

# Parallel and Distributed Systems

- Antonio Espinosa

[Antoniomiguel.espinosa@uab.cat](mailto:Antoniomiguel.espinosa@uab.cat)

QC3016 Tuesdays 12-14

Class content and practical work available in Virtual Campus

Evaluation: deliver of practical work

# Tools for data analysis in Distributed Systems

- System tools
- Database tools
  - Relational data bases
  - Distributed data Systems: Hadoop echosystem
- Cloud computing platforms

# Introduction to Linux

Parallel and Distributed Computing Systems

# Agenda

- Introduction
- Operating system services
- Basic linux utilities
- File management
- Folder management
- Compression of data

# What is linux?

What do you think is linux?

# Linux

- The **linux** kernel takes care of managing the computer resources (CPU, disks, file system, RAM, networkcards, ...)
- The operating system provides the means of communication with the linux kernel through well known commands and programs

# GNU Linux

- GNU Project started at 1983 had the goal of creating
  - a complete Unix-compatible operating system
  - Composed entirely of free software
- Joined with Linus Torvalds Linux kernel
- Developers worked to integrate GNU components with the Linux kernel

# Why linux?

- **Open source:** the *kernel* code is viewable by anyone.
- **Open development:** anybody can propose enhancements /change it to their needs.
- **Free-to-use:** anybody can use the kernel to drive their own computer or device (always interesting in academics: a lot of development here)
- **Stable:** it is one of the most stable OS you can find.



# Linux in your computer

- Different **distributions**
- They all use the Linux OS kernel
- They add their graphical user interface
- They add tools and applications (backup, text editor,...)
- They provide a software package manager initialization & configuration scripts  
commercial support
- There are many distributions (600+), but perhaps only a few that really matter...

# Distribution families

- **Server**

- Commercial support (company driven): SUSE Linux ES, Red Hat EL, Ubuntu Server
- Free and community driven: Debian, CentOS, Scientific Linux

- **Personal computer**

- For home use: Mint, Ubuntu ..

# Desktop environments

- KDE
- Gnome
- Unity
- Cinnamon

[http://en.wikipedia.org/wiki/Desktop\\_environment](http://en.wikipedia.org/wiki/Desktop_environment)

# File managers

- Nautilus
- Dolphin
- Krusader
- Thunar

[http://en.wikipedia.org/wiki/File\\_manager](http://en.wikipedia.org/wiki/File_manager)

# Run Linux in “Live mode”

- Put Linux on a USB stick, and tell your computer to start up from the USB stick, instead of the hard drive
- <http://unetbootin.sourceforge.net/>

# Keywords

- operating system
- linux
- GNU
- open source
- distribution
- desktop environment
- live CD

# Welcome to Linux

A quick guide to Linux usage

# The structure of Linux

- A Linux system is organised
  - Drives
  - Partitions
  - Folders
  - Files
- A basic task in data analysis pipelines is handling and storing large datasets



# Knowing the Linux file hierarchy

- Open the “computer” icon on the desktop
  - Locate hard disk icon: File System
  - Locate CD/DVD drive

# The root directory

- Linux is installed on the disk of the computer
- The root directory is called “/”
  - It is the start of the file system
- Look for the home folder

# The home folder

- One of the folders under the root directory is called “home”
- Every user of a Linux system has a folder in “/home”
  - Look for your folder in “/home”

# Permissions in your home

- Linux security is applied to every folder and file
- You can only create files and folders in you own home
- The rest of the folders are managed for the administrator of the system

# Create a “bin” folder in your home

- Use the right-click button of the mouse or use the File menu
  - Create new folder
- Create a folder and name it “bin”

# Visualize the tree structure

- Go back to the root directory. Then, change the view to “List” using the menu option

# Visualize the tree structure

- Clicking on the “+” expands the contents of that folder.
- A “path” is the location of a file or a folder from the “root” directory  
/home/toni/Downloads/data.txt

# Where are programs located

- /bin
- /lib
- /lib64
- /sbin
- /usr



# Disk and share information

- /dev
- /media
- /mnt

# The administrator's home

- /root

# Configuration files

- /etc

# Where to store temporal data

- /tmp
- /var

# “Everything is a file in linux”

Devices and status are accessible by reading the contents of the files

- Disk: `/dev/sda`
- Memory: `/proc/meminfo`
- Mouse: `/dev/mouse1`
- Keyboard: `/dev/input1`

# Configuration files are text

- Go to /etc
- Open /etc/passwd file
  - What do you recognise?

