Informatics on High-throughput Sequencing Data

(Summer Course 2020)

Day 2

Agenda

- Unix-based systems.
- Why Linux!
- Let's start!
- Linux Commands for:
  - Files & Directories.
  - System.
  - Process Management.
  - Networking.
  - Compression.
  - Searching.
- Piping output.
- Wildcard character.
- Redirecting output.
- Stream Editor (Sed).
- Linux tools for text files processing.
- Shell Scripting

UNIX- based systems

- The UNIX OS was born in the late 1960s.
- AT&T Bell Labs released an operating system called Unix written in C, which allows quicker modification, acceptance, and portability.
- Examples of Unix-based OS : Linux, OS X
- All have a similar underlying system, and a similar set of command line tools.
- Linux is an operating system built by Linus Torvalds at the University of Helsinki in 1991.
- The name "Linux" comes from the Linux kernel.

Why Linux?

- Linux kernel is open source (free to install and update according to your needs).
- Many Distros customized for specific purposes such as web server, network switches, smart phone systems, etc.
- Secured and have a big community to manage its security, updates, etc.
- Servers with linux can run for years without stop (stable and reliable os)
- Majority of bioinformatics/computational biology software developed only for Linux.
- Most programs are command-line (i.e. launched by entering a command in a terminal window rather than through GUI).
- Scripting and system tools available on Linux allow customization of any analysis.

Getting Started !!

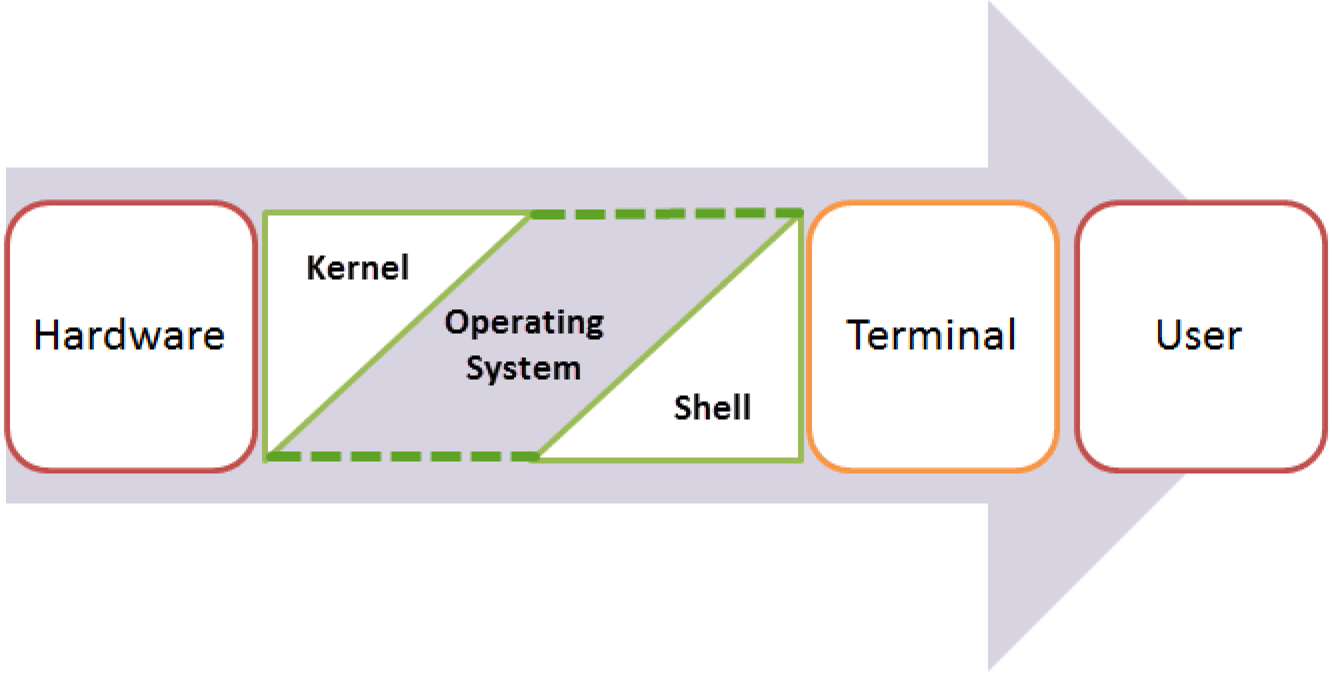
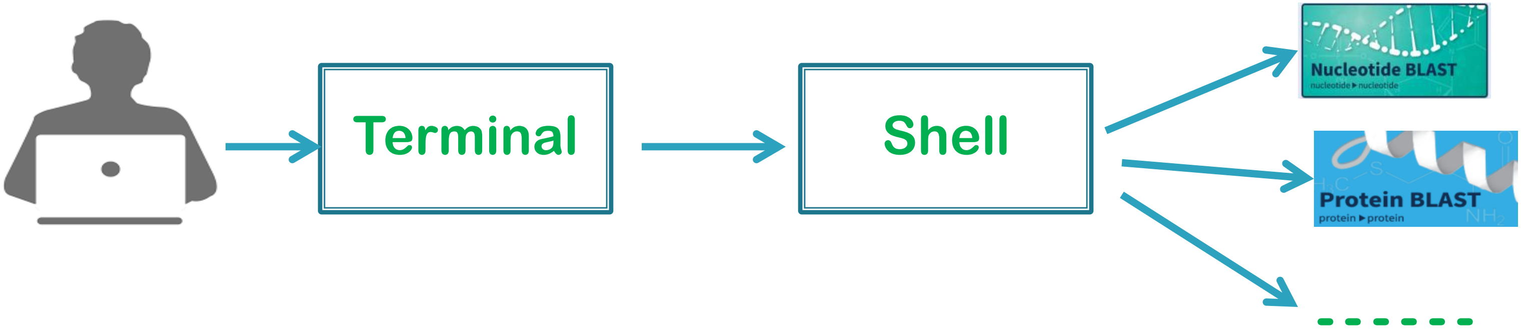


Image credits: https://www.guru99.com/introduction-to-shell-scripting.html

Getting Started !!

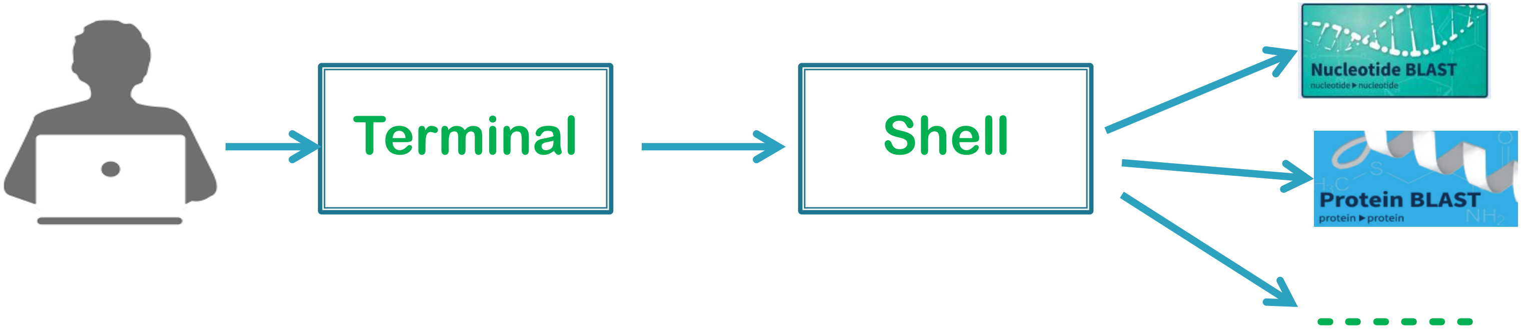


Terminal

Shell

- User communicates with Linux machine via commands typed in the terminal window.
- Commands are interpreted by a program called shell.
- Shell, an interface between Linux and User.

Getting Started !!



Terminal

Shell

- Types of shells:
  - Bourne Shell (prompt for this shell is $)
    - sh, bash
  - C shell (prompt for this shell is %)
    - csh, tcsh, zsh

Getting Started !!

Regular account

✓Perform standard tasks.
✓Access their own home directories.

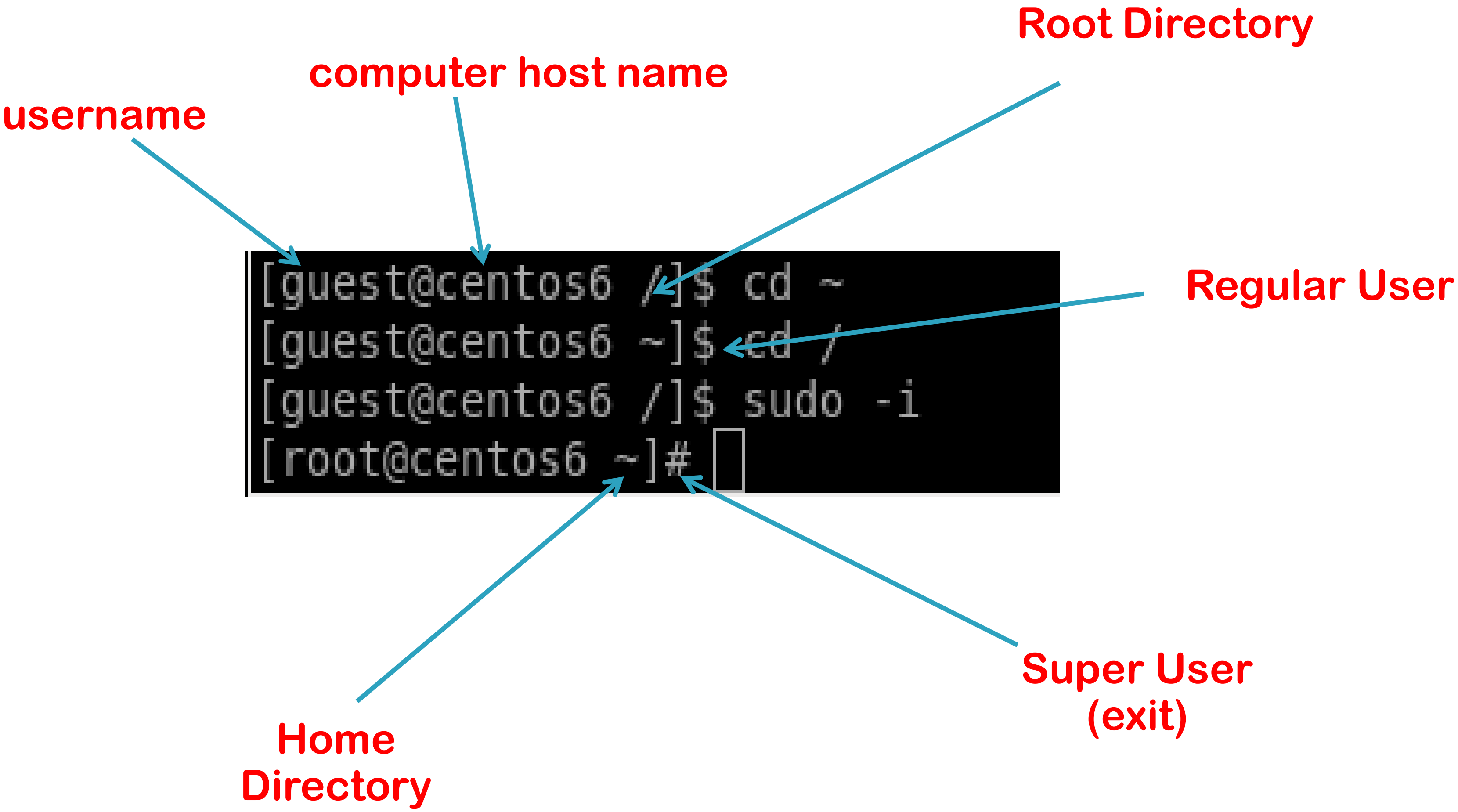
Super account (root)

