Introduction to Shell-based Data Processing

Lecture 2: Example (auto creating accounts)
Practice 2: Logging in and changing password

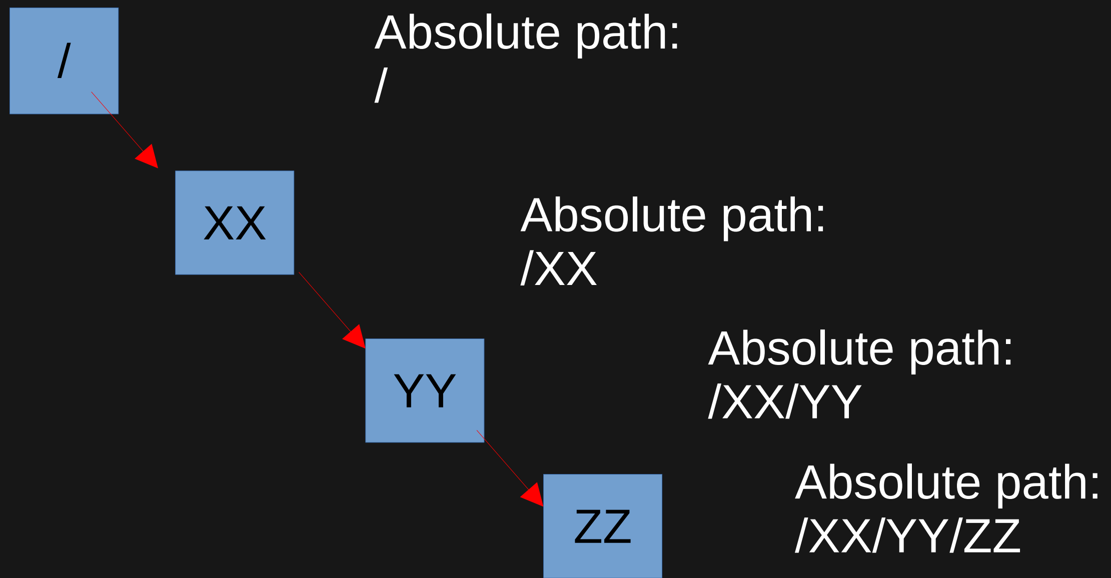
Richard Veale
Kyoto University, Fall 2024

Review from Lecture 1

- 1) Shell is a mechanism for interacting with a computer
  - Graphical Shell versus Textual Shell
- 2) Open shell on windows:
  - Run “command prompt” program
- 3) Connecting to a remote machine:
  - Used “secure shell” (ssh) to connect to a remote server over network (internet) (ssh XXX@YYYY)
- 4) Basic BASH (unix shell) commands:
  - View current directory (pwd), create directory (mkdir), change current directory (cd), echo text (echo), redirect output to file (> or >>), create a file (touch)

Clarifications from Lecture 1

- 1) Directory hierarchy format:
  - In unix filesystems “/” is the **root** directory (base of all paths to files in system)
  - “/” separates hierarchy levels
  - /XX/YY/ZZ means:
    - File or directory ZZ inside directory YY inside directory XX inside the root.



Absolute path:
/

Absolute path:
/XX

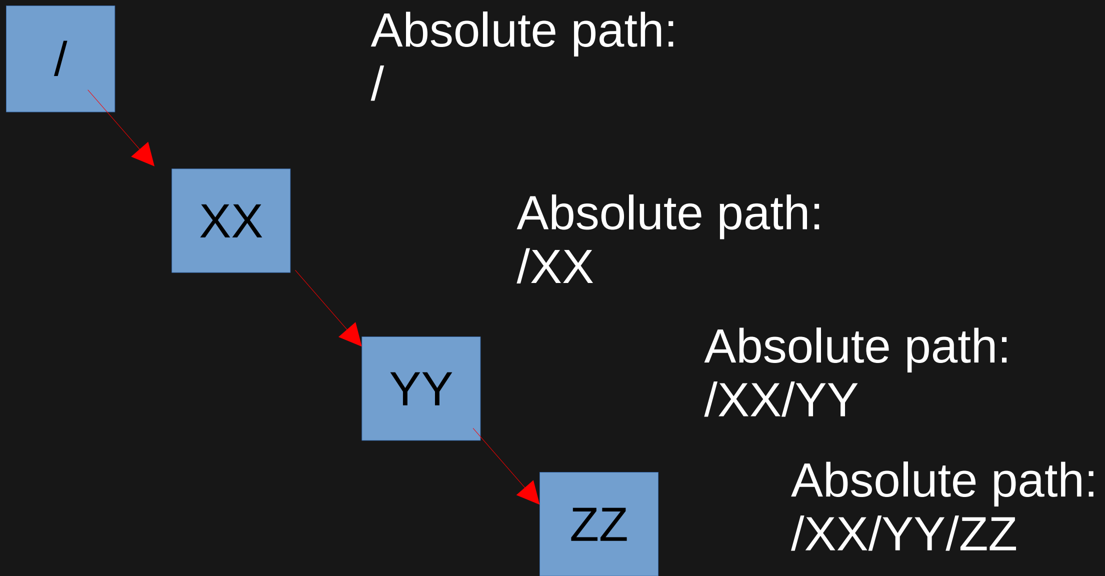
Absolute path:
/XX/YY

Absolute path:
/XX/YY/ZZ

Clarifications from Lecture 1

- “Absolute” path means path from root
- Path means the pathway to get to a file/directory
  - (On windows: C://Users/XX for example)
  - (On unix-like: /Users/XX)

Absolute Path will always start with “/”



Absolute path:
/

Absolute path:
/XX

Absolute path:
/XX/YY

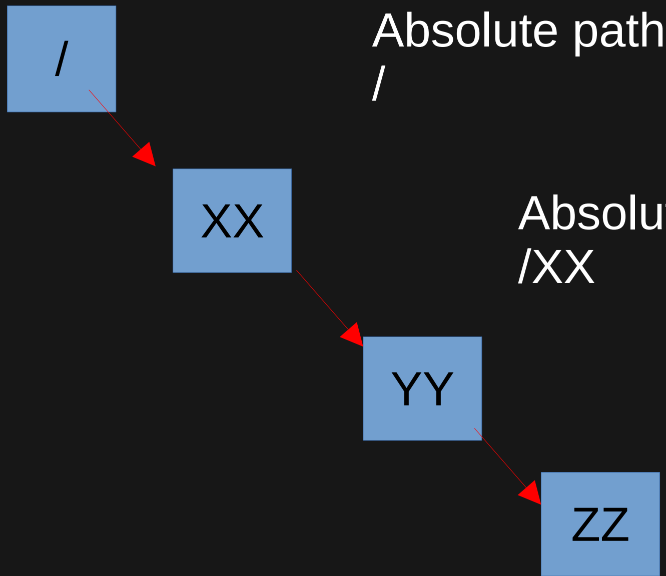
Absolute path:
/XX/YY/ZZ

Clarifications from Lecture 1

Absolute Path will always start with “/”

If it is not absolute, it will not start with “/”

(Example: XX/YY is a *relative* path)



Absolute path:
/

Absolute path:
/XX

Absolute path:
/XX/YY

Absolute path:
/XX/YY/ZZ

Clarifications from Lecture 1

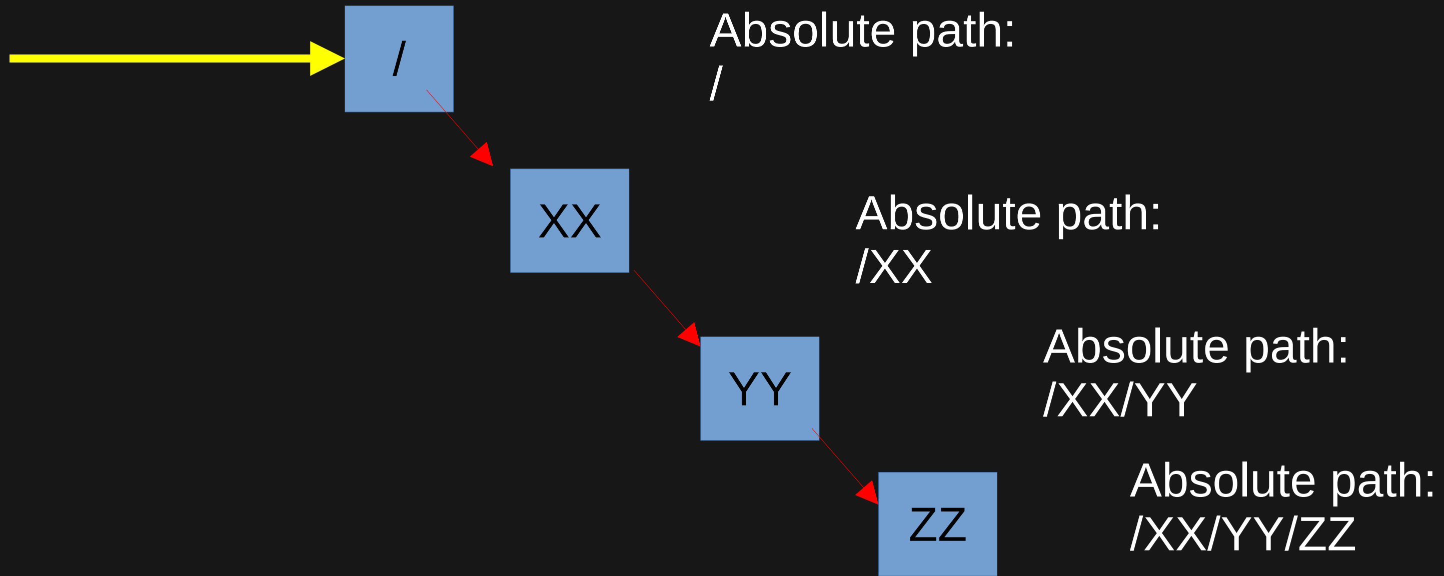
<u>Absolute Path</u> will always start with “/”