# Data analysis Linux tools

- Linux includes a list of tools to manipulate, search and analyse text files
- The best way of using text management tools is by using the terminal
- But, the terminal is a complex non-graphical environment
  - Press ctrl+alt+F1
  - Commands are text and usage must be learned
  - Press ctrl+alt+F7

# Use a terminal program

- Open a terminal program using the menu
  - Look for terminal application in your desktop
  - Search for “terminal” application



# The terminal

- The way of interacting with the terminal is by typing the commands and the data files to work with
- What you type is interpreted by the program “bash” which does what the command asks
- Bash is a shell program: a command interpreter

# Terminal work basics

- A command line is always positioned somewhere in the file system
- What you type is case-sensitive
- Prompt usually shows
  - Username
  - Machine name
  - Current working directory

# Learning command-line tools

- type: date
- type: whoami
- “pwd”: Print Working Directory
  - Use it to check where the shell is located
- Type <command> and press <enter> to execute the command

# Navigate in the file system

- ls: list contents of current directory
- cd <dir>: change to this directory
  - The <dir> word must be a directory
  - This additional word is an argument
- What will happen if we type “ls” now?

# Try these commands

- `ls`
- `ls -l`
- `ls -lt`
- `cd`
- `ls`
- `pwd`
- `cd .`
- `cd ..`
- `history`
- `<up arrow>`

# Navigating in the file system

- The result is that the prompt has changed position from the original folder to the new location described by the argument

# Commands run on files

- We usually provide file names as arguments to the command
- We can check the type of a file with the command “file”  
\$ file data.txt

# Help please!

- How do you know the commands and the arguments of everything?
- Linux systems have interactive help manuals to look for information
  - The program “man” displays the manual of the command requested
  - For example, the manual of “ls”: `man ls`



# Other ways of getting help

- which – Display which executable program will be executed
- help – Get help for shell builtins
- man – Display a command's manual page
- apropos – Display a list of appropriate commands
- info – Display a command's info entry
- whatis – Display a very brief description of a command

# Data compression

Compressed data is used...

- When moving data, do it in its smallest form
- When data is not recently used
- When a project is finished but data must be stored

# Linux compression tools

Most widely used compression tools:

- gzip: block sorting compression

gzip file

gunzip file

- bzip2: GNU zip

bzip2 file


bunzip2 file

`gzip data.txt -> data.txt.gz`

# Compressing folder and contents

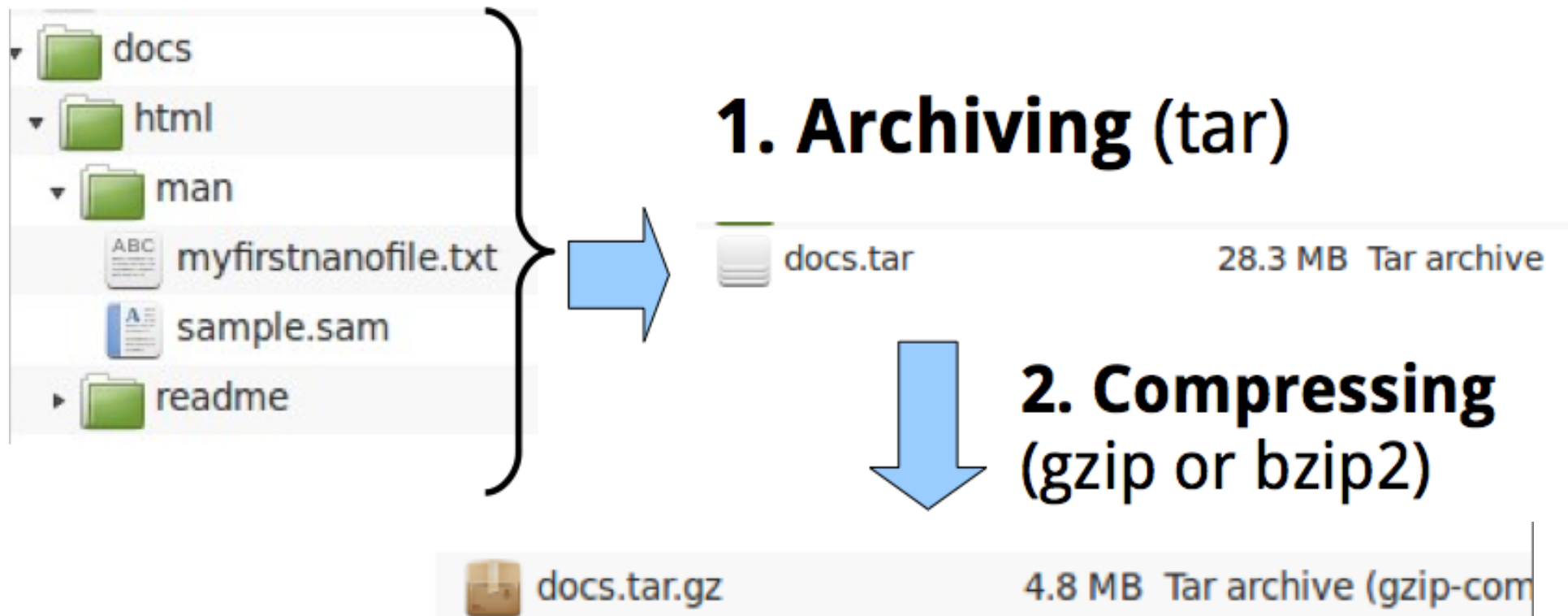
- tar is a tool for bundling a set of files or directories into a single archive
- The resulting file is called a tar ball
- To create a tarball:  

```
$ tar -cf archive.tar file1 file2
```
- To extract files from tarball:  

```
$ tar -xvf path/to/archive.tar
```