✓Admin privilege
✓ created at the time of installation .

Service account

✓Allow some services such as Apache to access your computer.
✓ Exists in some Linux server editions.

Getting Started !!



```
[guest@centos6 /]$ cd ~
[guest@centos6 ~]$ cd /
[guest@centos6 /]$ sudo -i
[root@centos6 ~]#
```

username

computer host name

Root Directory

Regular User

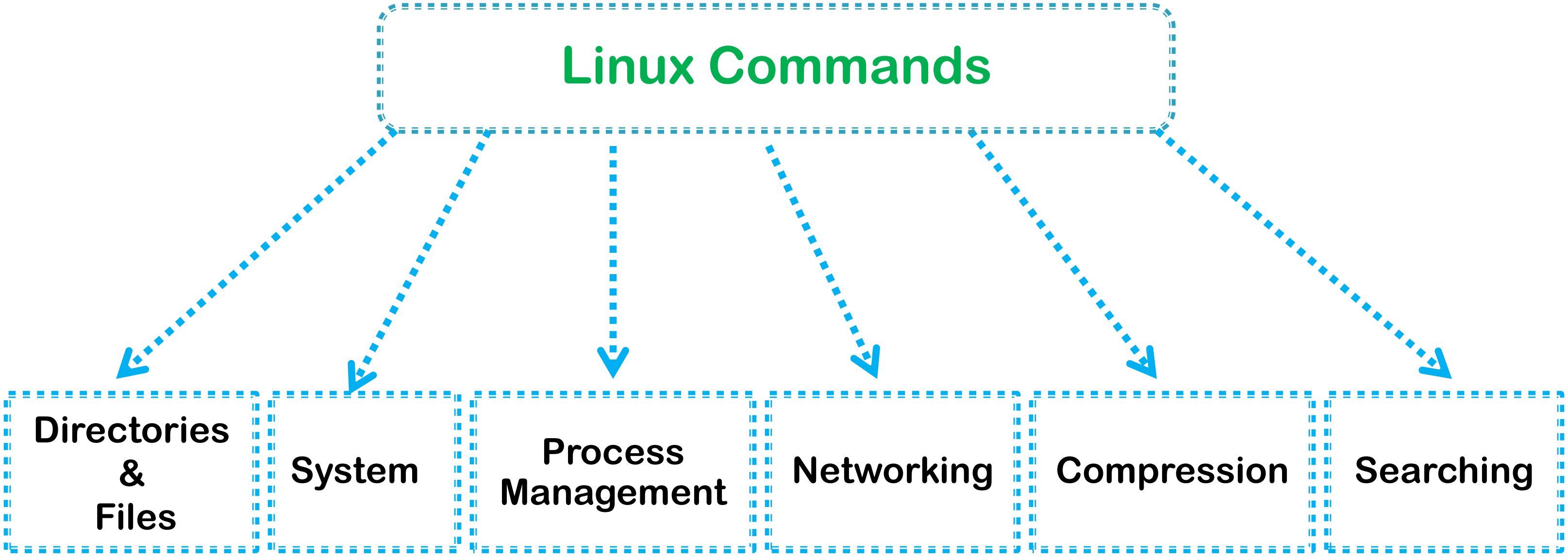
Home Directory

Super User (exit)

Why CLI?

- Flexible (more options, many functions with one command).
- Load and execute is very fast (does not consume space from system memory).

Getting Started !!



Linux Commands

Directories & Files

System

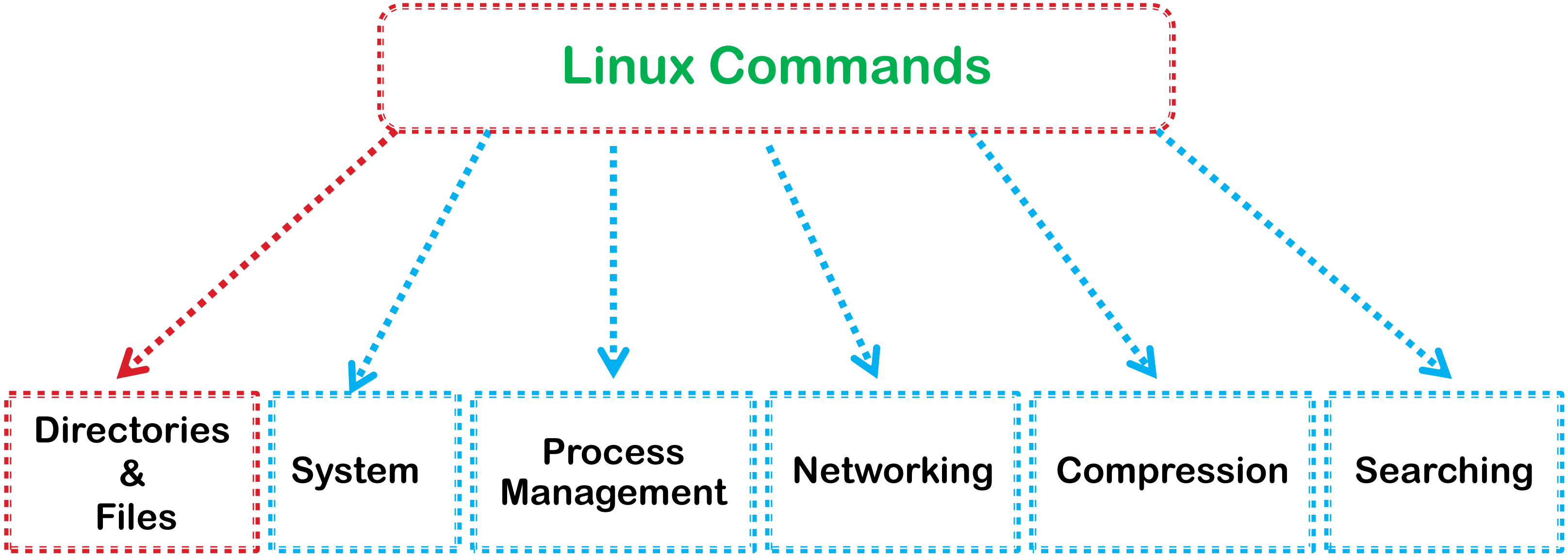
Process Management

Networking

Compression

Searching

Getting Started !!