If it is not absolute, it will not start with “/”

(Example: XX/YY is a *relative* path)

Example:
You are here

```
$ cd /
$ pwd
/
```



Absolute path:
/

Absolute path:
/XX

Absolute path:
/XX/YY

Absolute path:
/XX/YY/ZZ

Clarifications from Lecture 1

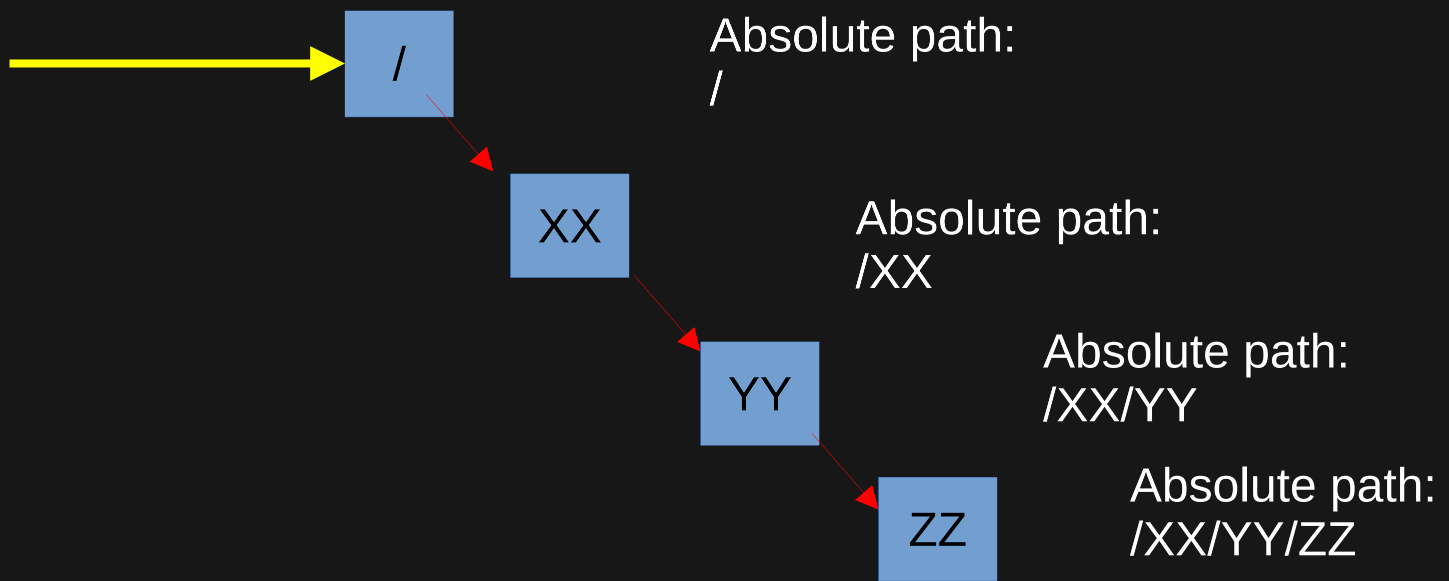
Absolute Path will always start with “/”

If it is not absolute, it will not start with “/”

(Example: XX/YY is a *relative* path)

Example:
You are here

```
$ ls
XX
<some other
things>
```



Absolute path:
/

Absolute path:
/XX

Absolute path:
/XX/YY

Absolute path:
/XX/YY/ZZ

Clarifications from Lecture 1

If you are in /XX, typing “ls YY” will list the contents of directory YY *relative to current directory*

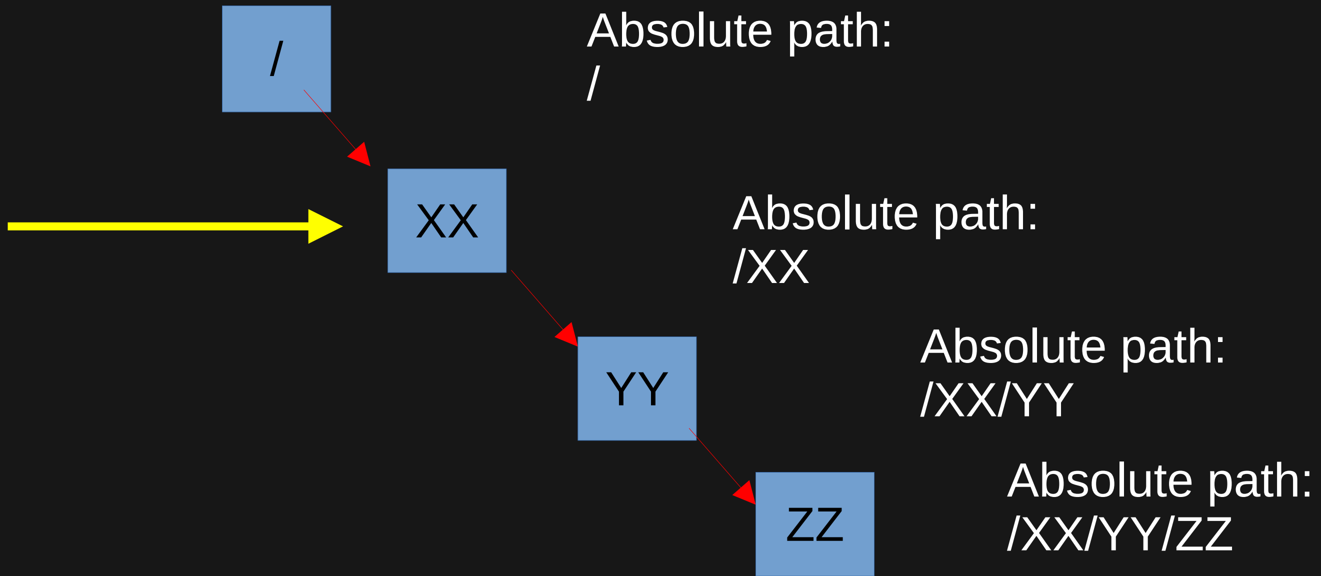
It will print absolute /XX/YY, since you are in /XX