# Typical compression case

- First archive and then compress



# Tar+compression

- Use tar with the z or j option
- Create a compressed tar archive
  - \$ tar cvfz mytararchive.tar.gz docs/
  - \$ tar cvfj mytararchive.tar.bz docs/
- Decompress a compressed tar archive
  - \$ tar xvfz mytararchive.tar.gz
  - \$ tar xvfj mytararchive.tar.bz

# Working with large text files

- Tools to read compressed text files to avoid unpacking
  - zcat file
  - bzcat file
- Compression represents a balance between time and storage space.  
Less space takes more time

# Checking available space in disk

- To check the storage that is used on the different disks use  
\$ df -h
- To check the size of files or directories  
\$ du -sh \*



# WORK!: tar and gzip

- Download dataset.tar.gz
- Unpack file dataset.tar.gz
- Check if files are OK
- Compress jan2017articles.csv with bzip2
- Do the same with gzip
- Compare the sizes of the compressed files
- Pack dataset data folder with tar and gzip

# Exercise 2: large files

- Download articles-large.tar.gz
- How to check if file is downloaded correctly?
  - Use `man md5sum`, look for `-c` option
- Use `md5sum` with the `.md5` file
- What is the use of:
  - file `articles-large.tar.gz`
  - `apropos gzip`

# More about commands

- Every command has arguments and options
  - come before the arguments
  - are separated by spaces
  - Start with “-” or “--”
- For example: “ls -l /bin”
  - ls: the program
  - l: the option
  - /bin: the argument
- Most used option: “--help”

# Using short options

- You can use multiple options to the command. The order is seldom important

```
ls -l -t /bin
```

```
ls -t -l /bin
```

- Short options can be combined into one string

```
ls -r -l -t
```

```
ls -rtl
```