Linux Commands

Directories & Files

System

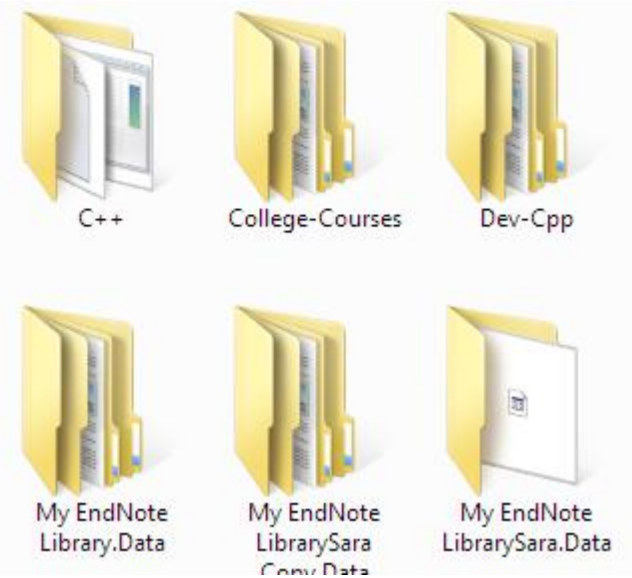
Process Management

Networking

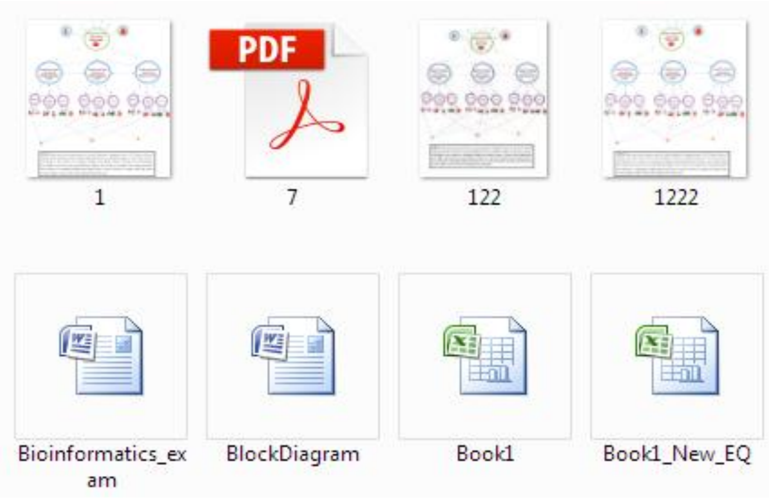
Compression

Searching

Directories and Files

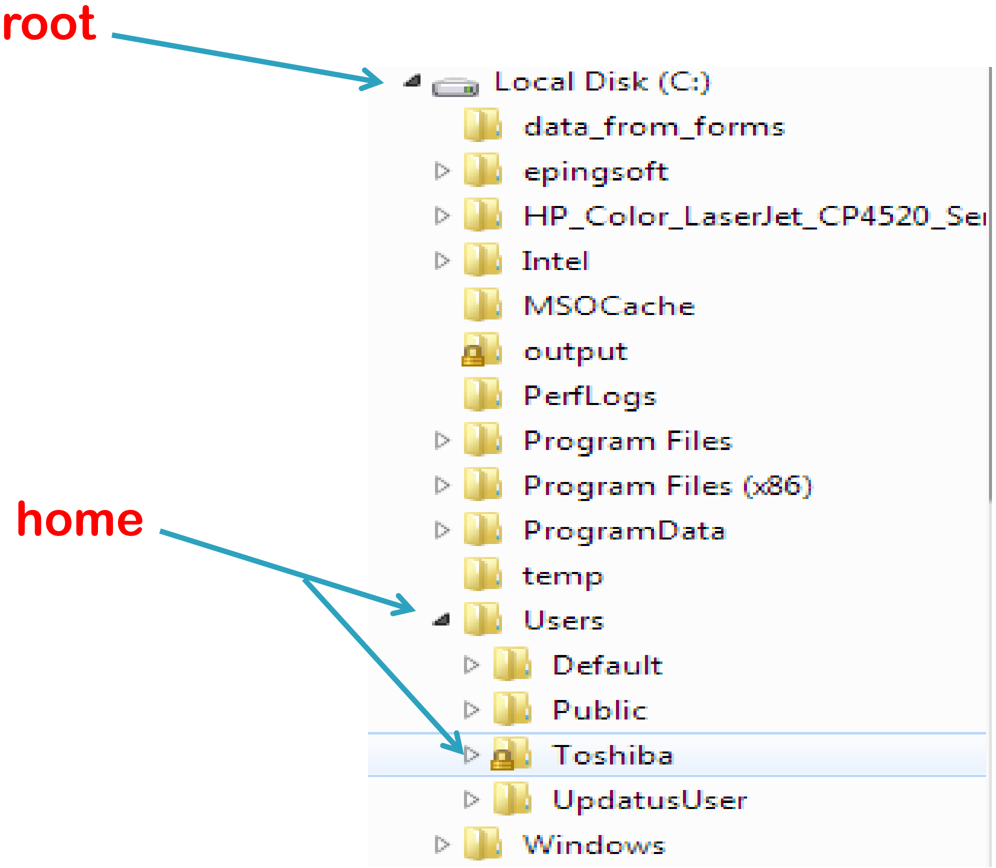


Directories (Folders)



Files

Directories



root

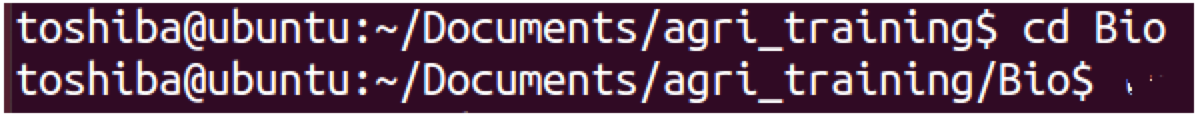
home

```
Local Disk (C:)
    data_from_forms
    epingsoft
    HP_Color_LaserJet_CP4520_Se
    Intel
    MSOCache
    output
    PerfLogs
    Program Files
    Program Files (x86)
    ProgramData
    temp
    Users
        Default
        Public
        Toshiba
        UpdatusUser
    Windows
```

Directories (Traverse)

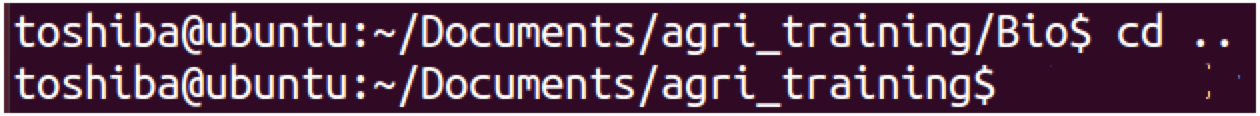
`cd Bio`	Change directory to Bio.
`cd ..`	Go up one directory.
`cd /`	Go to the root directory.
`cd ~`	Go to your home directory.
`cd -`	Go to the last directory you were just in.
`pwd`	Print current working directory.

Directories (Traverse)



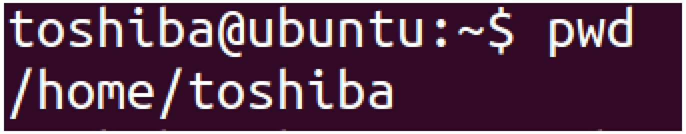
```
toshiba@ubuntu:~/Documents/agri_training$ cd Bio
toshiba@ubuntu:~/Documents/agri_training/Bio$
```

cd Bio



```
toshiba@ubuntu:~/Documents/agri_training/Bio$ cd ..
toshiba@ubuntu:~/Documents/agri_training$
```

cd ..



```
toshiba@ubuntu:~$ pwd
/home/toshiba
```

pwd

Notes

- Both su and sudo are used to run commands with root permissions.
  - The su command switches to the super user – or root user – . You'll have to enter the root account's password.
  - Once you're done running commands in the root shell, you should type exit to leave the root shell and go back to limited-privileges mode.

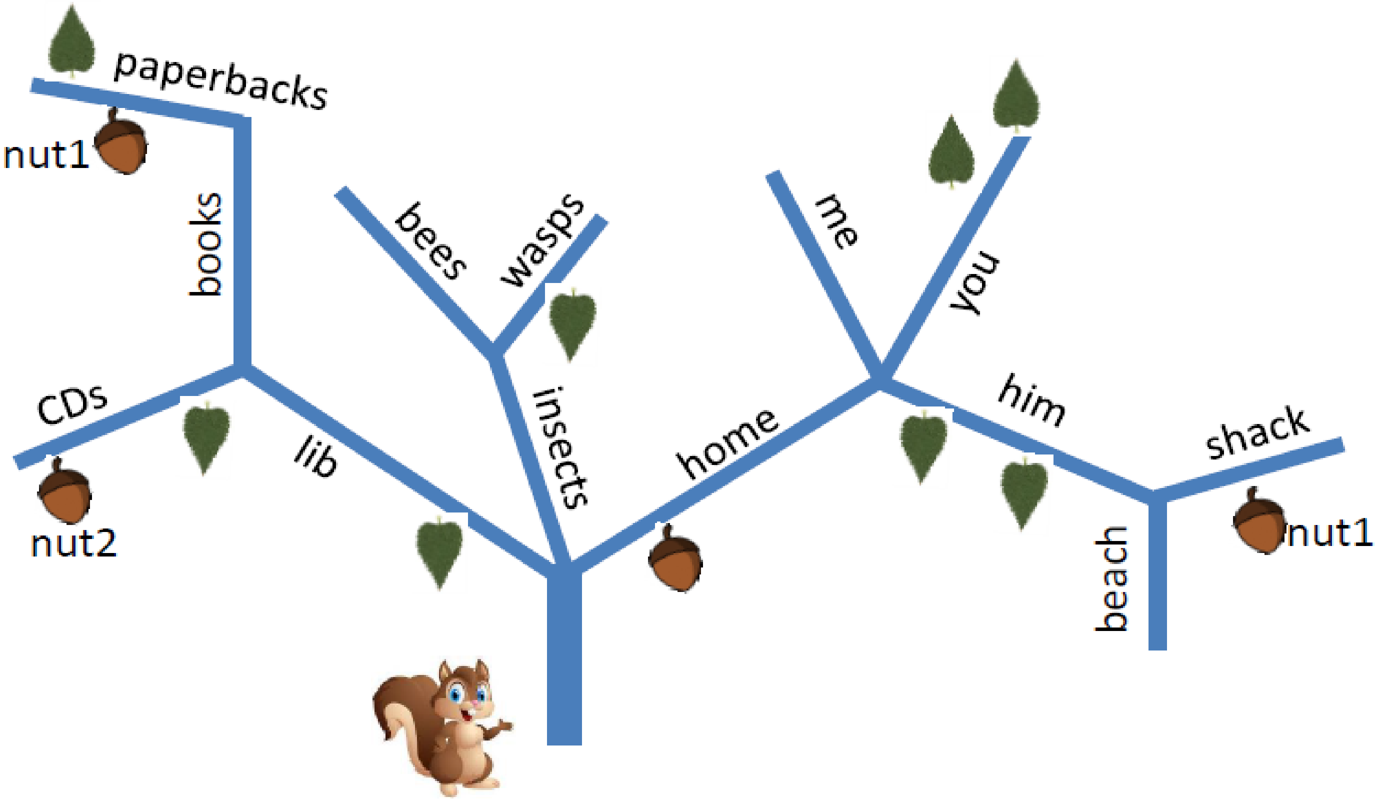
https://www.howtogeek.com/111479/htg-explains-whats-the-difference-between-sudo-su/

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

Braches = directories

leaves, nuts = files



Direct squirrel to **nut1** (on the right) using commands:

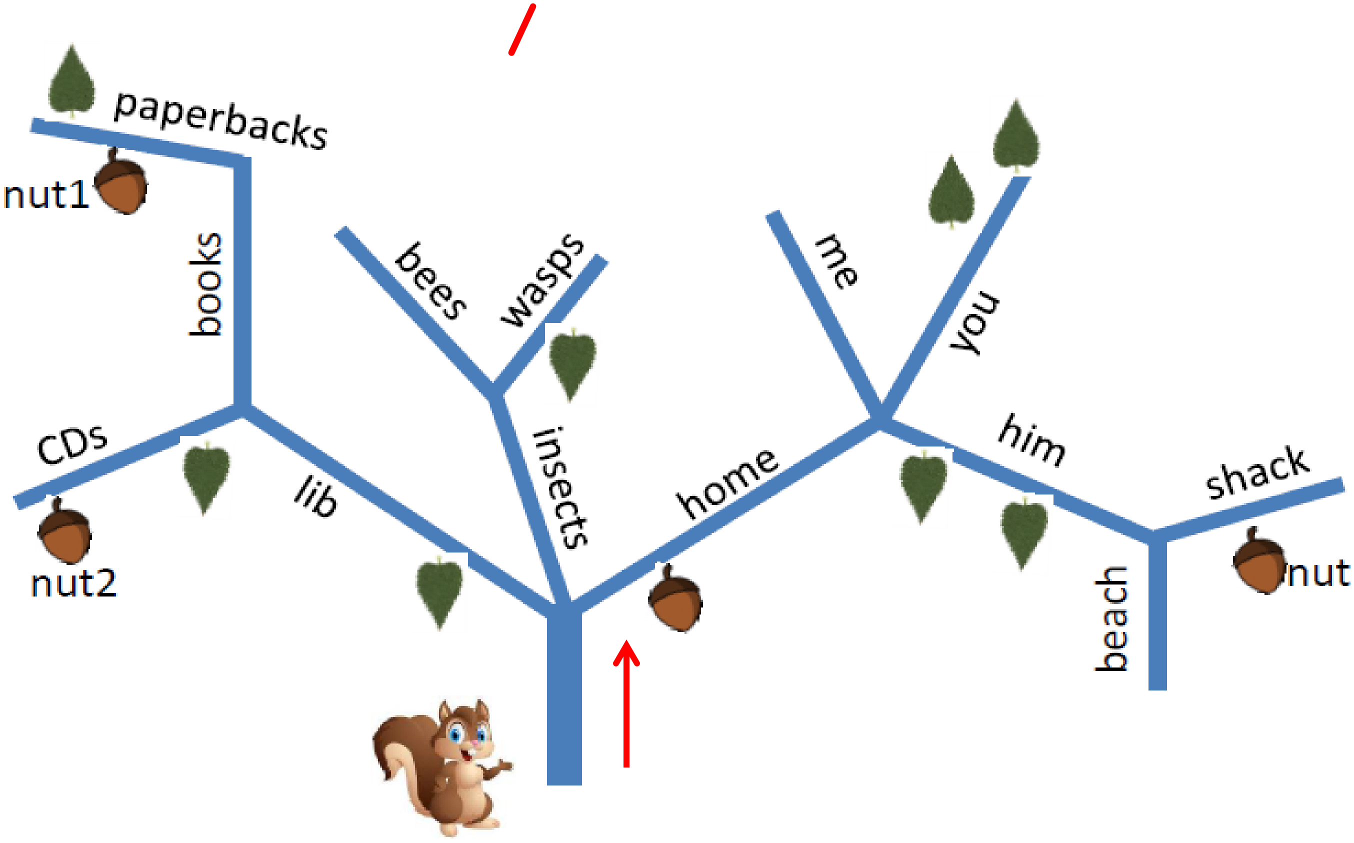
`/`	get on the main trunk (referred to as **root**)
`some_name/`	from where you are, turn into branch "**some_name**"
`../`	return to the previous branch (closer to root)
`./`	stay where you are

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

Braches = directories

leaves, nuts = files



Direct squirrel to **nut1** (on the right) using commands:

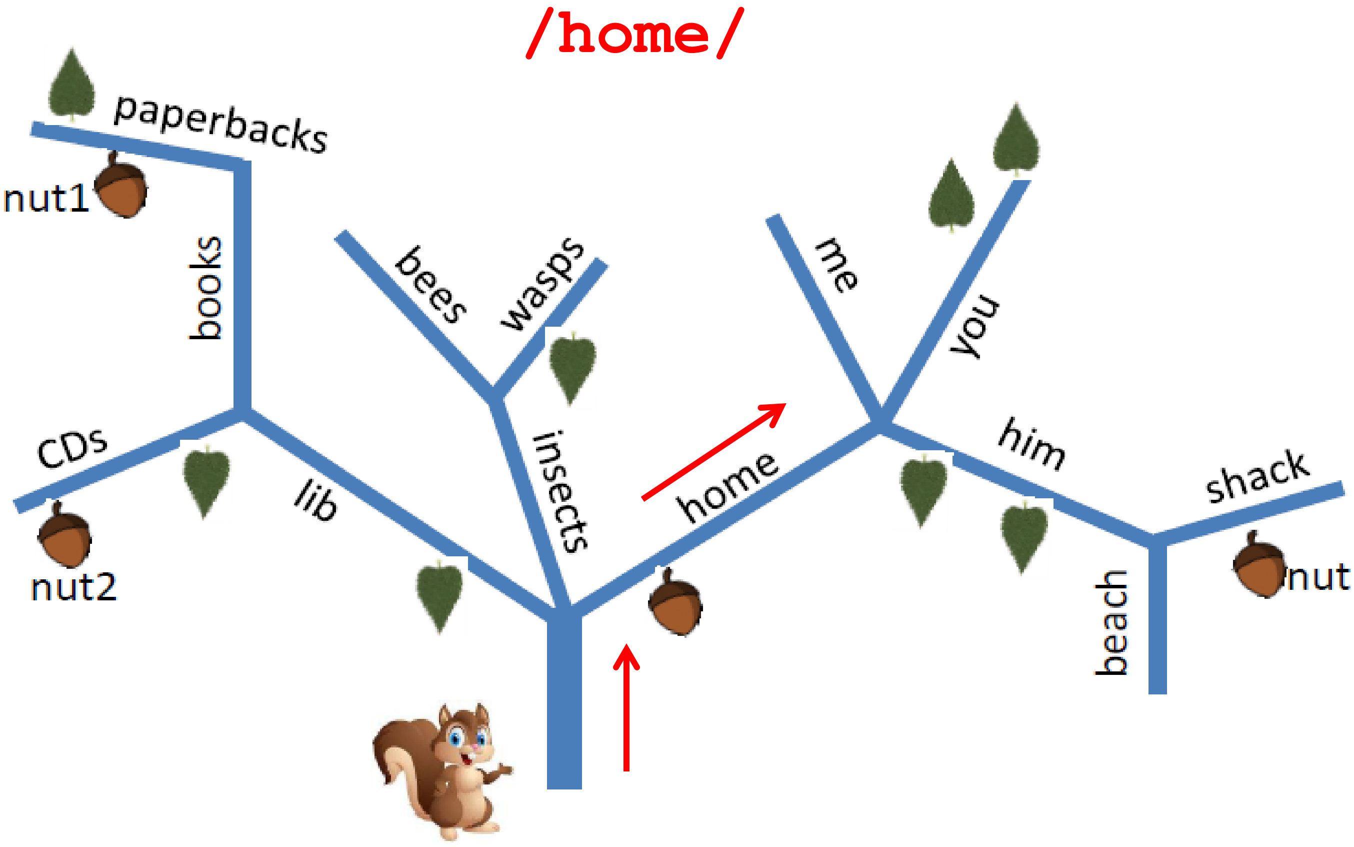
/	get on the main trunk (referred to as **root**)
some_name/	from where you are, turn into branch "**some_name**"
../	return to the previous branch (closer to root)
./	stay where you are

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

Braches = directories

leaves, nuts = files



Direct squirrel to **nut1** (on the right) using commands:

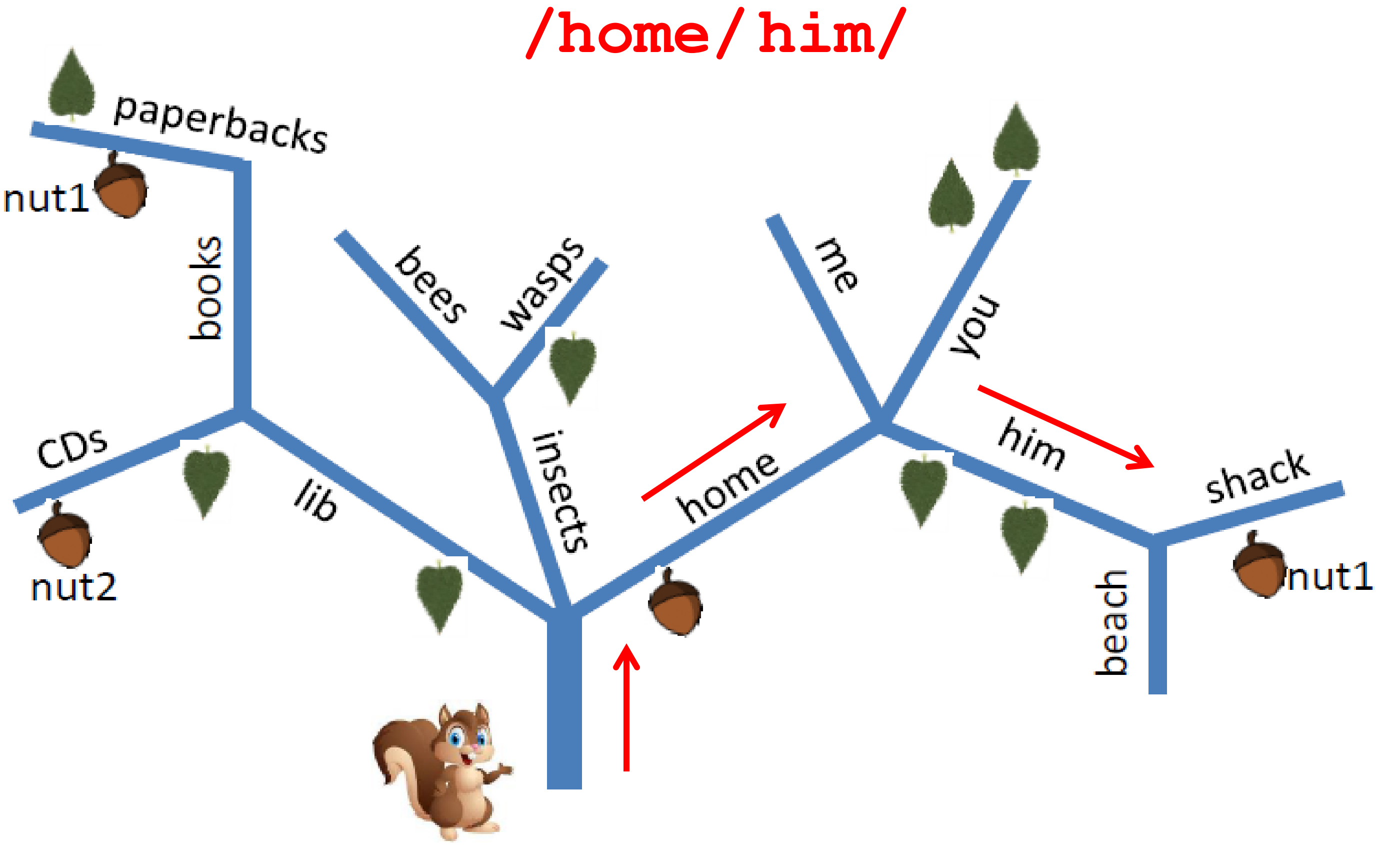
`/`	get on the main trunk (referred to as **root**)
`some_name/`	from where you are, turn into branch "**some_name**"
`../`	return to the previous branch (closer to root)
`./`	stay where you are

