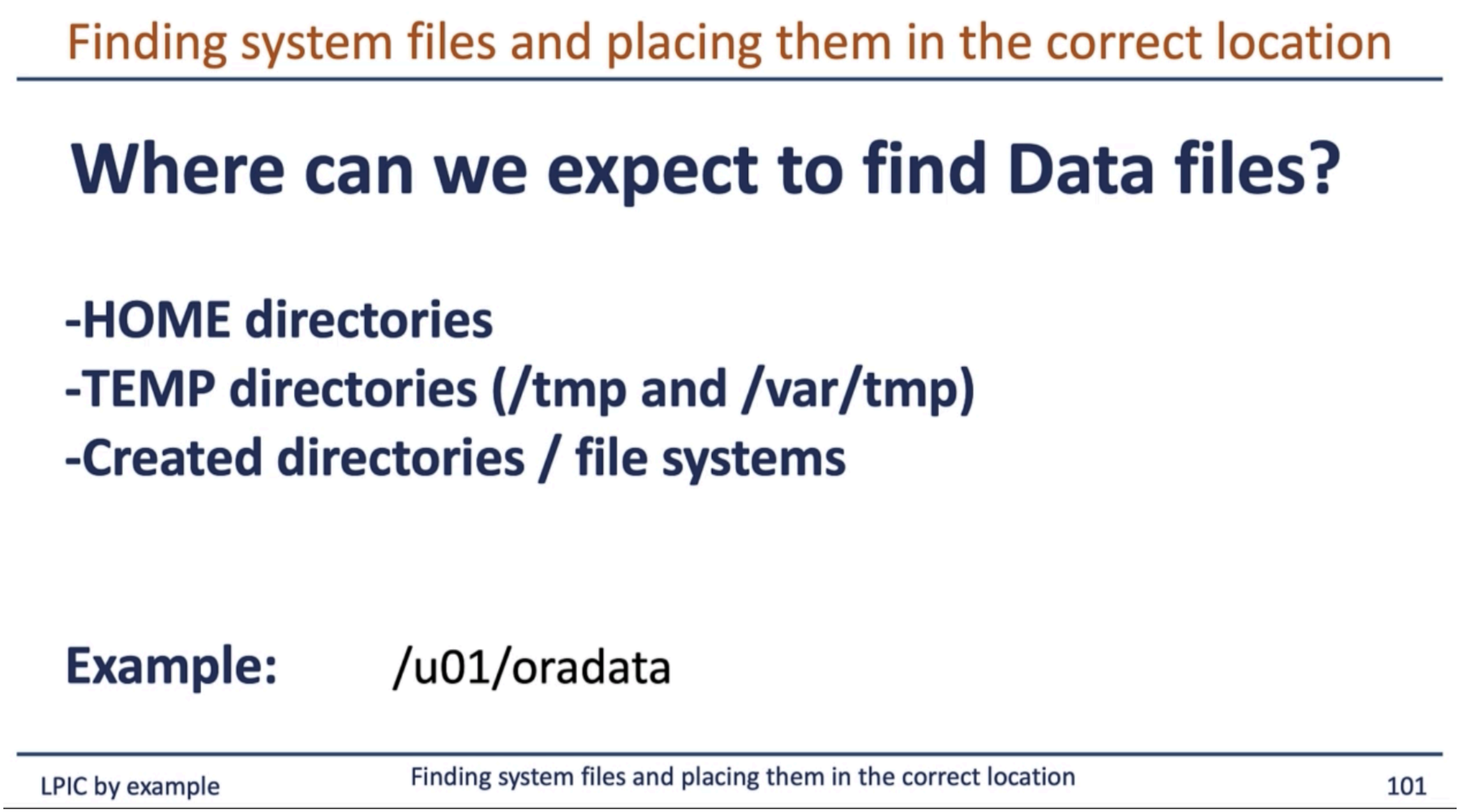
LPIC by example



Finding system files and placing them in the correct location

What are considered System Files?