Example:
You are here

```
$cd /XX

$pwd
/XX

$ ls
YY
```

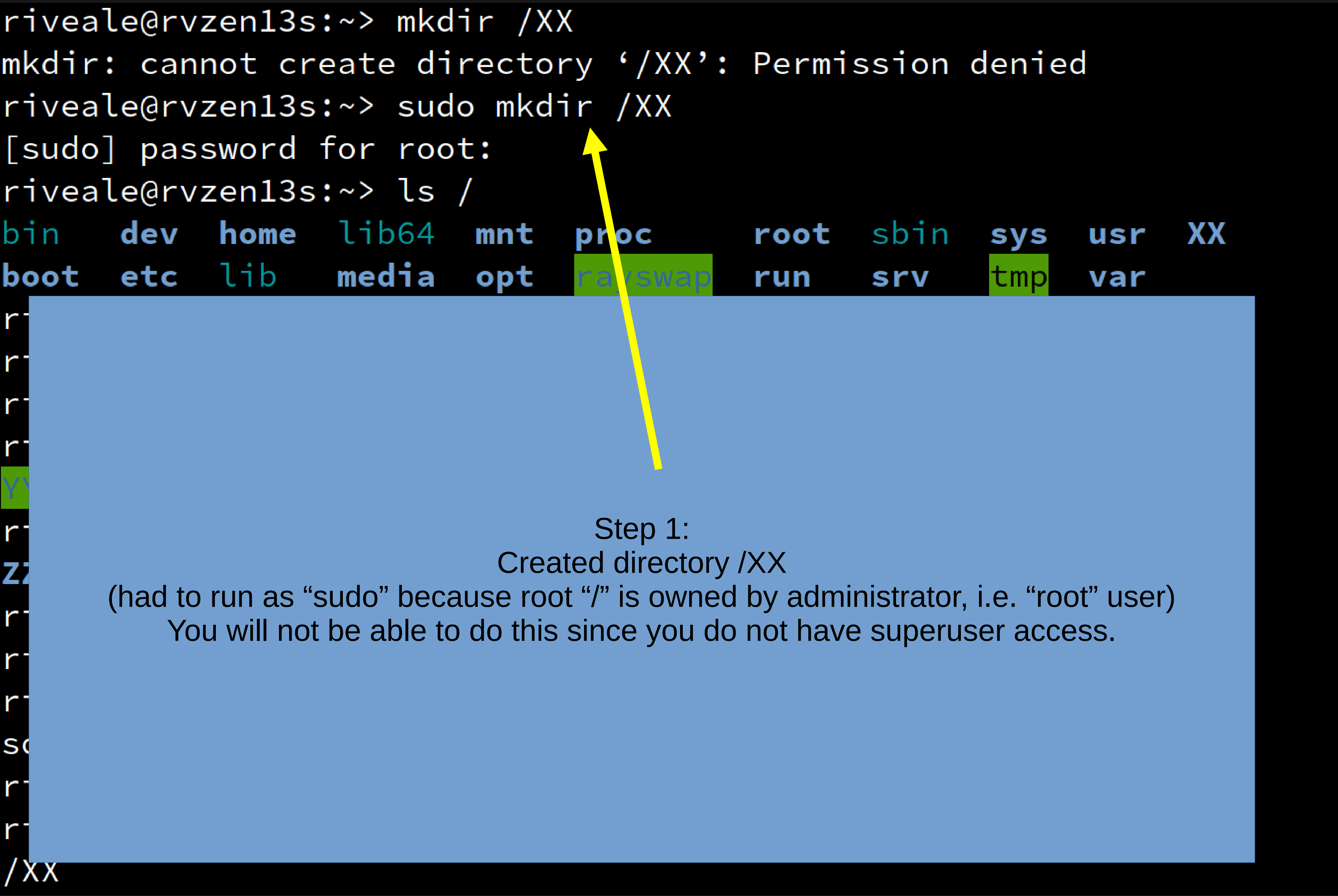


Absolute path:
/

Absolute path:
/XX

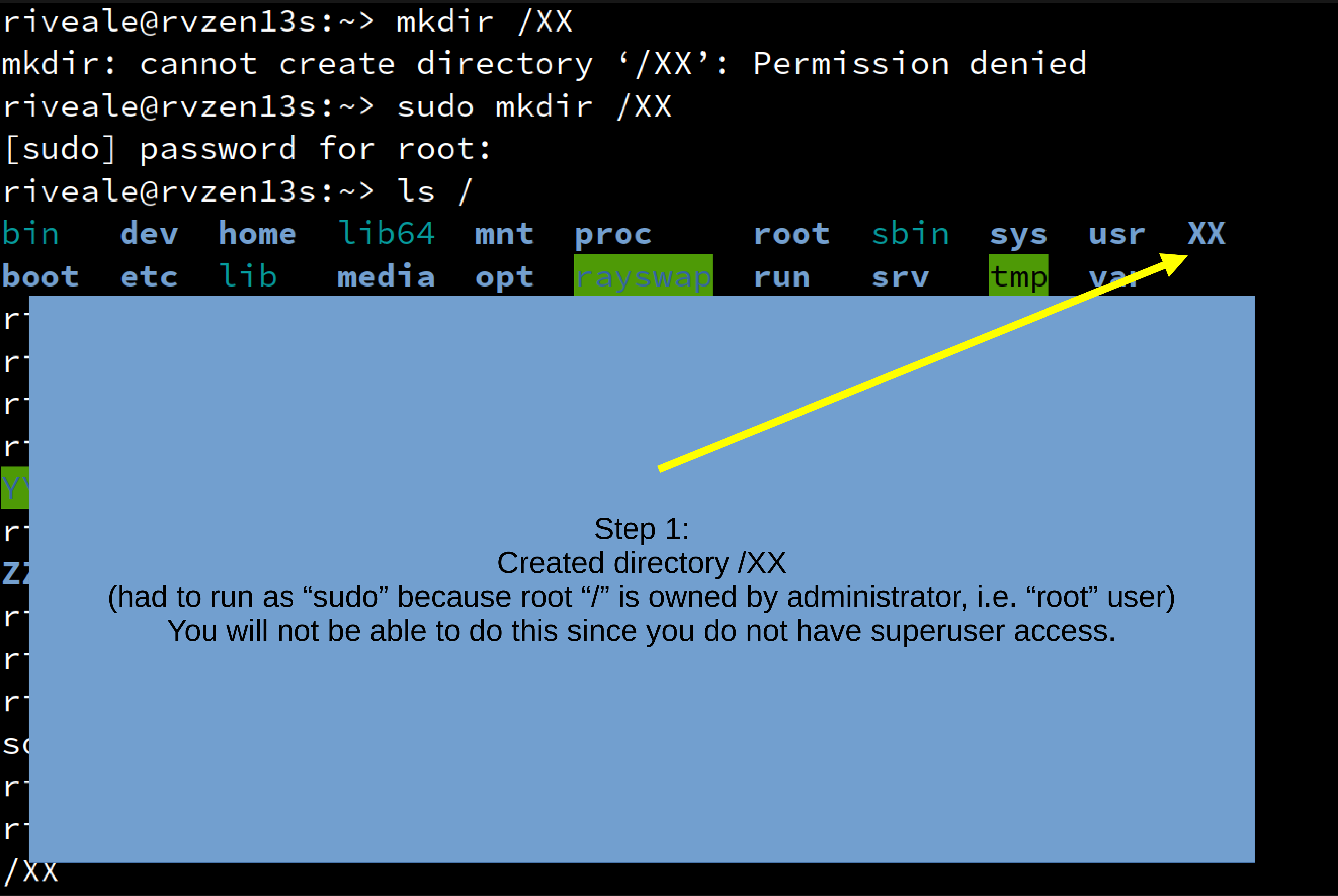
Absolute path:
/XX/YY

Absolute path:
/XX/YY/ZZ



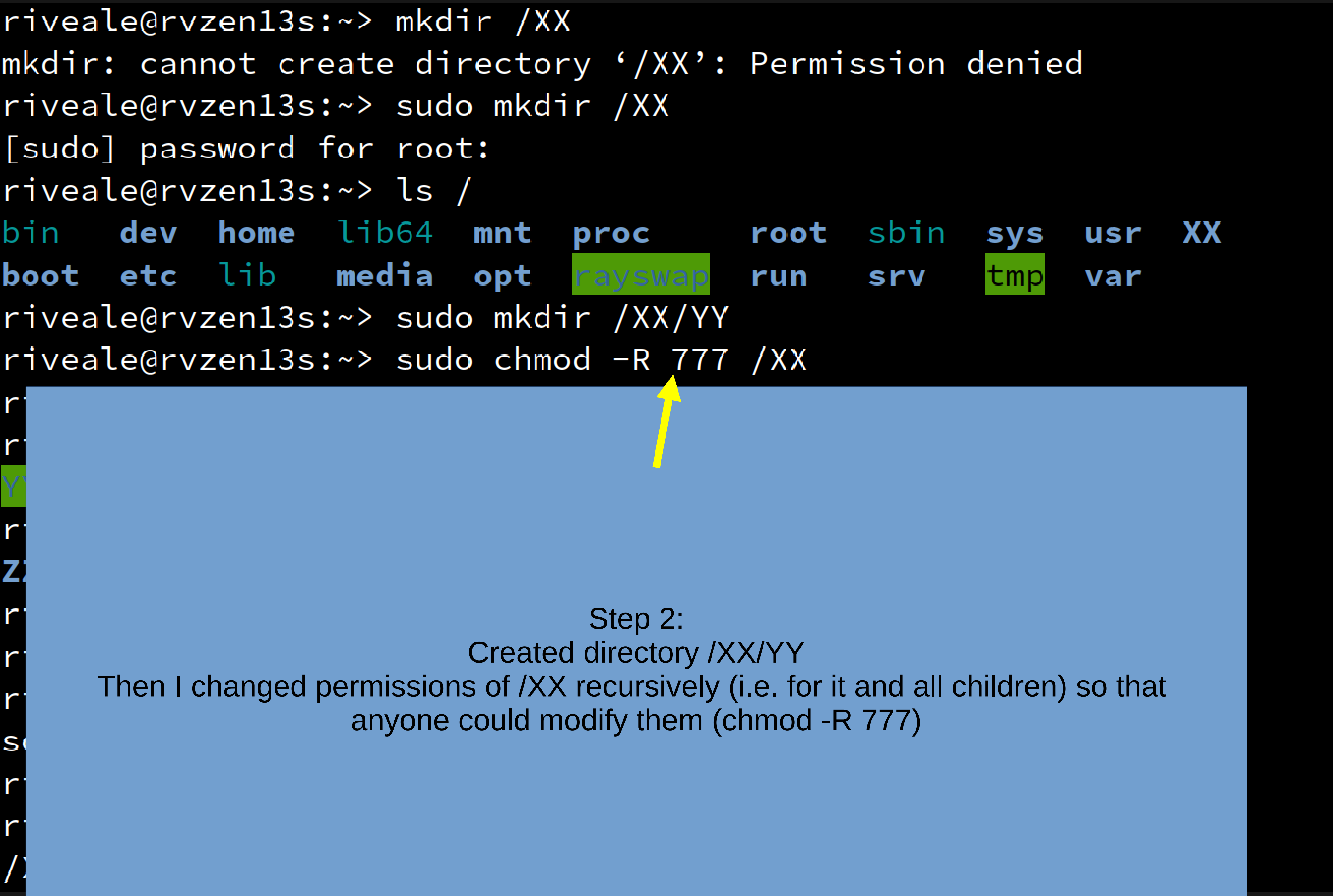
```
riveale@rvzen13s:~> mkdir /XX
mkdir: cannot create directory ‘/XX’: Permission denied
riveale@rvzen13s:~> sudo mkdir /XX
[sudo] password for root:
riveale@rvzen13s:~> ls /
bin    dev   home  lib64  mnt  proc      root  sbin  sys  usr  XX
boot   etc   lib   media  opt  rayswap   run   srv   tmp  var
r
r
r
r
YY
r
ZZ
r
r
r
sc
r
r
/XX
```

Step 1:
Created directory /XX
(had to run as “sudo” because root “/” is owned by administrator, i.e. “root” user)
You will not be able to do this since you do not have superuser access.