# Options can also have arguments

- Example: show the contents of the directory, sorted by the size of the files

`ls --sort=size /bin`

`ls -w 80 /bin`

`ls -lr -w 80 /bin`

`ls -rlw 80 /bin`

`ls -wrl 80 /bin`

`ls -w80 /bin -lr`

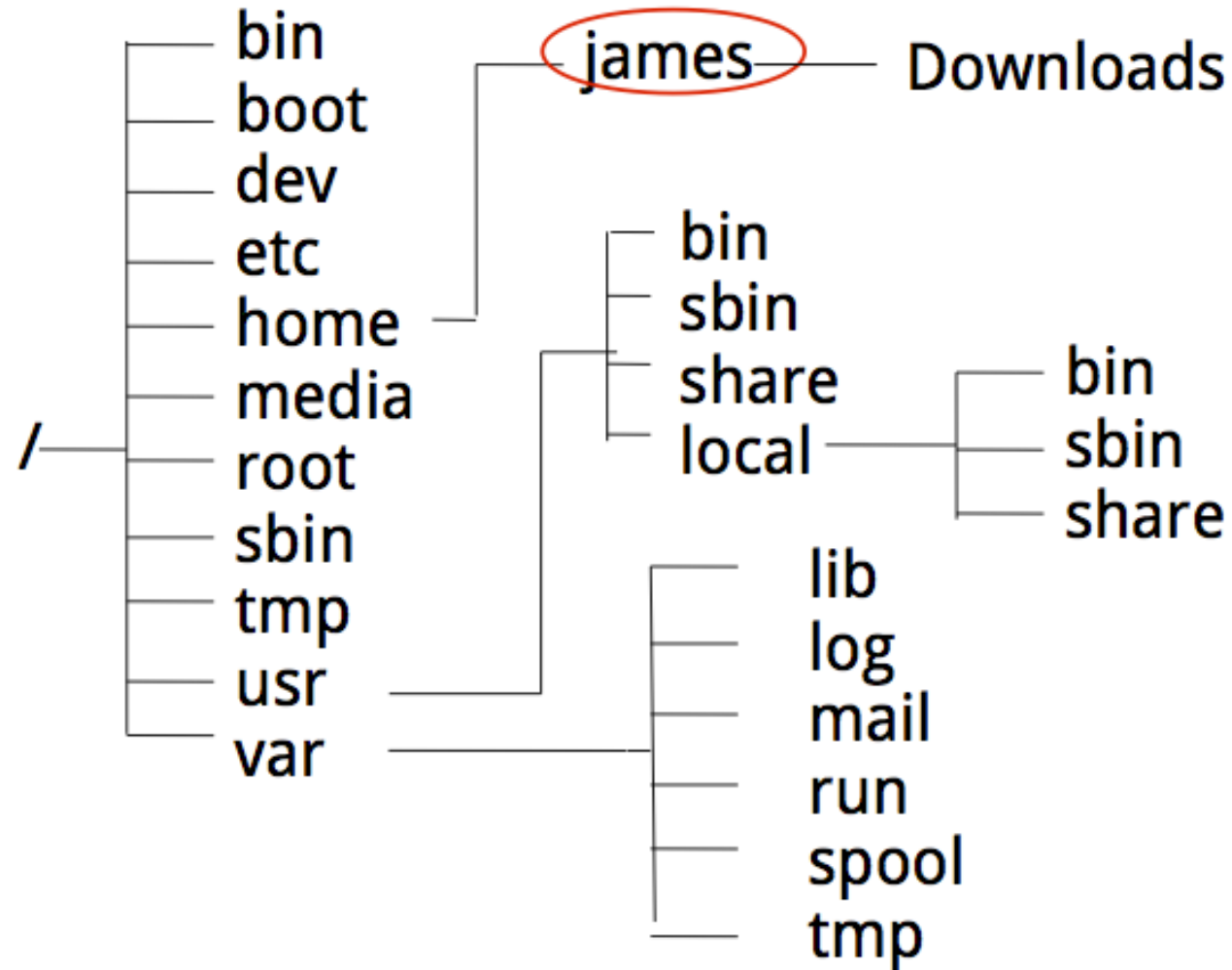
# Basic commands

Command	Explanation
pwd	Print working directory
ls	Print content of directory
cd	Change directory
cat	Print the contents of a file
cp	Copy a file
mv	Move a file