“System files have been installed within the file system by means of package management or manually (eg tar archive) by the root user. ”

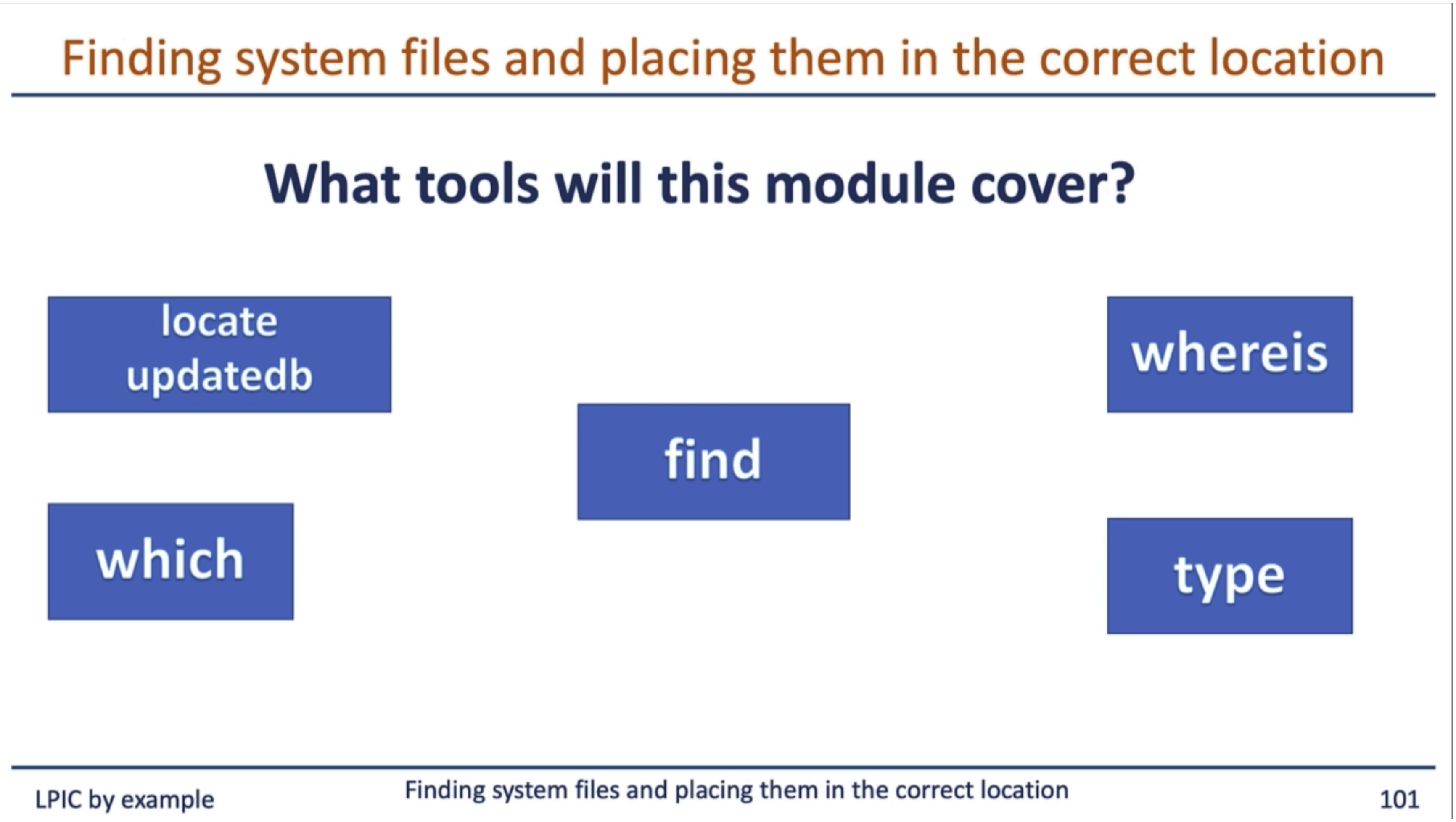


Finding system files and placing them in the correct location

Where can we expect to find Data files?

-HOME directories
-TEMP directories (/tmp and /var/tmp)
-Created directories / file systems

Example: /u01/oradata



Finding system files and placing them in the correct location

What tools will this module cover?