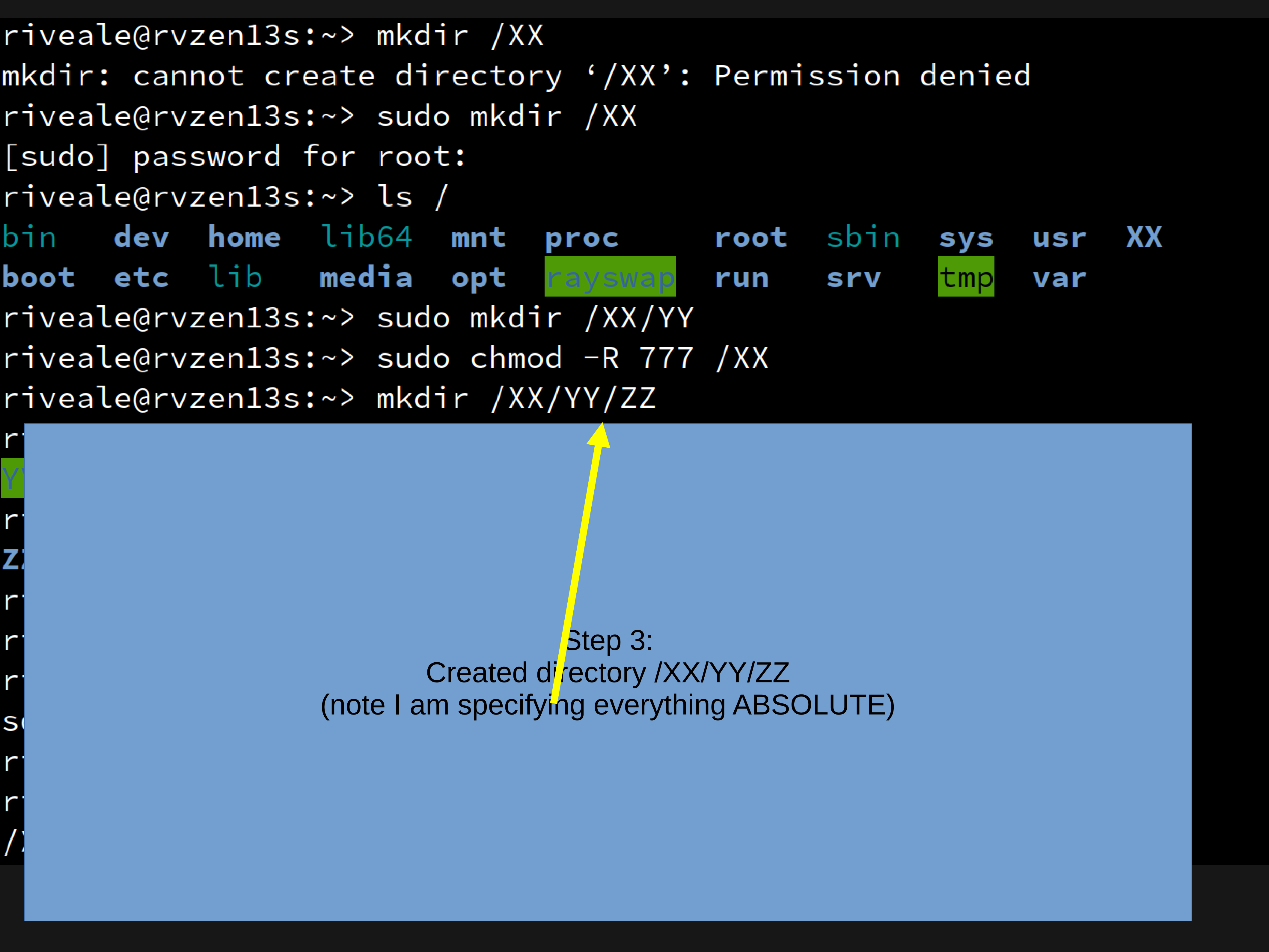
```
riveale@rvzen13s:~> mkdir /XX
mkdir: cannot create directory ‘/XX’: Permission denied
riveale@rvzen13s:~> sudo mkdir /XX
[sudo] password for root:
riveale@rvzen13s:~> ls /
bin   dev  home  lib64  mnt   proc      root  sbin  sys  usr  XX
boot  etc  lib   media  opt   rayswap   run   srv   tmp  var
r
r
r
r
YY
r
ZZ
r
r
r
sc
r
r
/XX
```

Step 1:
Created directory /XX
(had to run as “sudo” because root “/” is owned by administrator, i.e. “root” user)
You will not be able to do this since you do not have superuser access.



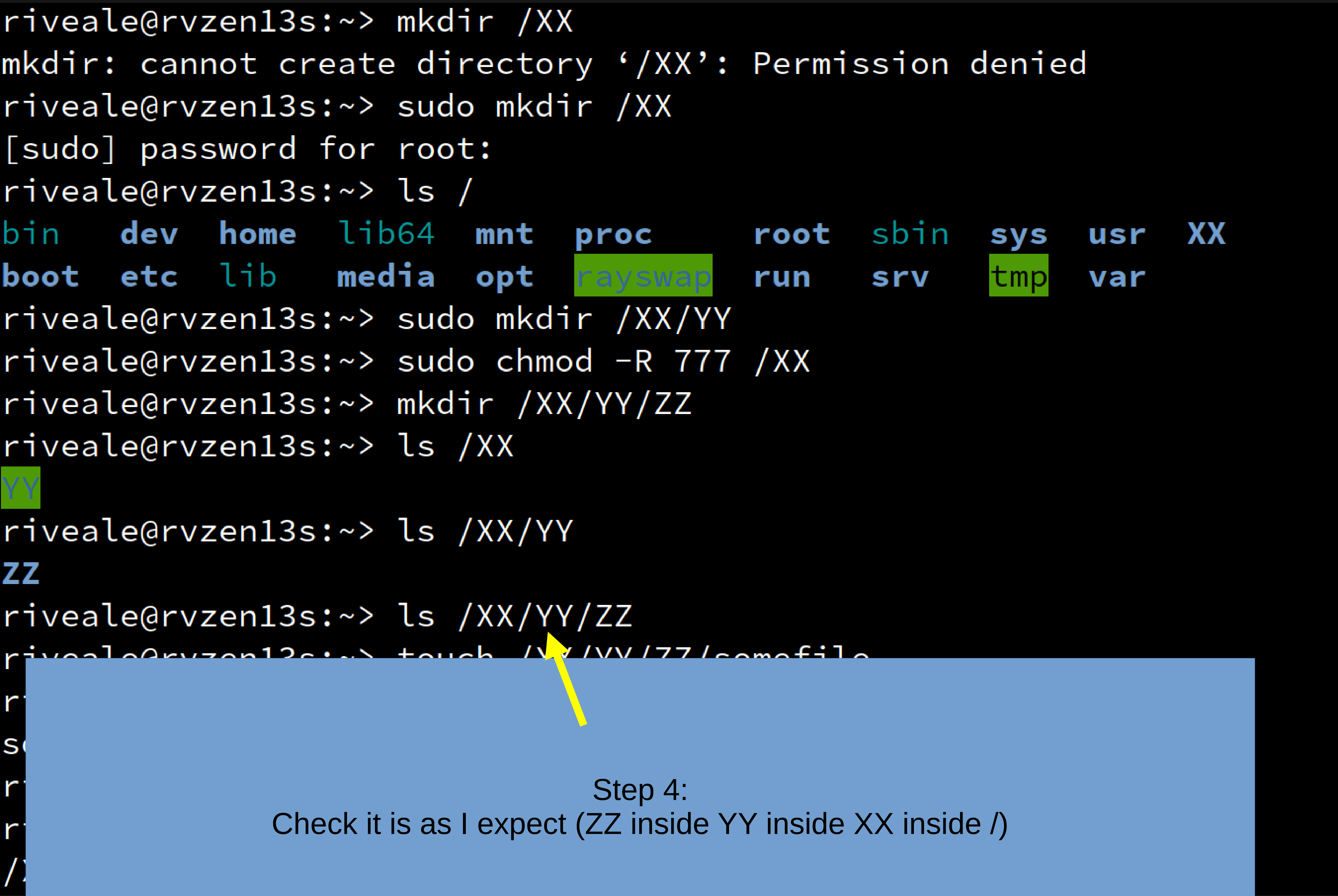
```
riveale@rvzen13s:~> mkdir /XX
mkdir: cannot create directory ‘/XX’: Permission denied
riveale@rvzen13s:~> sudo mkdir /XX
[sudo] password for root:
riveale@rvzen13s:~> ls /
bin   dev  home  lib64  mnt  proc     root  sbin  sys  usr  XX
boot  etc  lib   media  opt  rayswap  run   srv   tmp  var
riveale@rvzen13s:~> sudo mkdir /XX/YY
riveale@rvzen13s:~> sudo chmod -R 777 /XX
r
r
Y
r
Z
r
r
r
s
r
r
/
```