rm	Remove a file
less	Read the contents of a file
clear	Clear the terminal screen
head	Show the first 10 lines of a file
tail	Show the last 10 lines of a file
nano	Text editor, to modify text files
wget	Download a file from an URL

# Terminal productivity

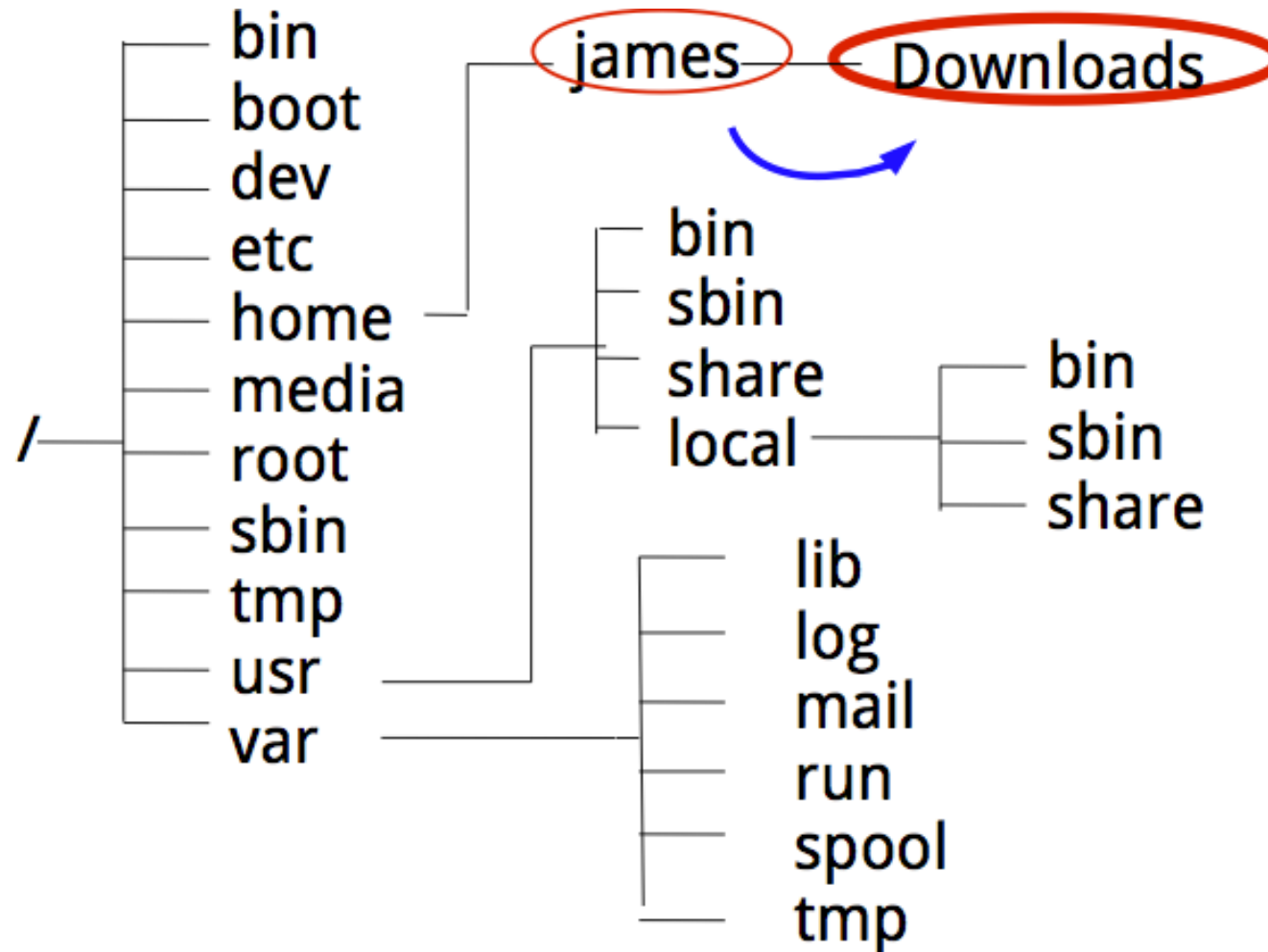
- Start typing “ls Down” and then press <tab>
  - What happens?
- Start typing “ls XX” and press <tab>
  - What happens?
- Play with “up arrow”
- Type
  - \$ history