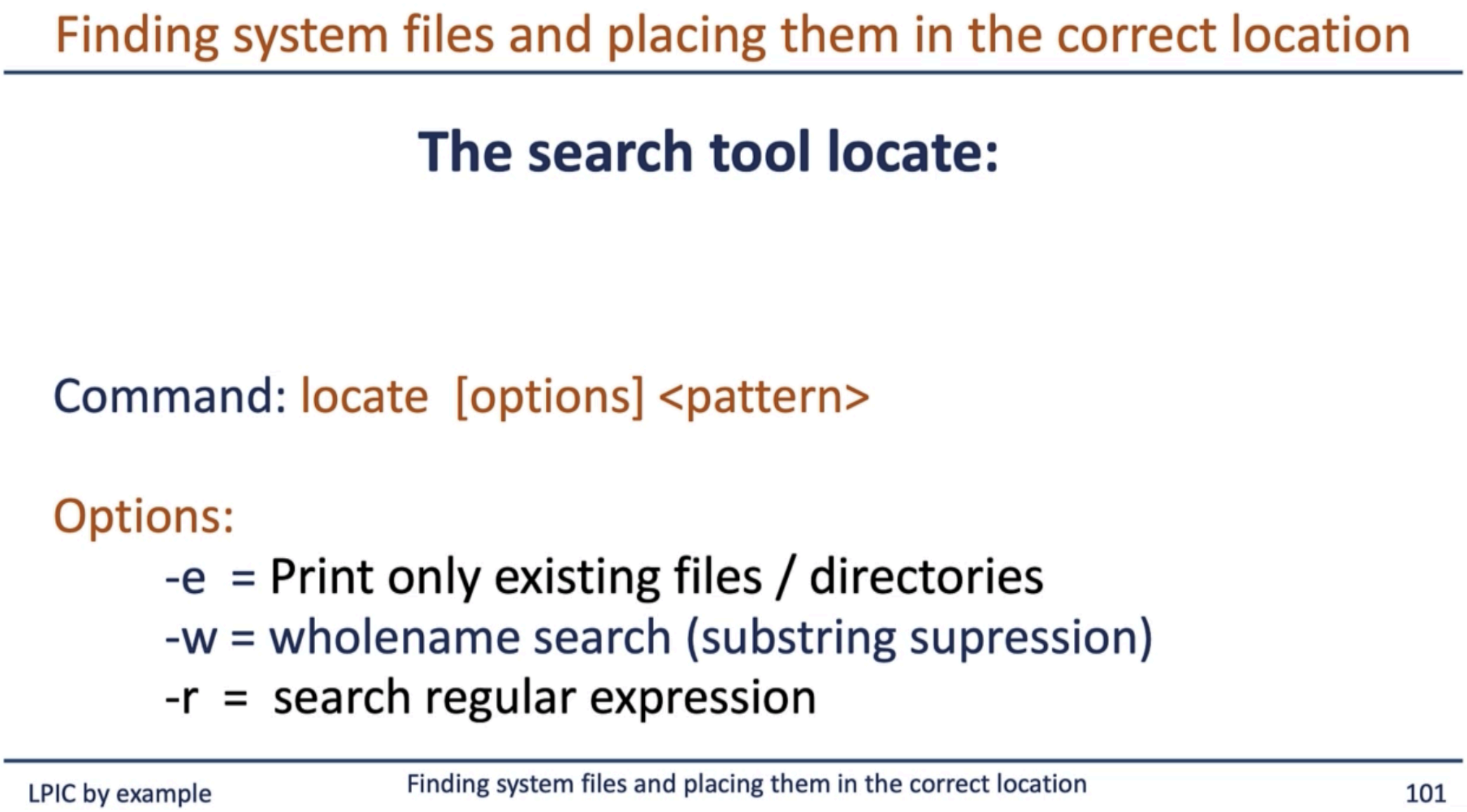
locate
updatedb

which

find

whereis

type



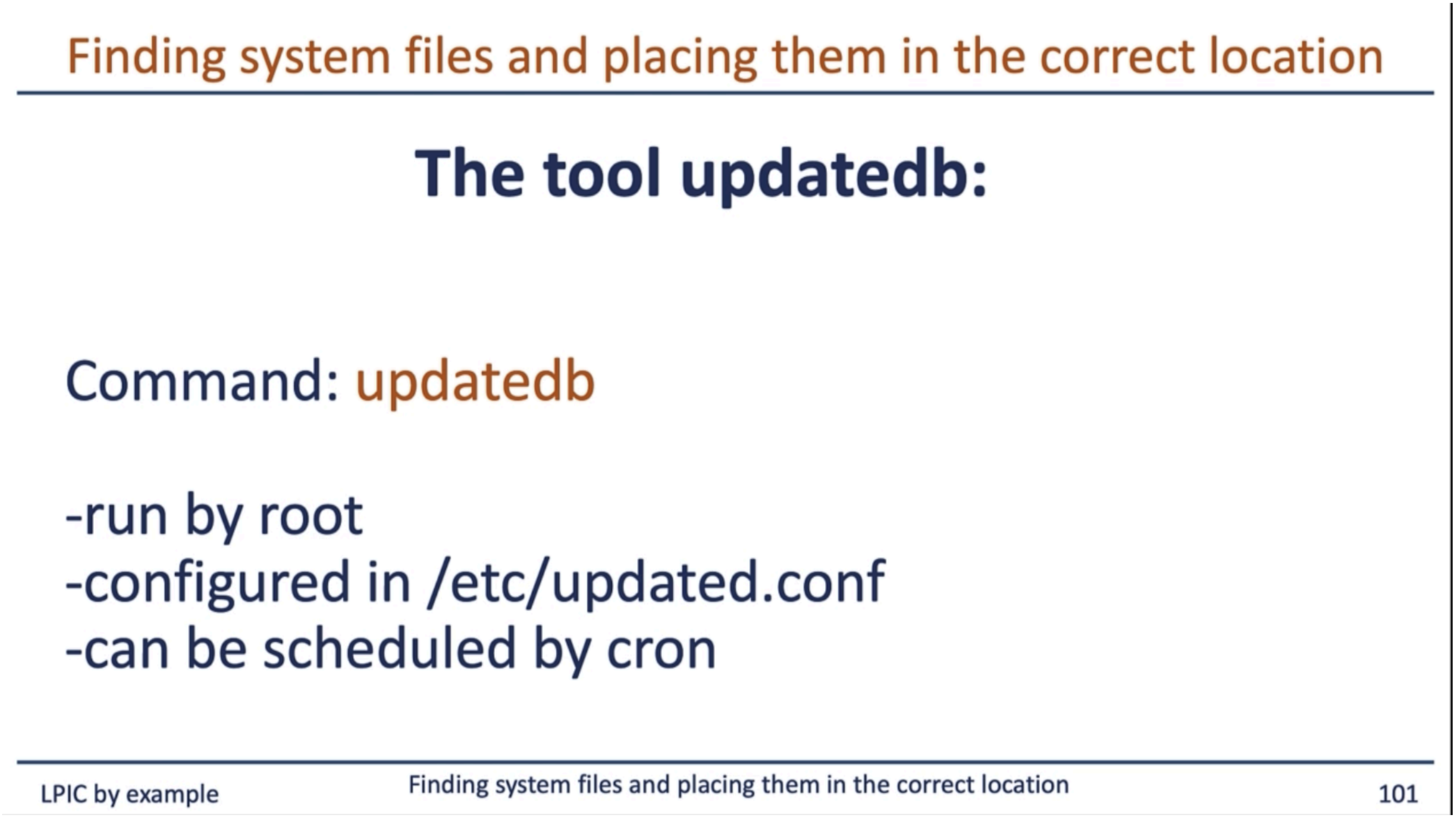
Finding system files and placing them in the correct location

The search tool locate:

Command: locate [options] <pattern>

Options:

-e = Print only existing files / directories
-w = wholename search (substring supression)
-r = search regular expression