Step 2:
Created directory /XX/YY
Then I changed permissions of /XX recursively (i.e. for it and all children) so that anyone could modify them (chmod -R 777)



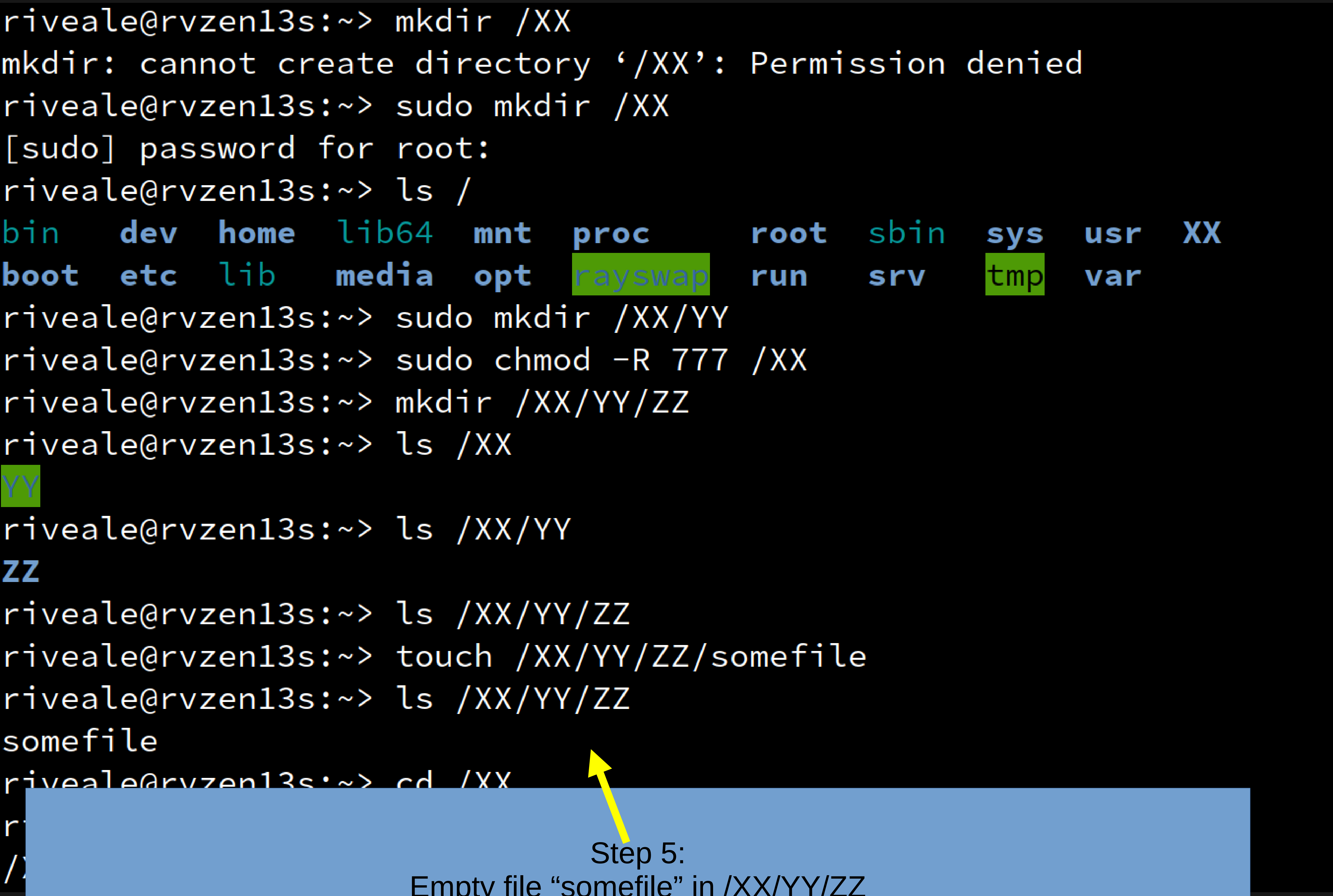
```
riveale@rvzen13s:~> mkdir /XX
mkdir: cannot create directory ‘/XX’: Permission denied
riveale@rvzen13s:~> sudo mkdir /XX
[sudo] password for root:
riveale@rvzen13s:~> ls /
bin    dev   home  lib64  mnt   proc      root  sbin  sys  usr  XX
boot   etc   lib   media  opt   rayswap   run   srv   tmp  var
riveale@rvzen13s:~> sudo mkdir /XX/YY
riveale@rvzen13s:~> sudo chmod -R 777 /XX
riveale@rvzen13s:~> mkdir /XX/YY/ZZ
r
Y
r
Z
r
r
r
s
r
r
/
```

Step 3:
Created directory /XX/YY/ZZ
(note I am specifying everything ABSOLUTE)



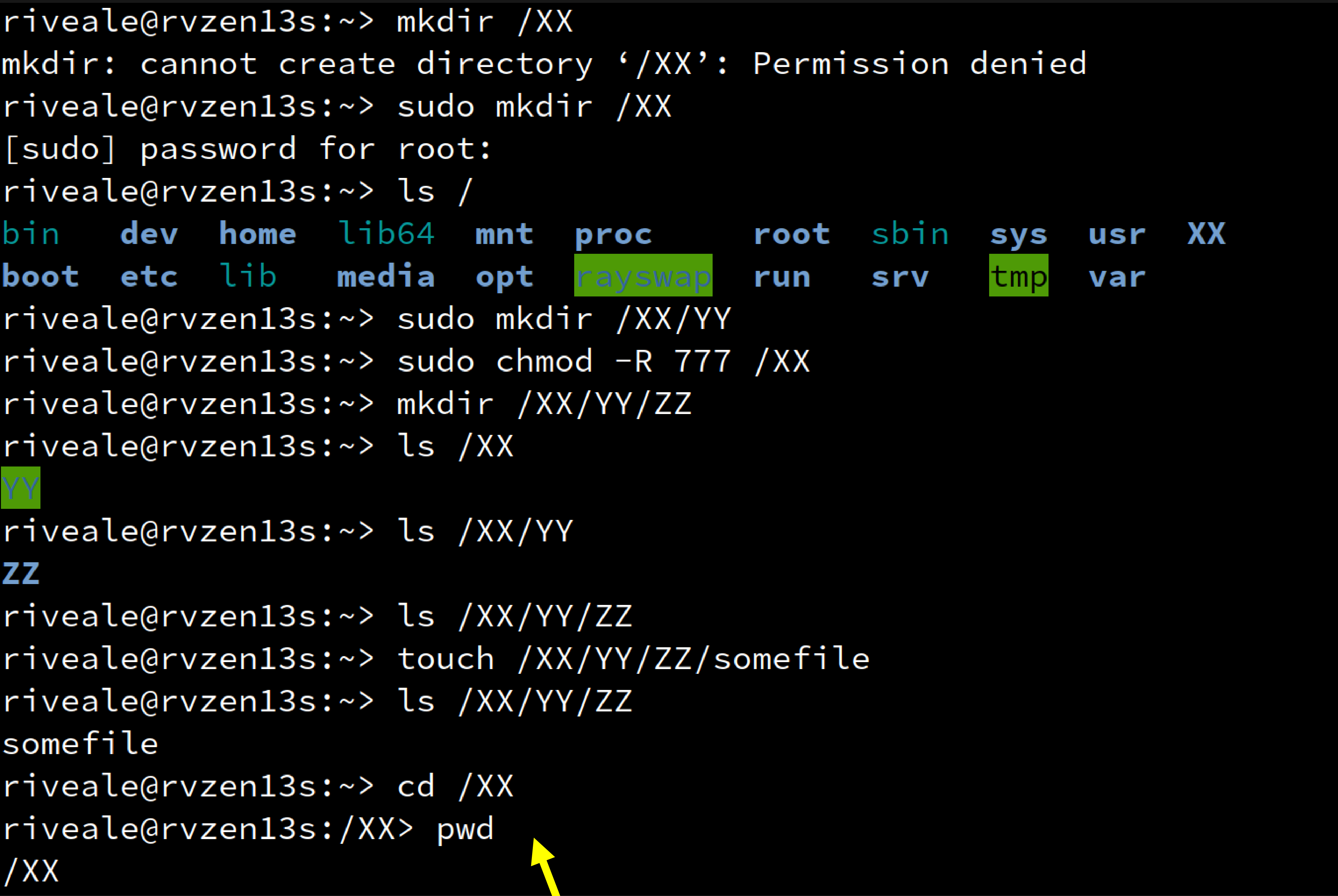
```
riveale@rvzen13s:~> mkdir /XX
mkdir: cannot create directory ‘/XX’: Permission denied
riveale@rvzen13s:~> sudo mkdir /XX
[sudo] password for root:
riveale@rvzen13s:~> ls /
bin    dev   home   lib64  mnt   proc      root   sbin   sys   usr   XX
boot   etc   lib    media  opt   rayswap   run    srv    tmp   var
riveale@rvzen13s:~> sudo mkdir /XX/YY
riveale@rvzen13s:~> sudo chmod -R 777 /XX
riveale@rvzen13s:~> mkdir /XX/YY/ZZ
riveale@rvzen13s:~> ls /XX
YY
riveale@rvzen13s:~> ls /XX/YY
ZZ
riveale@rvzen13s:~> ls /XX/YY/ZZ
riveale@rvzen13s:~> touch /XX/YY/ZZ/somefile
r
s
r
r
/
```