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

Braches = directories

leaves, nuts = files



Direct squirrel to **nut1** (on the right) using commands:

`/`	get on the main trunk (referred to as **root**)
`some_name/`	from where you are, turn into branch "`some_name`"
`../`	return to the previous branch (closer to root)
`./`	stay where you are

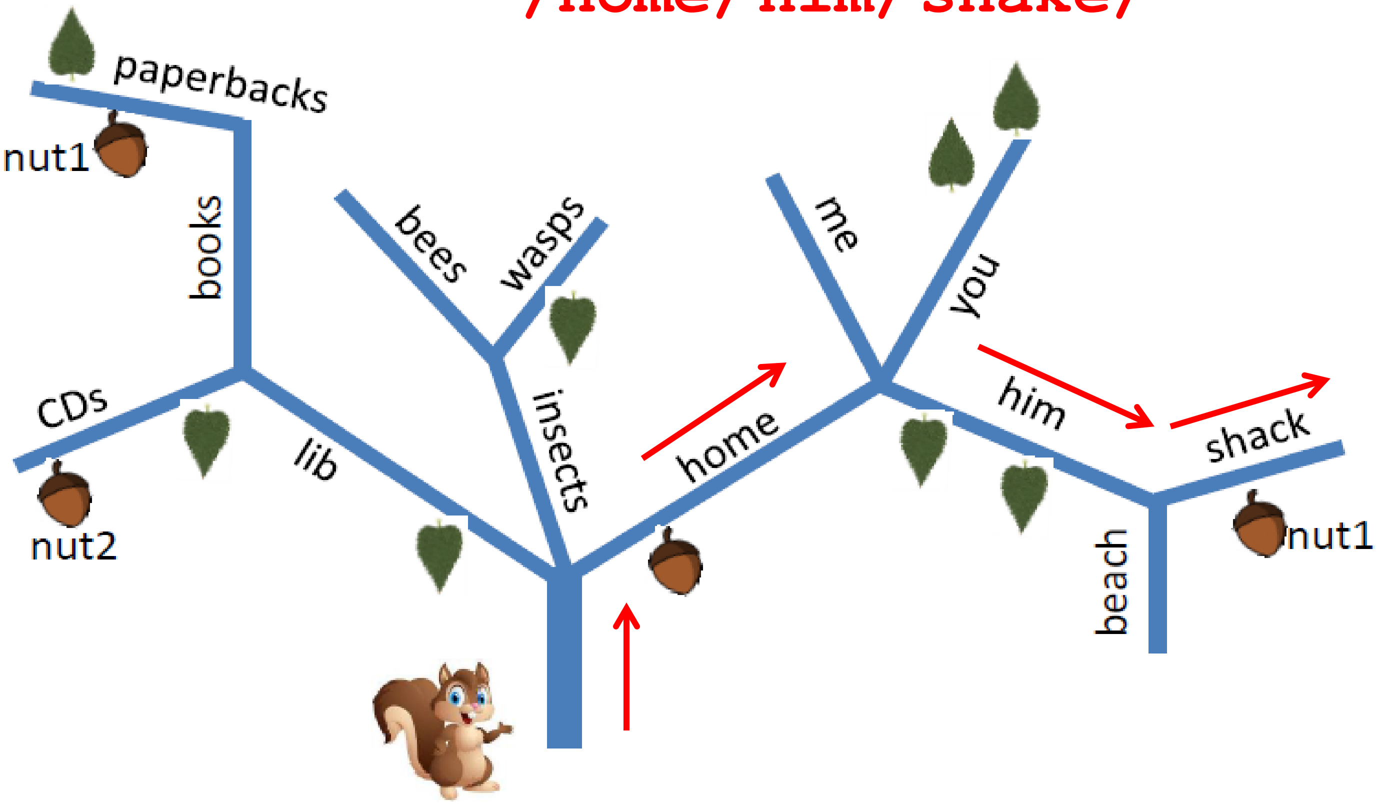
Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

Braches =
directories

leaves, nuts
= files

/home/ him/ shake/



Direct squirrel to **nut1** (on the right) using commands:

`/`	get on the main trunk (referred to as **root**)
`some_name/`	from where you are, turn into branch "**some_name**"
`../`	return to the previous branch (closer to root)
`./`	stay where you are

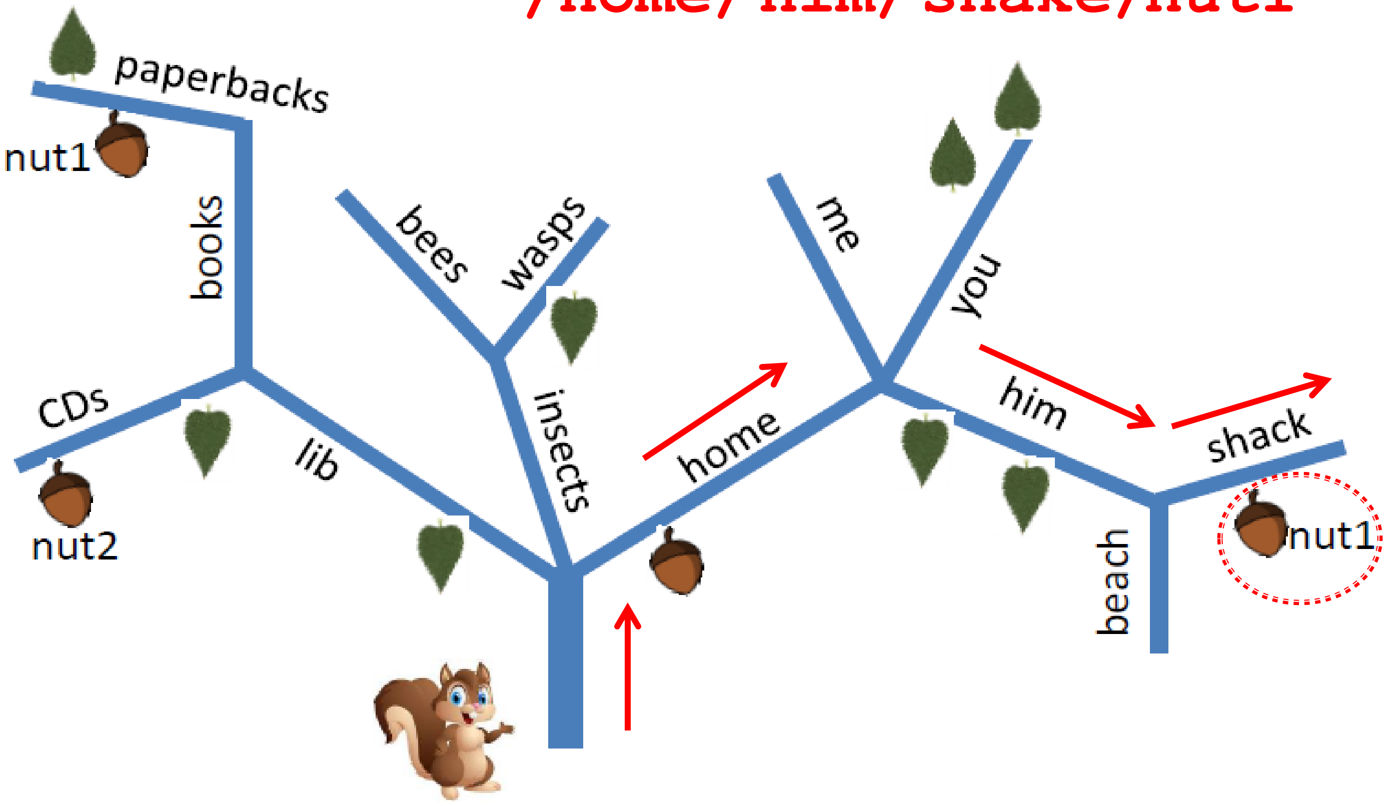
Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

Braches = directories

leaves, nuts = files

/home/ him/ shake/nut1



Direct squirrel to **nut1** (on the right) using commands:

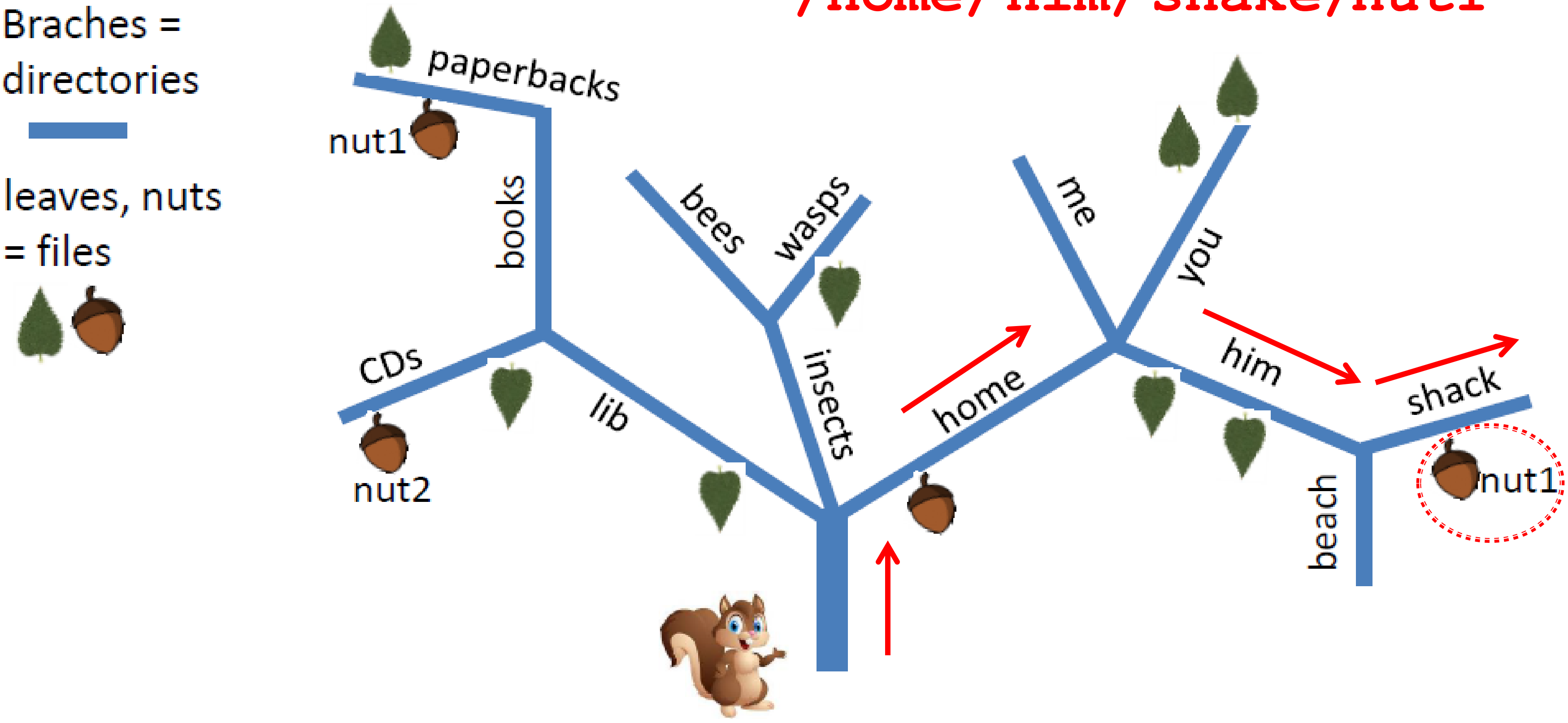
`/`	get on the main trunk (referred to as **root**)
`some_name/`	from where you are, turn into branch "`some_name`"
`../`	return to the previous branch (closer to root)
`./`	stay where you are

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

Braches = directories

leaves, nuts = files

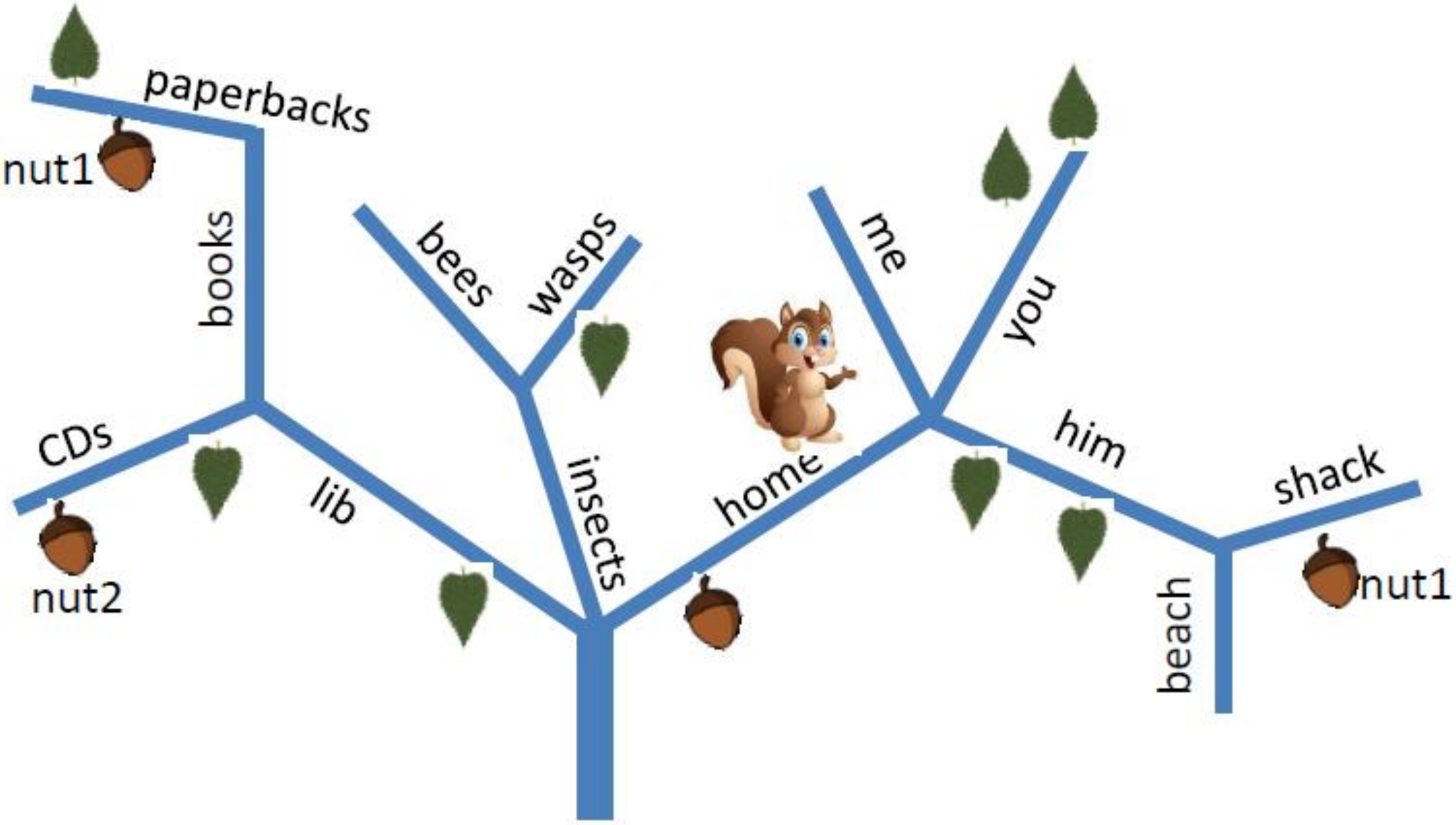


/home/him/shake/nut1

This is called absolute path starting from the root (i.e. the trunk)

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

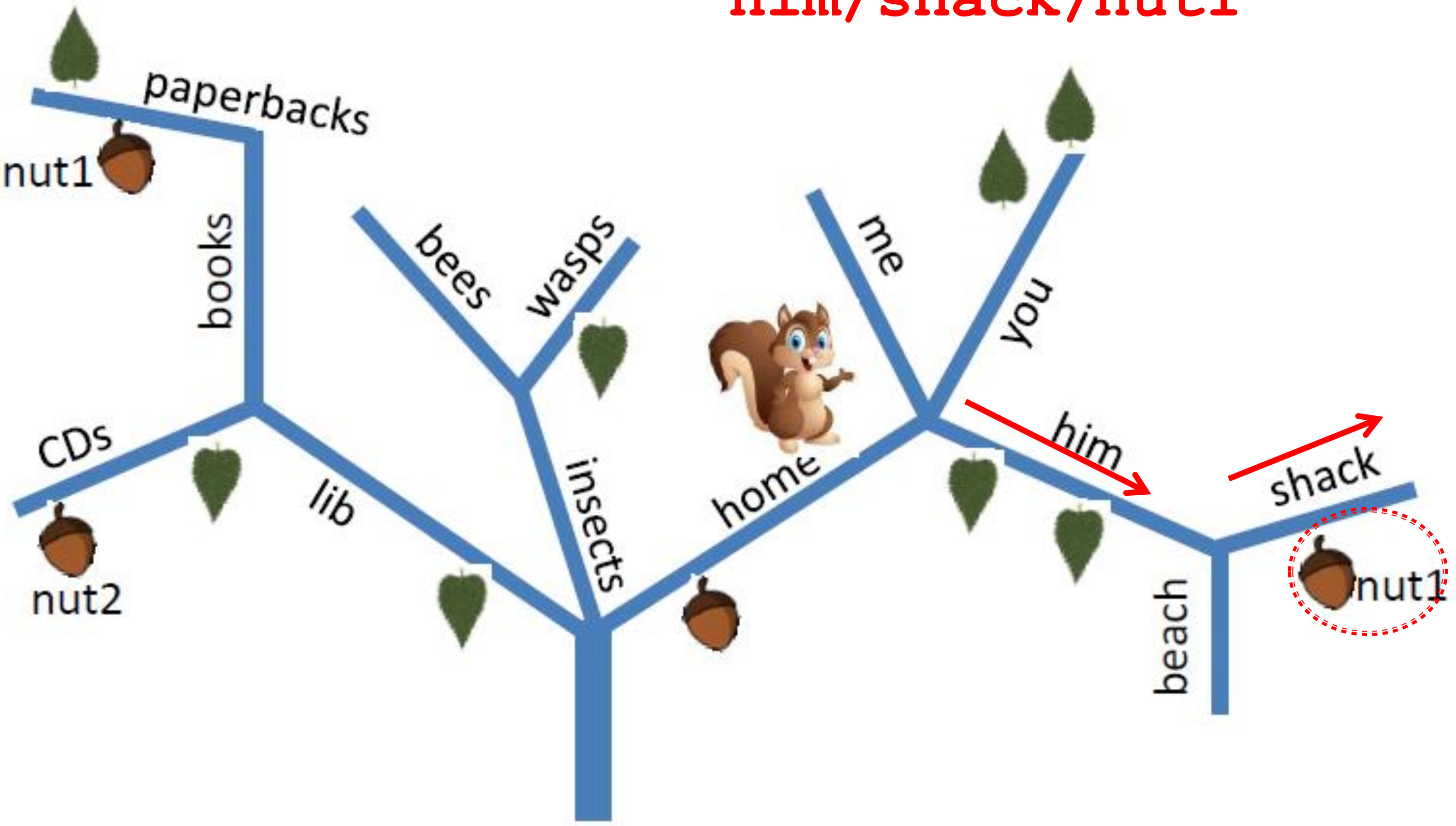


Assume squirrel sitting on home.