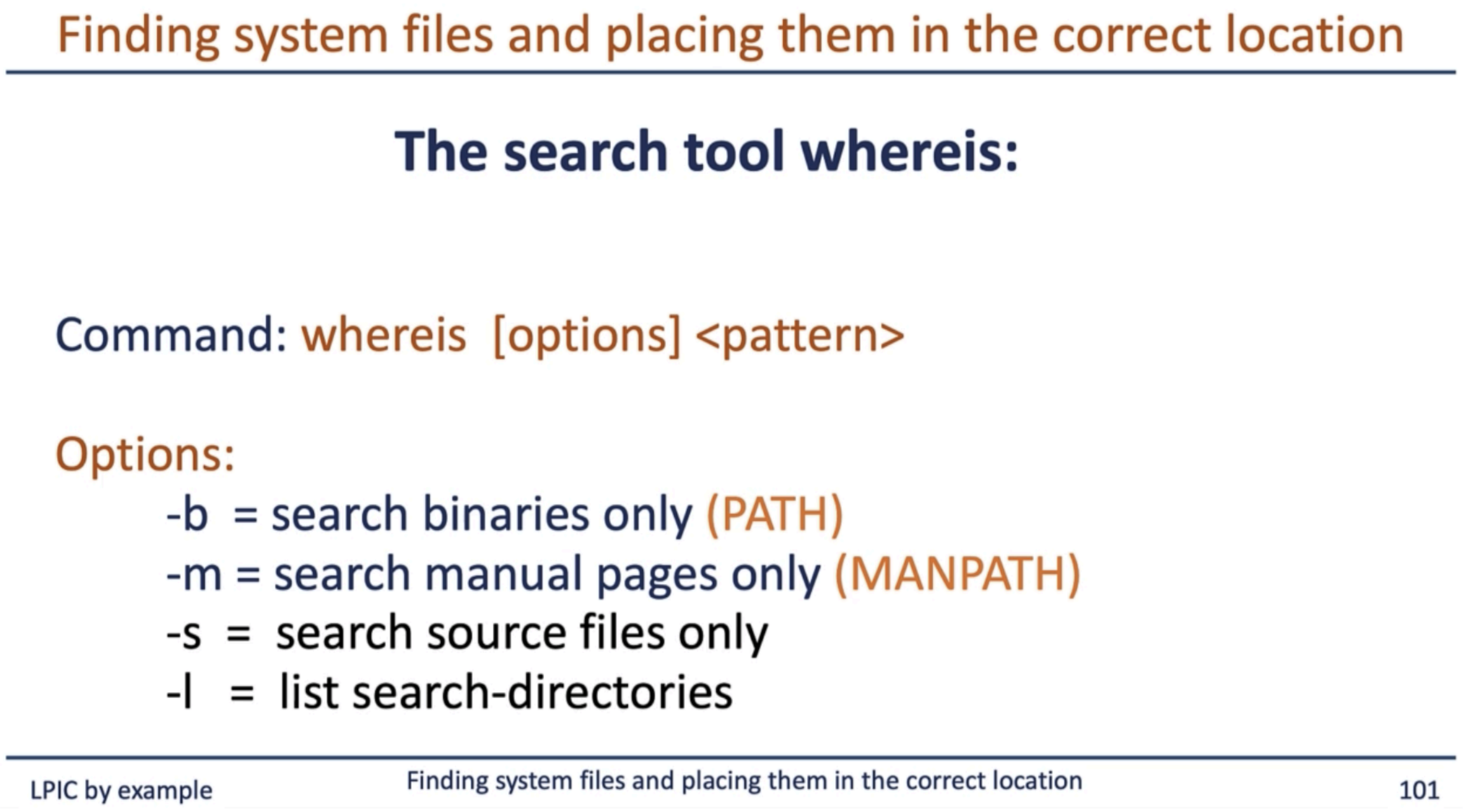


Finding system files and placing them in the correct location

The tool updatedb:

Command: updatedb

-run by root
-configured in /etc/updated.conf
-can be scheduled by cron



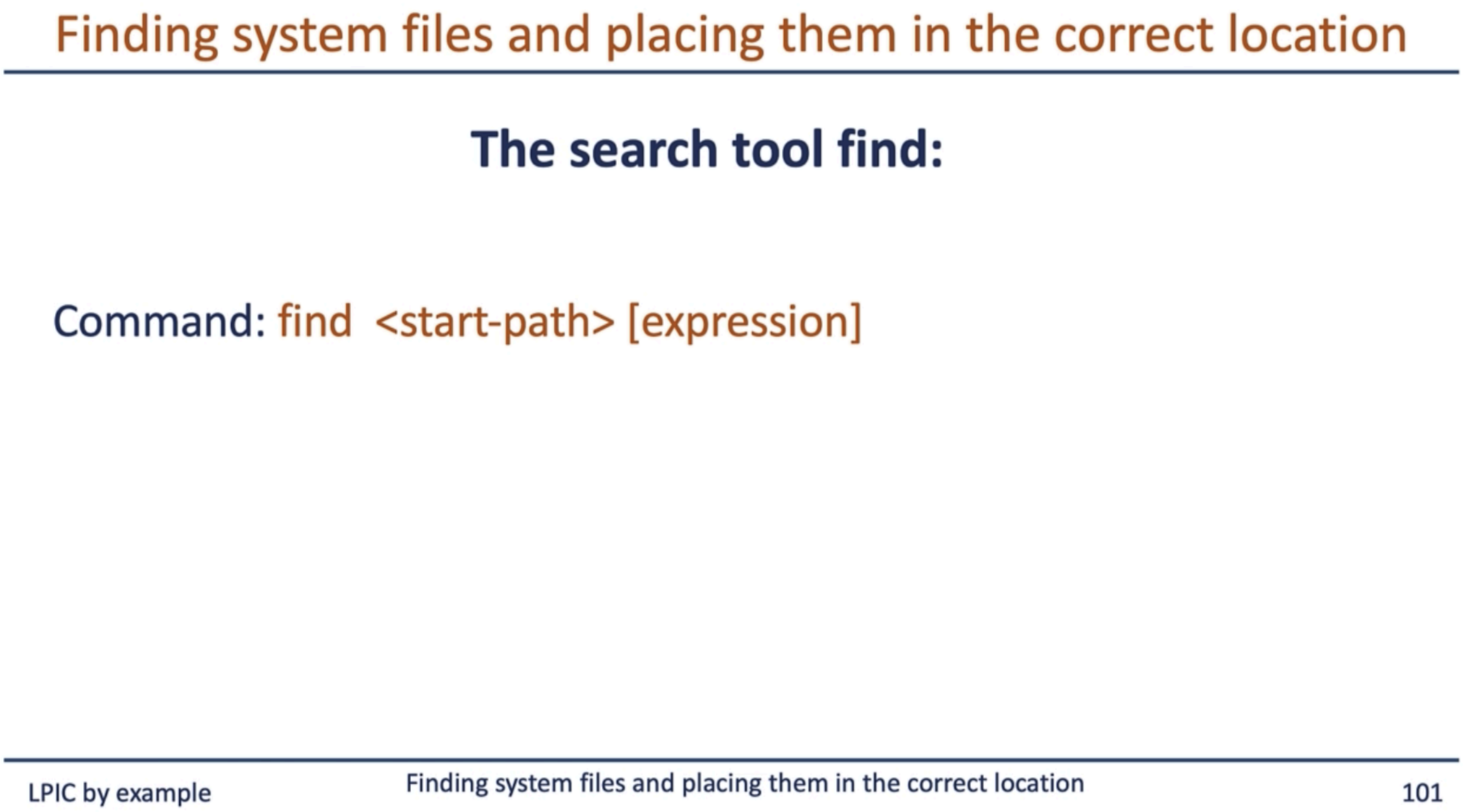
Finding system files and placing them in the correct location

The search tool whereis:

Command: whereis [options] <pattern>

Options:

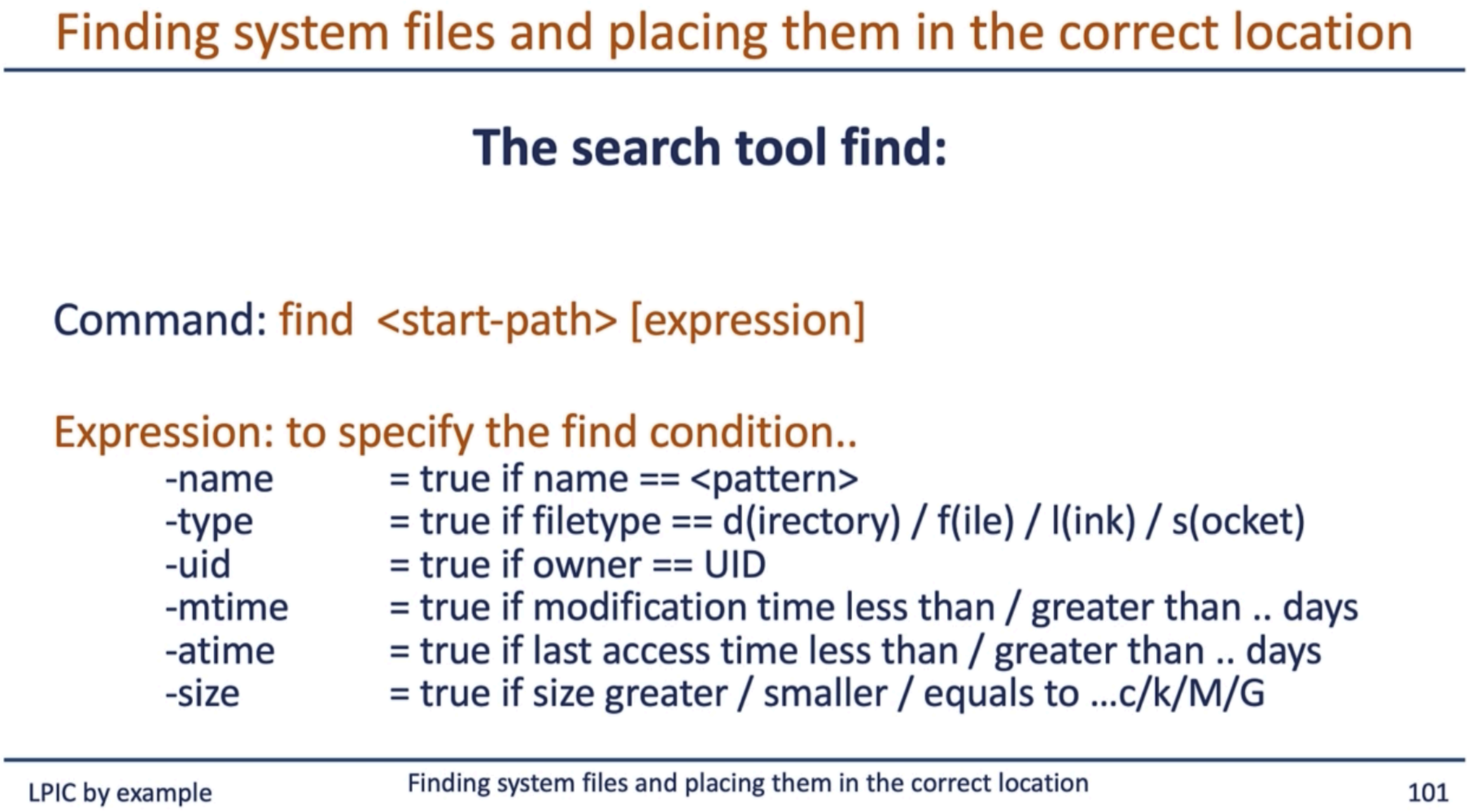
- -b = search binaries only (PATH)
- -m = search manual pages only (MANPATH)
- -s = search source files only
- -l   = list search-directories



Finding system files and placing them in the correct location

The search tool find:

Command: `find <start-path> [expression]`



Finding system files and placing them in the correct location

The search tool find:

Command: find <start-path> [expression]

Expression: to specify the find condition..

-name = true if name == <pattern>
-type = true if filetype == d(irectory) / f(ile) / l(ink) / s(ocket)
-uid = true if owner == UID
-mtime = true if modification time less than / greater than .. days
-atime = true if last access time less than / greater than .. days
-size = true if size greater / smaller / equals to …c/k/M/G