Step 4:
Check it is as I expect (ZZ inside YY inside XX inside /)



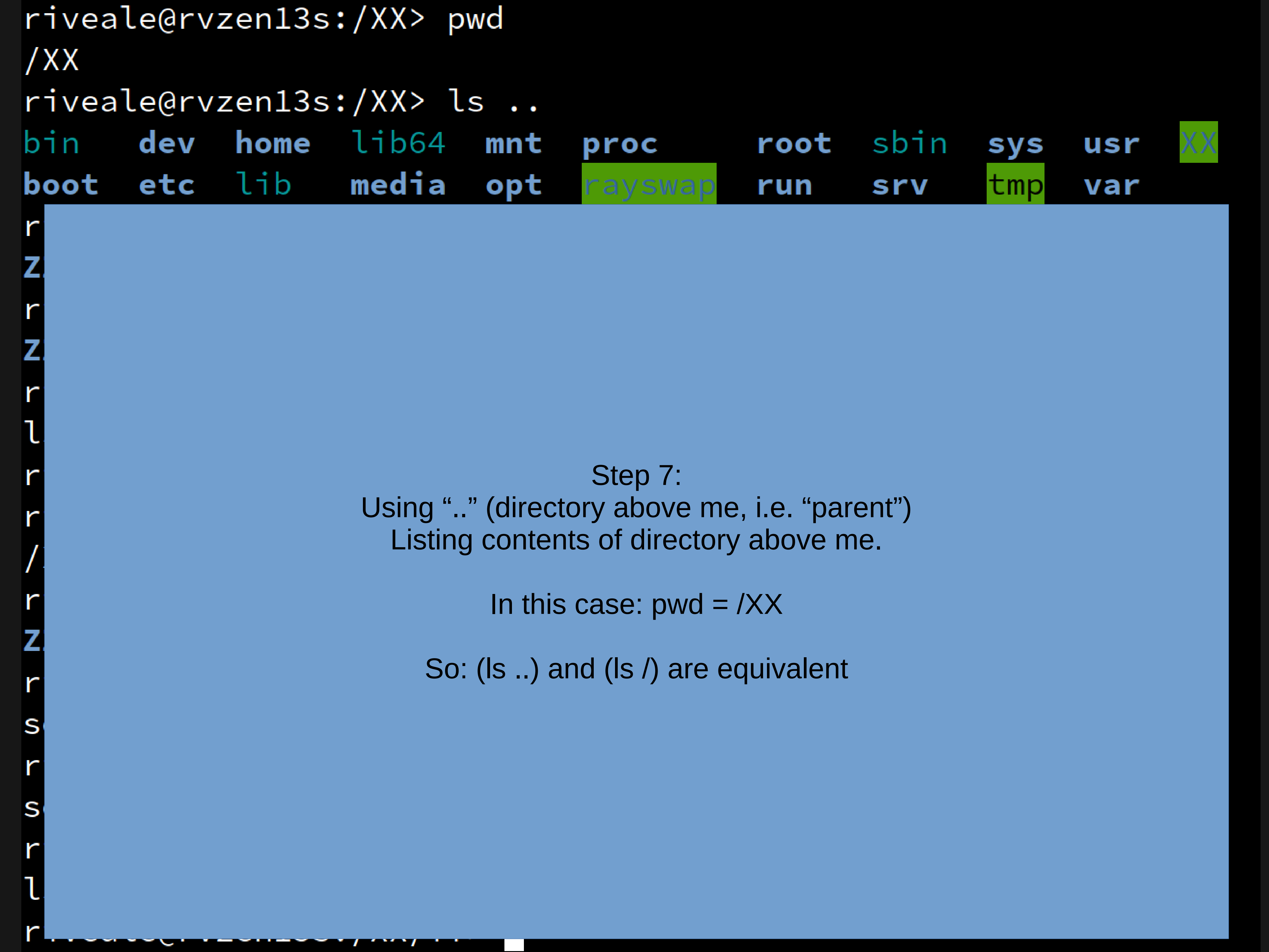
```
riveale@rvzen13s:~> mkdir /XX
mkdir: cannot create directory ‘/XX’: Permission denied
riveale@rvzen13s:~> sudo mkdir /XX
[sudo] password for root:
riveale@rvzen13s:~> ls /
bin   dev   home   lib64   mnt   proc      root   sbin   sys   usr   XX
boot  etc   lib    media   opt   rayswap   run    srv    tmp   var
riveale@rvzen13s:~> sudo mkdir /XX/YY
riveale@rvzen13s:~> sudo chmod -R 777 /XX
riveale@rvzen13s:~> mkdir /XX/YY/ZZ
riveale@rvzen13s:~> ls /XX
YY
riveale@rvzen13s:~> ls /XX/YY
ZZ
riveale@rvzen13s:~> ls /XX/YY/ZZ
riveale@rvzen13s:~> touch /XX/YY/ZZ/somefile
riveale@rvzen13s:~> ls /XX/YY/ZZ
somefile
riveale@rvzen13s:~> cd /XX
r
/
```

Step 5:
Empty file “somefile” in /XX/YY/ZZ



```
riveale@rvzen13s:~> mkdir /XX
mkdir: cannot create directory ‘/XX’: Permission denied
riveale@rvzen13s:~> sudo mkdir /XX
[sudo] password for root:
riveale@rvzen13s:~> ls /
bin   dev  home  lib64  mnt  proc     root  sbin  sys  usr  XX
boot  etc  lib   media  opt  rayswap  run   srv   tmp  var
riveale@rvzen13s:~> sudo mkdir /XX/YY
riveale@rvzen13s:~> sudo chmod -R 777 /XX
riveale@rvzen13s:~> mkdir /XX/YY/ZZ
riveale@rvzen13s:~> ls /XX
YY
riveale@rvzen13s:~> ls /XX/YY
ZZ
riveale@rvzen13s:~> ls /XX/YY/ZZ
riveale@rvzen13s:~> touch /XX/YY/ZZ/somefile
riveale@rvzen13s:~> ls /XX/YY/ZZ
somefile
riveale@rvzen13s:~> cd /XX
riveale@rvzen13s:/XX> pwd
/XX
```