# Paths and the working directory

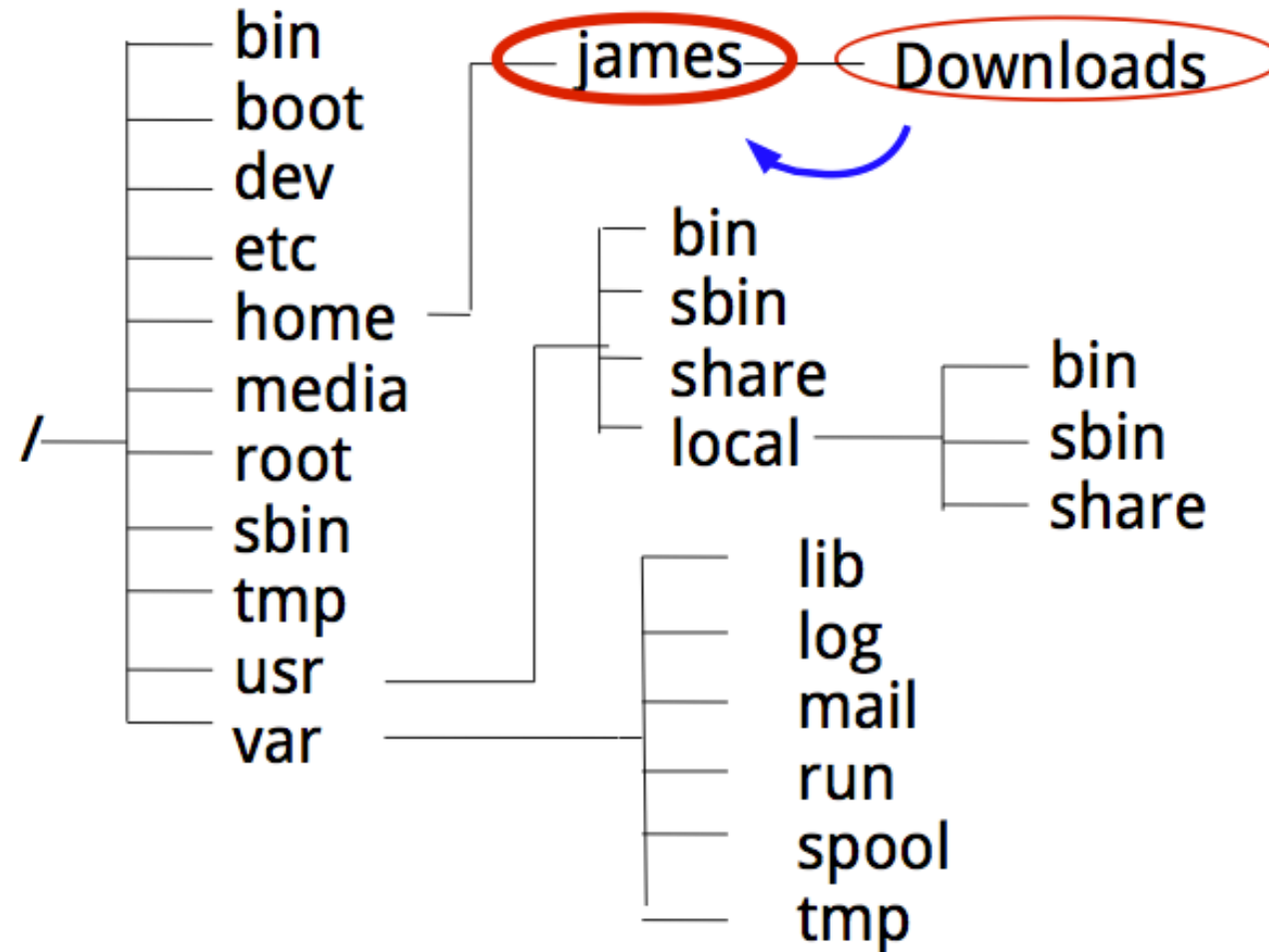




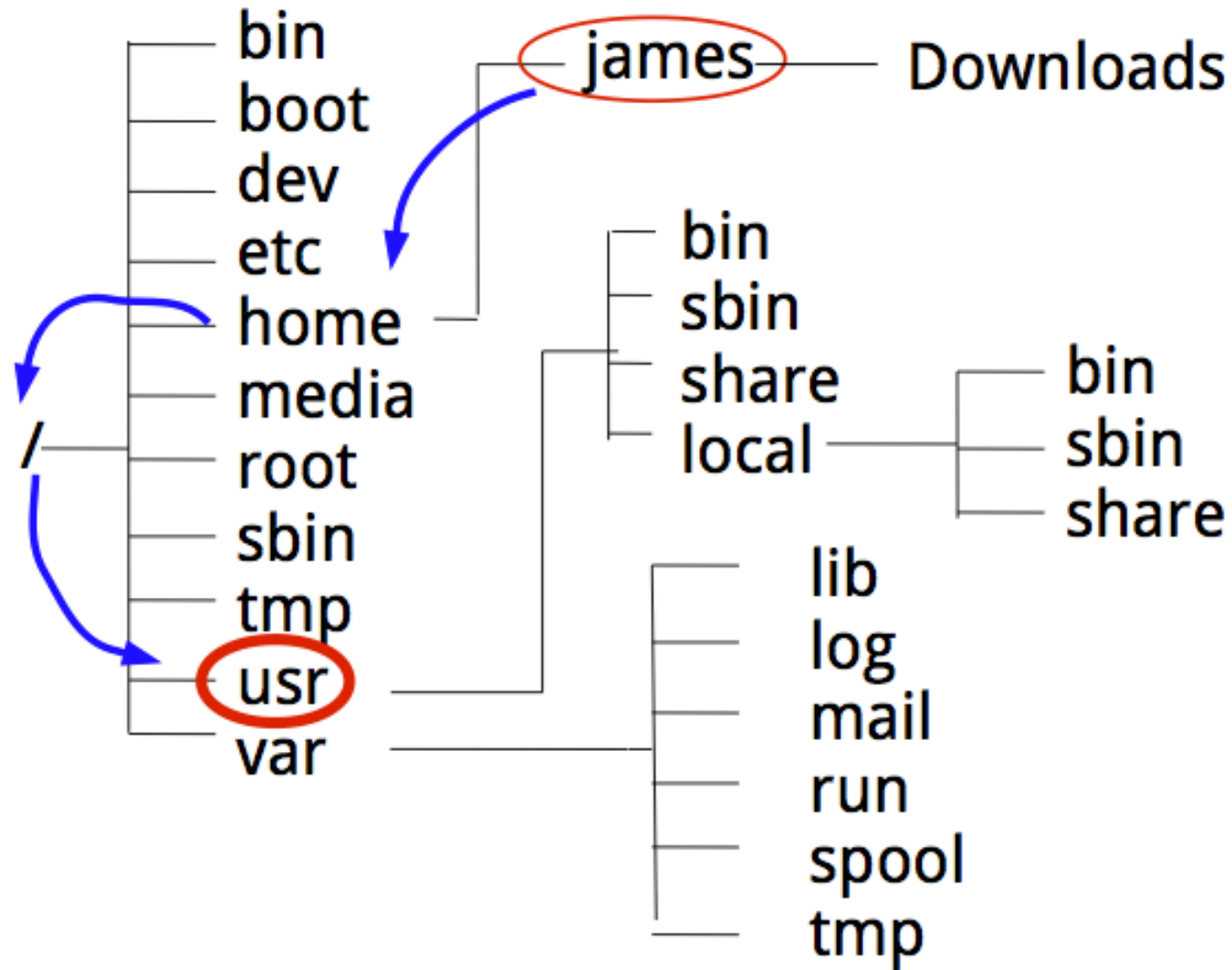