Step 6:
Moving to being inside /XX

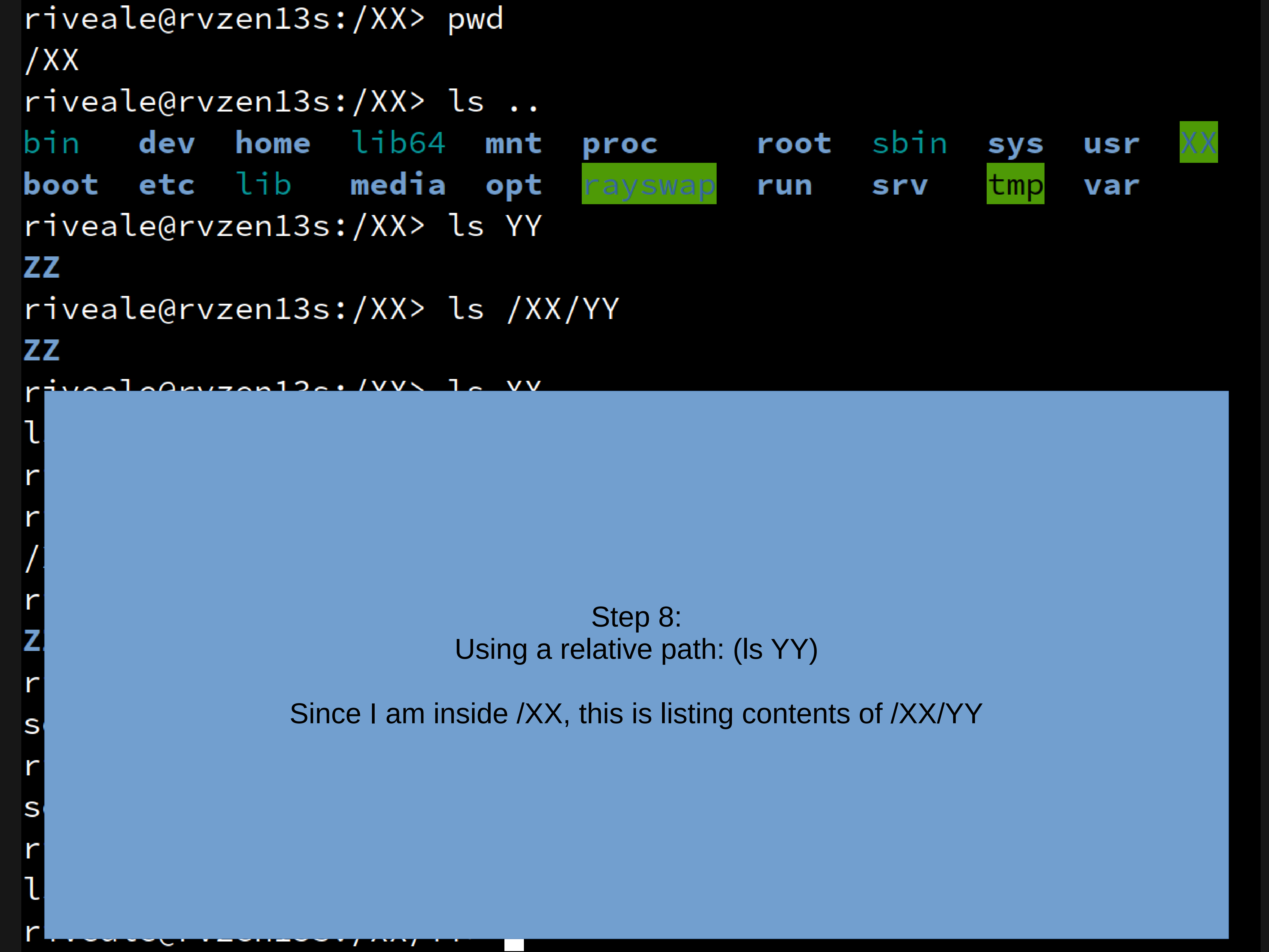


```
riveale@rvzen13s:/XX> pwd
/XX
riveale@rvzen13s:/XX> ls ..
bin   dev  home  lib64  mnt  proc     root  sbin  sys  usr  XX
boot  etc  lib   media  opt  rayswap  run   srv   tmp  var
r
Z
r
Z
r
l
r
r
/
r
Z
r
s
r
s
r
l
r
```

Step 7:
Using “..” (directory above me, i.e. “parent”)
Listing contents of directory above me.

In this case: pwd = /XX

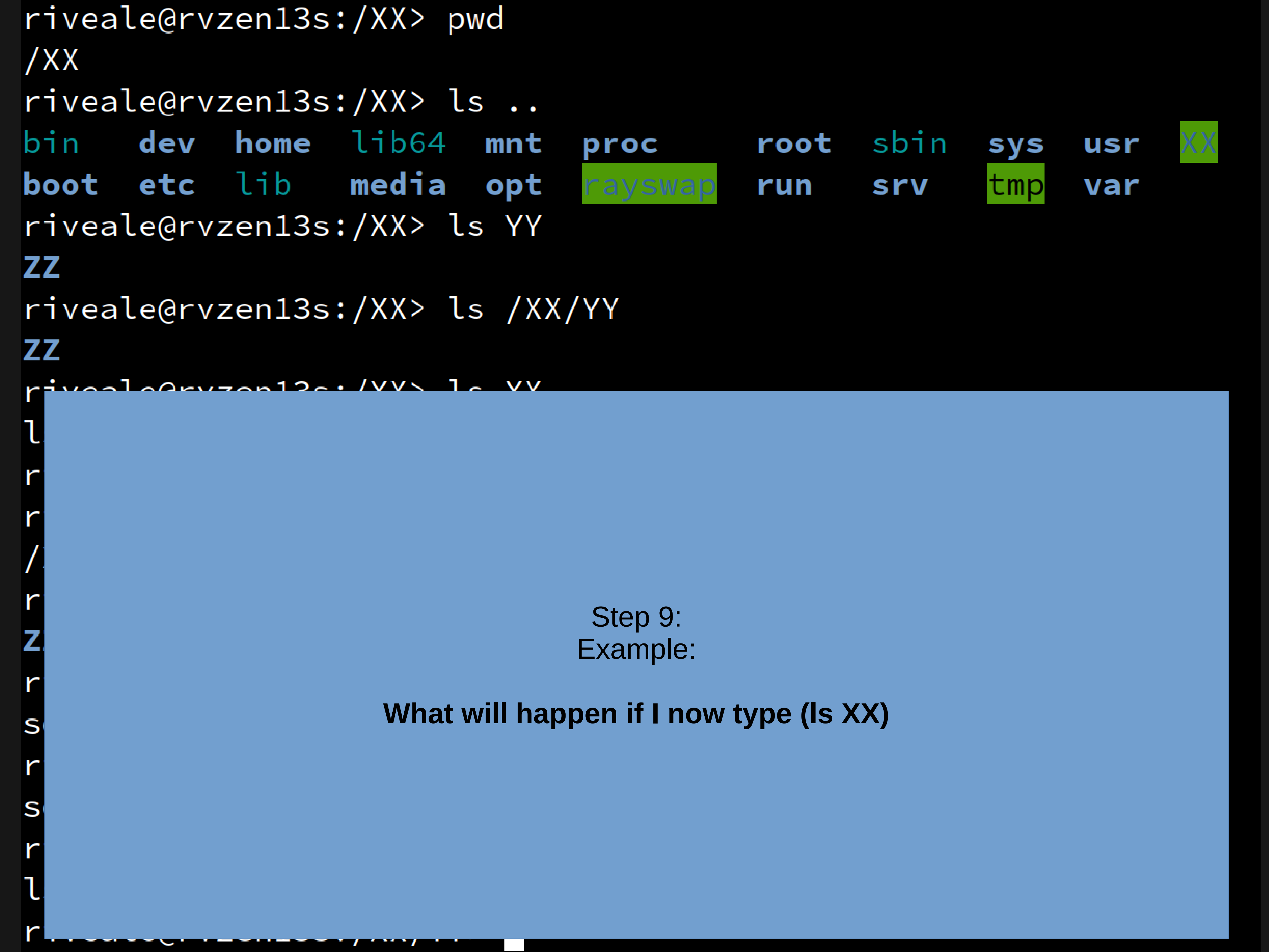
So: (ls ..) and (ls /) are equivalent



```
riveale@rvzen13s:/XX> pwd
/XX
riveale@rvzen13s:/XX> ls ..
bin   dev  home  lib64  mnt  proc     root  sbin  sys  usr  XX
boot  etc  lib   media  opt  rayswap  run   srv   tmp  var
riveale@rvzen13s:/XX> ls YY
ZZ
riveale@rvzen13s:/XX> ls /XX/YY
ZZ
riveale@rvzen13s:/XX> ls XX
l
r
r
/
r
Z
r
s
r
s
r
l
r
```

Step 8:
Using a relative path: (ls YY)

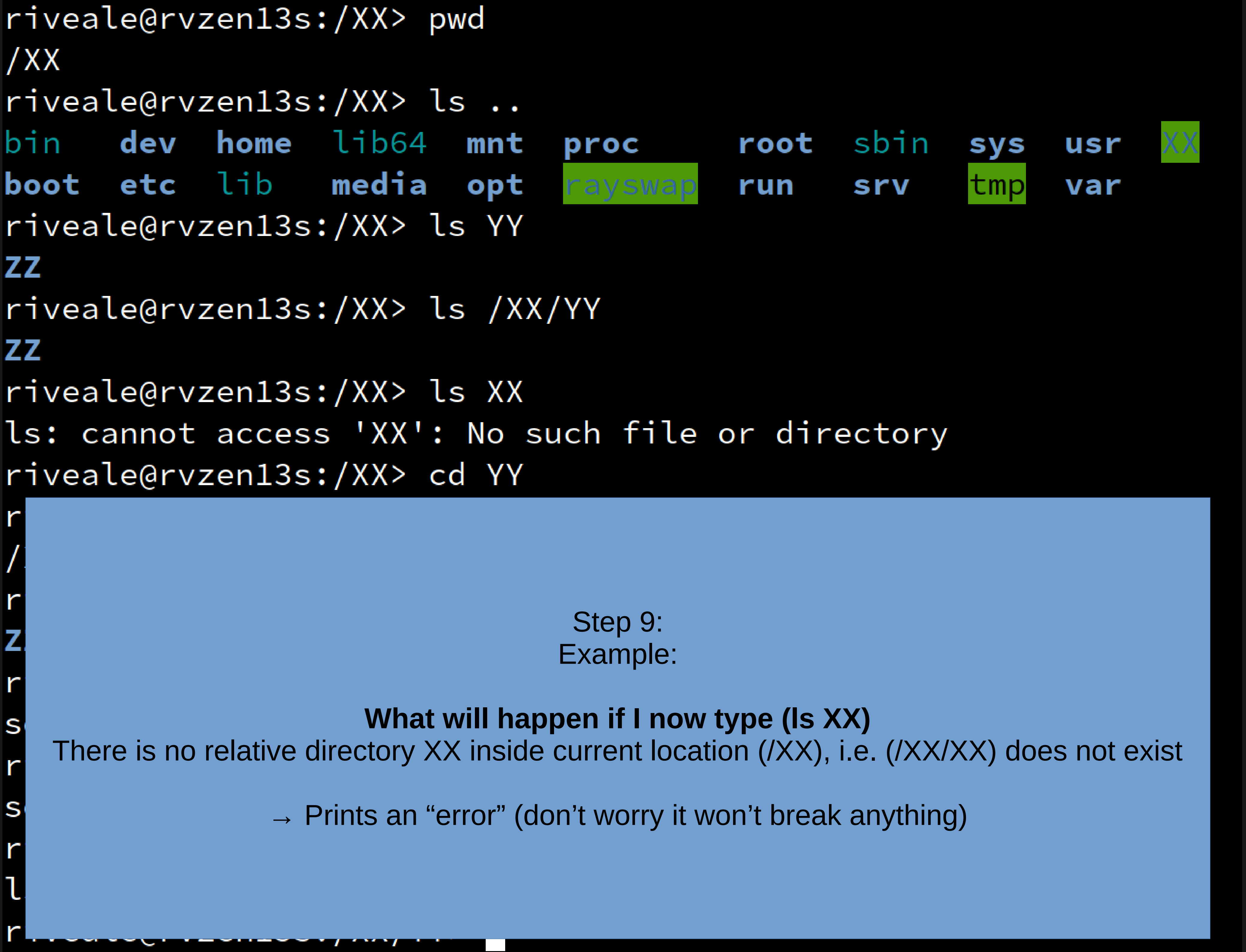
Since I am inside /XX, this is listing contents of /XX/YY



```
riveale@rvzen13s:/XX> pwd
/XX
riveale@rvzen13s:/XX> ls ..
bin   dev  home  lib64  mnt  proc      root  sbin  sys  usr  XX
boot  etc  lib   media  opt  rayswap   run   srv   tmp  var
riveale@rvzen13s:/XX> ls YY
ZZ
riveale@rvzen13s:/XX> ls /XX/YY
ZZ
riveale@rvzen13s:/XX> ls XX
l
r
r
/
r
Z
r
s
r
s
r
l
r
```

Step 9:
Example:

What will happen if I now type (ls XX)



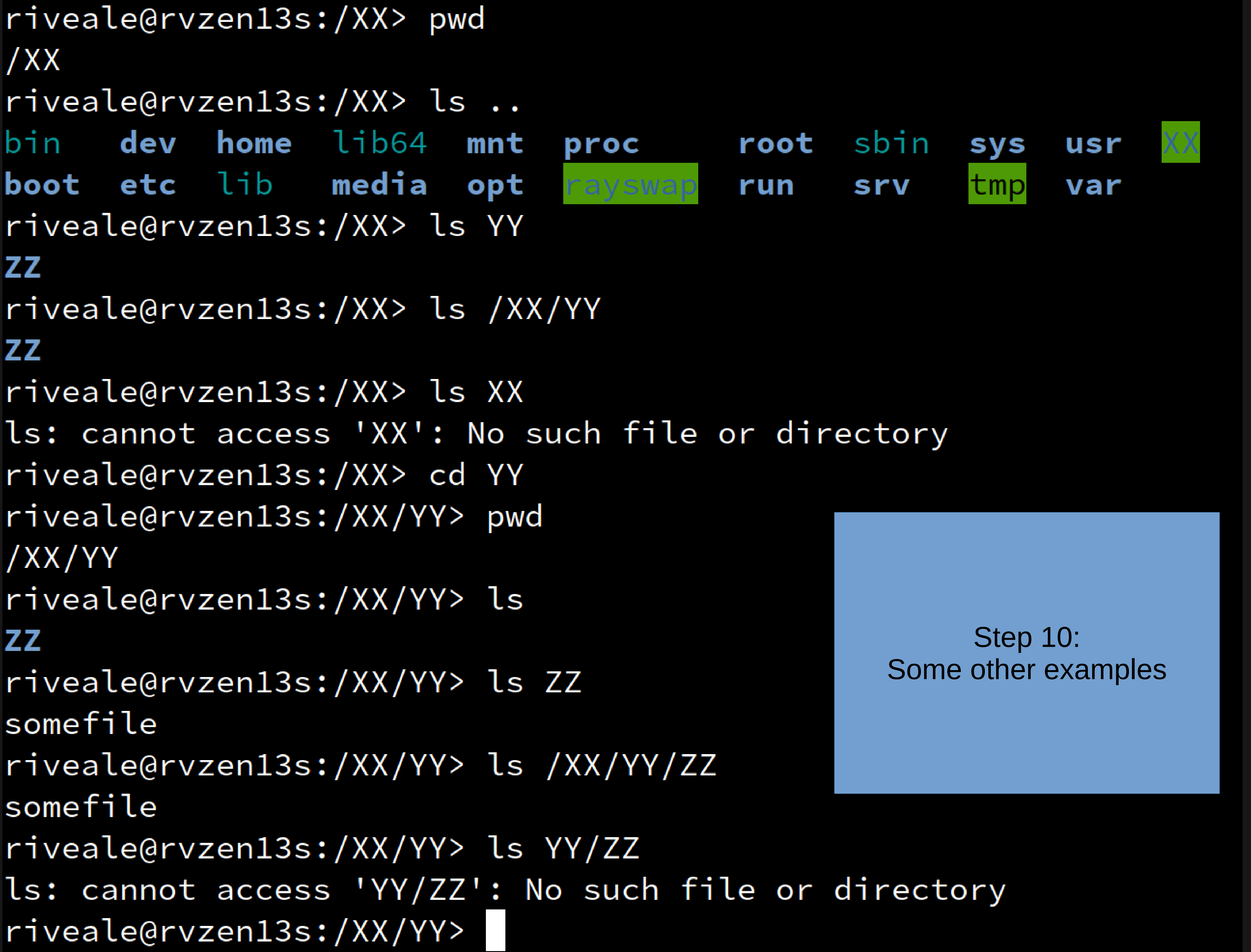
```
riveale@rvzen13s:/XX> pwd
/XX
riveale@rvzen13s:/XX> ls ..
bin    dev   home  lib64  mnt   proc      root  sbin  sys  usr  XX
boot   etc   lib   media  opt   rayswap   run   srv   tmp  var
riveale@rvzen13s:/XX> ls YY
ZZ
riveale@rvzen13s:/XX> ls /XX/YY
ZZ
riveale@rvzen13s:/XX> ls XX
ls: cannot access 'XX': No such file or directory
riveale@rvzen13s:/XX> cd YY
r
/
r
Z
r
s
r
s
r
l
r
```

Step 9:
Example:

What will happen if I now type (ls XX)

There is no relative directory XX inside current location (/XX), i.e. (/XX/XX) does not exist

→ Prints an “error” (don’t worry it won’t break anything)



```
riveale@rvzen13s:/XX> pwd
/XX
riveale@rvzen13s:/XX> ls ..
bin   dev  home  lib64  mnt  proc     root  sbin  sys  usr  XX
boot  etc  lib   media  opt  rayswap  run   srv   tmp  var
riveale@rvzen13s:/XX> ls YY
ZZ
riveale@rvzen13s:/XX> ls /XX/YY
ZZ
riveale@rvzen13s:/XX> ls XX
ls: cannot access 'XX': No such file or directory
riveale@rvzen13s:/XX> cd YY
riveale@rvzen13s:/XX/YY> pwd
/XX/YY
riveale@rvzen13s:/XX/YY> ls
ZZ
riveale@rvzen13s:/XX/YY> ls ZZ
somefile
riveale@rvzen13s:/XX/YY> ls /XX/YY/ZZ
somefile
riveale@rvzen13s:/XX/YY> ls YY/ZZ
ls: cannot access 'YY/ZZ': No such file or directory
riveale@rvzen13s:/XX/YY> 
```

Step 10:
Some other examples

Today: Creating Accounts

GOAL

- 1) Create list of all student names and IDs in the class
- 2) Create a unique “username” for each student by combining name and part of ID
- 3) Create an account on the server with that username and a password

Concepts

- Languages used today:
  - 1) Python
  - 2) Shell script (bash)
- Concepts used today:
  - 1) Variables (string variables, character variables, integers)
  - 2) Loops (iteration) → “for”, “while”
  - 3) Conditional branching (“if”)
  - 4) Arrays (lists) of variables
  - 5) “pandas” (a python library for tabular data processing)
- Editors:
  - Emacs (you can use nano/vim)

Continued in class…