# cd Downloads



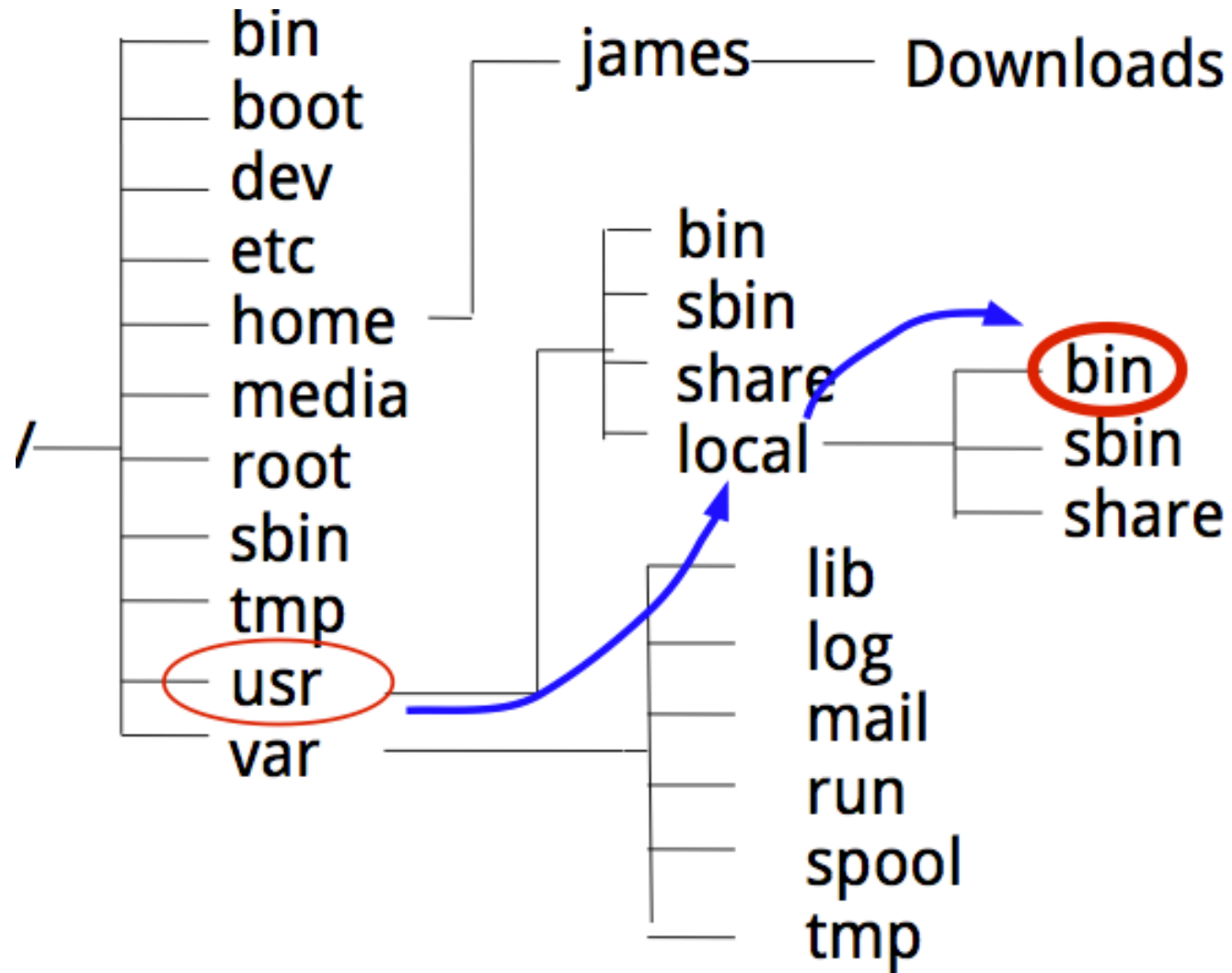
~/Downloads \$ cd ..



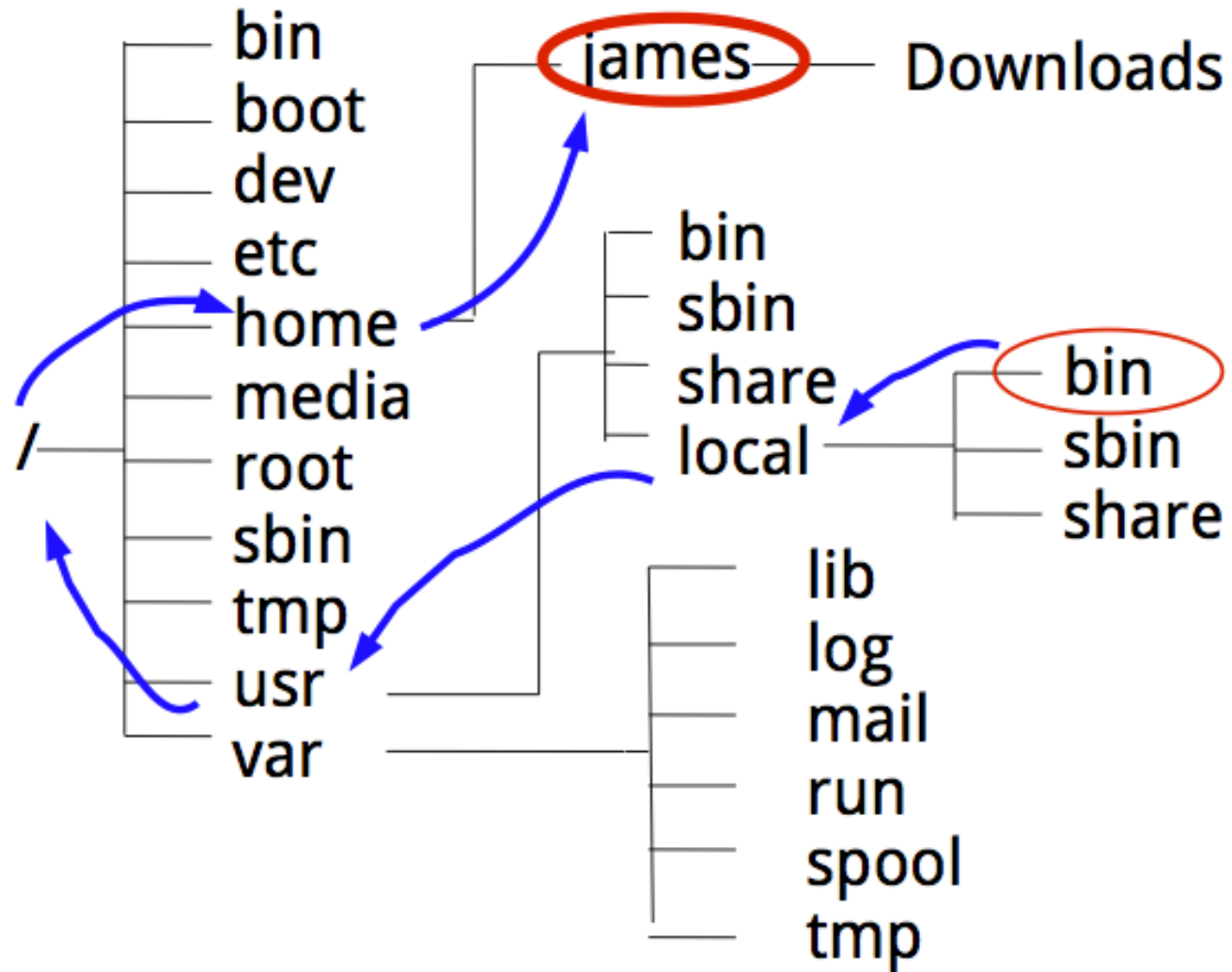
~ \$ cd ../../usr



```
/usr $ cd local/bin
```



```
/usr/local/bin $ cd ../../../../home/james
```



# Relative paths

- The path is set up relative to the current working directory  
Which steps we take from the current directory to reach the destination  
/usr/local/bin \$ cd ../../../../home/james  
/usr \$ cd local/bin  
cd ../../usr  
cd Downloads  
/Downloads \$ cd ..

# Absolute paths

- Sometimes is more convenient to point to the complete or absolute path

```
cd /home/james/Downloads
```

# Hidden directories

- Compare the output of `ls` versus `ls -a`
- Check with the file manager
- Hidden files and folders start with “.” which are not shown by default



# WORK!

- Unpack dataset2.tar.gz in a new project folder
- Go to /home/<your user>/dataset2
- `ls *.csv`
- `ls *2017.csv`
- `ls {may2017,may2016}.csv`
- `ls -lt may201?.csv`

# Moving data between directories

- Copy files

\$ cp <what> <to where>

- Move files

\$ mv <what> <to where>

- Remove files

\$ rm <filename>

- Create directory

\$ mkdir <foldername>

# Reading files

- \$ cat: display the content of the file at once
- \$ less: display the content page by page
- \$ nano: edit the content of the file
- \$ tail : show the last lines of the file
- \$ head: show the first lines of the file
  - How to show the 10 first lines of a file?

# WORK!

- Create a backup dataset3 folder
- Copy all csv files to backup folder
- Can you eliminate backup folder with rmdir?
  - What do you need to do first?

# Symbolic links

- A symbolic link is a file which points to the location of another file
- When we need to have copies of large files, if we use symlinks no actual data is copied
- When the original file is deleted, symlinks are not valid

# Create a symlink

```
mkdir dataset3
```

```
cd dataset3
```

```
ln -s ../dataset2/jan2016.csv jan2016_link.csv
```

# Symlinks

- When the symlink is created you can check the file with `ls`.
- Symlinks are deleted with `unlink`

```
ls -lh jan2016_link.csv  
unlink jan2016_link.csv  
ls -l
```

# WORK!

- Download dataset3.tar.gz file
- Unpack dataset
- Make a symbolic link in dataset4/csv/jan2017.csv
- Display the file in the terminal
- Show the first 10 lines of the symbolic link file
- Show the last 5 lines of the file
- Show the size of the file



# Further reading

- The Bash Reference Manual is a reference guide to the bash shell.
  - <http://www.gnu.org/software/bash/manual/bashref.html>
- The Bash FAQ contains answers to frequently asked questions regarding bash.
  - <http://mywiki.woledge.org/BashFAQ>
- The GNU Project provides extensive documentation for its programs, which form the core of the Linux command line experience.
  - <http://www.gnu.org/manual/manual.html>