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

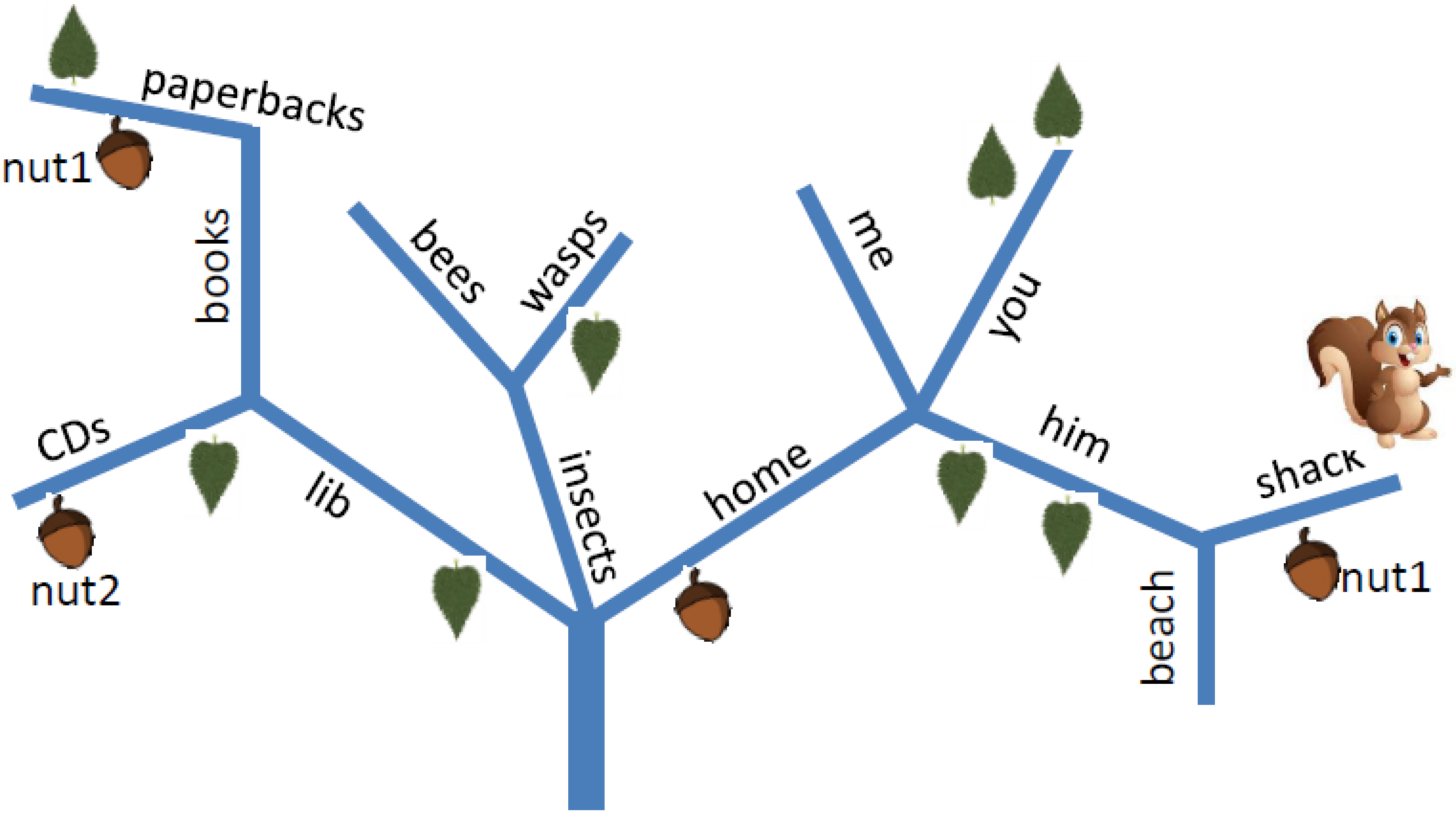
him/shack/nut1



This is called relative path (starting from “where we are”).

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree



Assume squirrel sitting on shack

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

```
nut1
./nut1
```

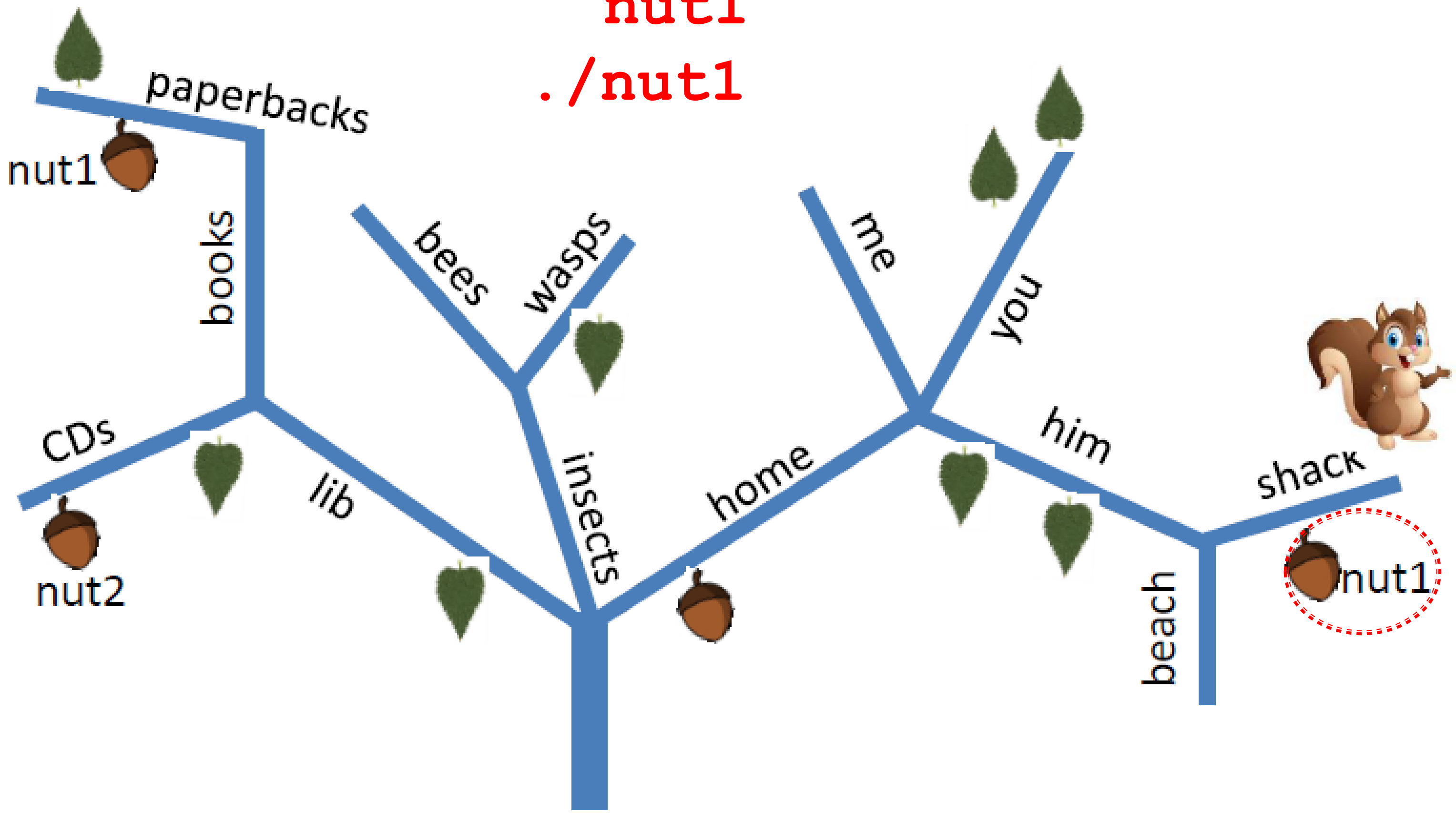
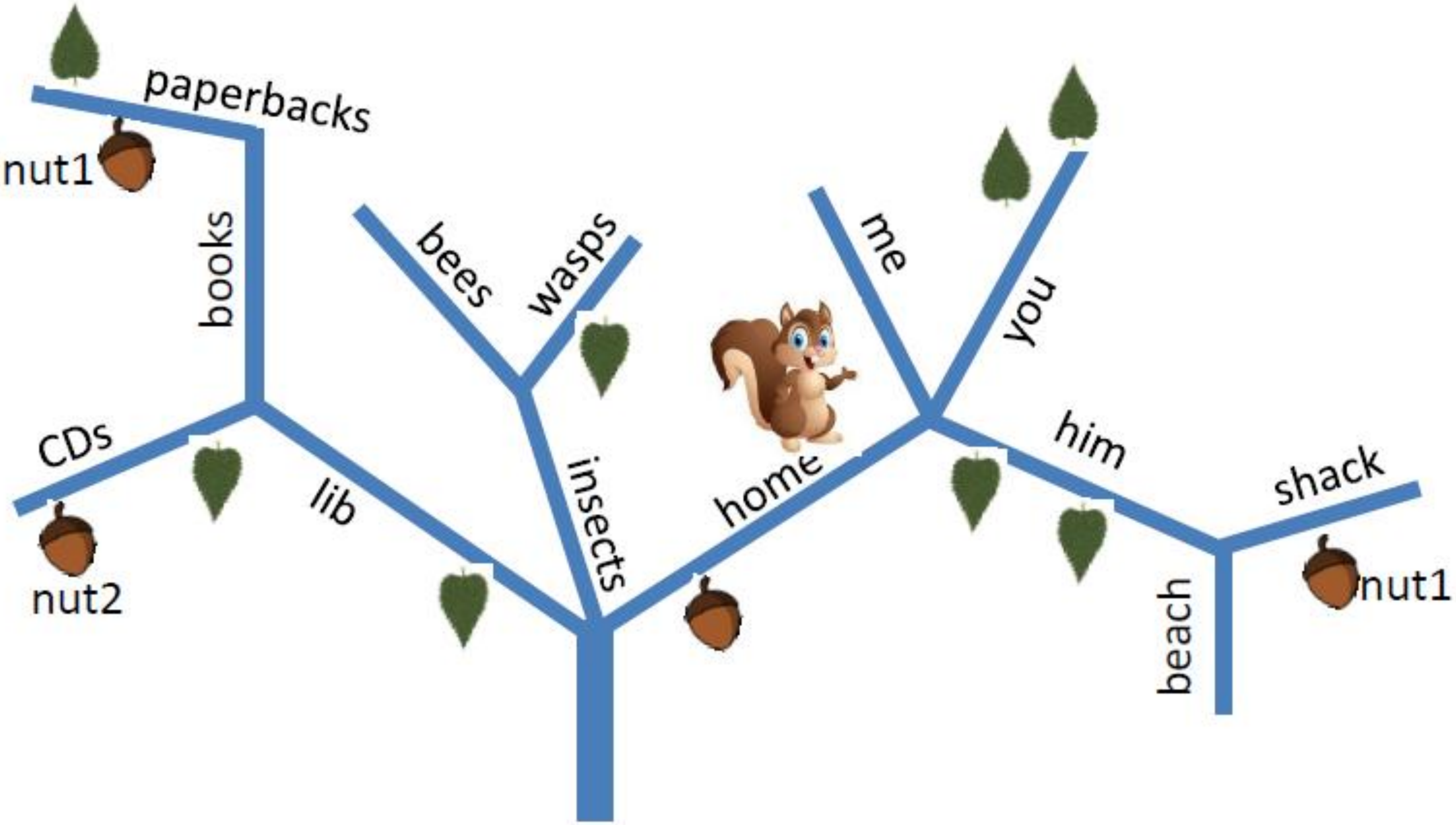


Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree



Assume squirrel sitting on home, we would like to make him jump to the ground.

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

../

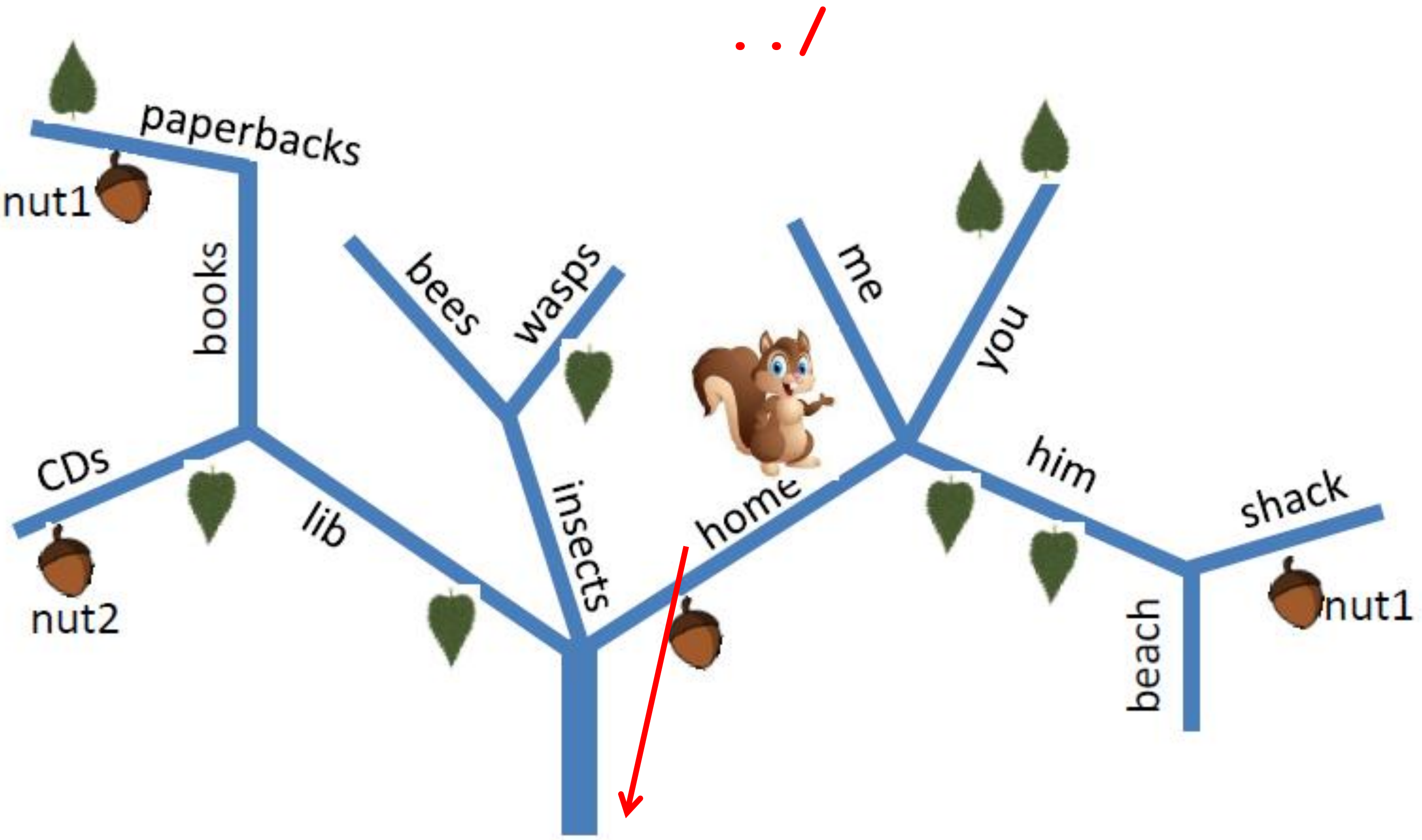
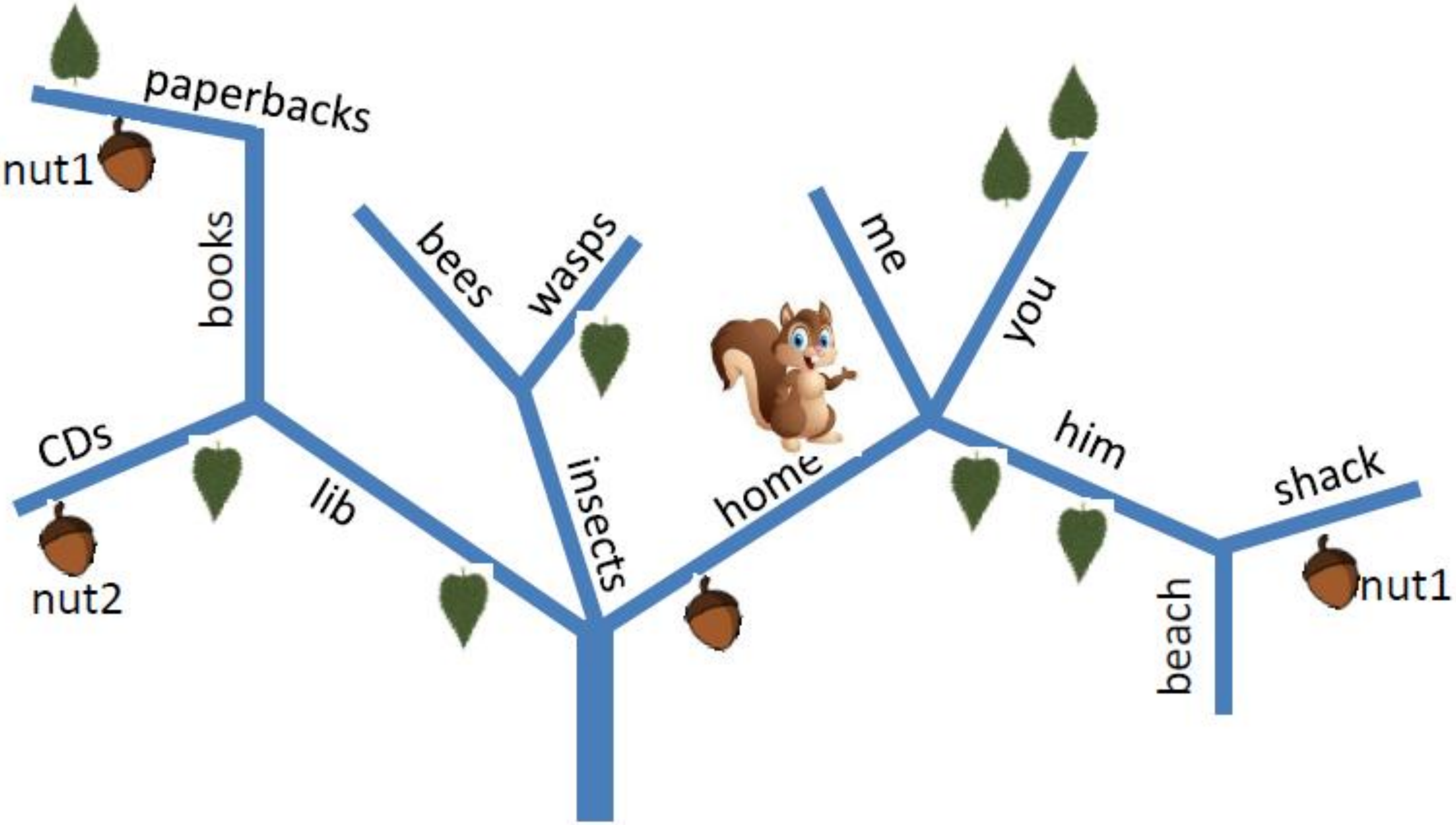


Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree



Assume squirrel sitting on home, we would like to make him stay there.

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Linux directory tree

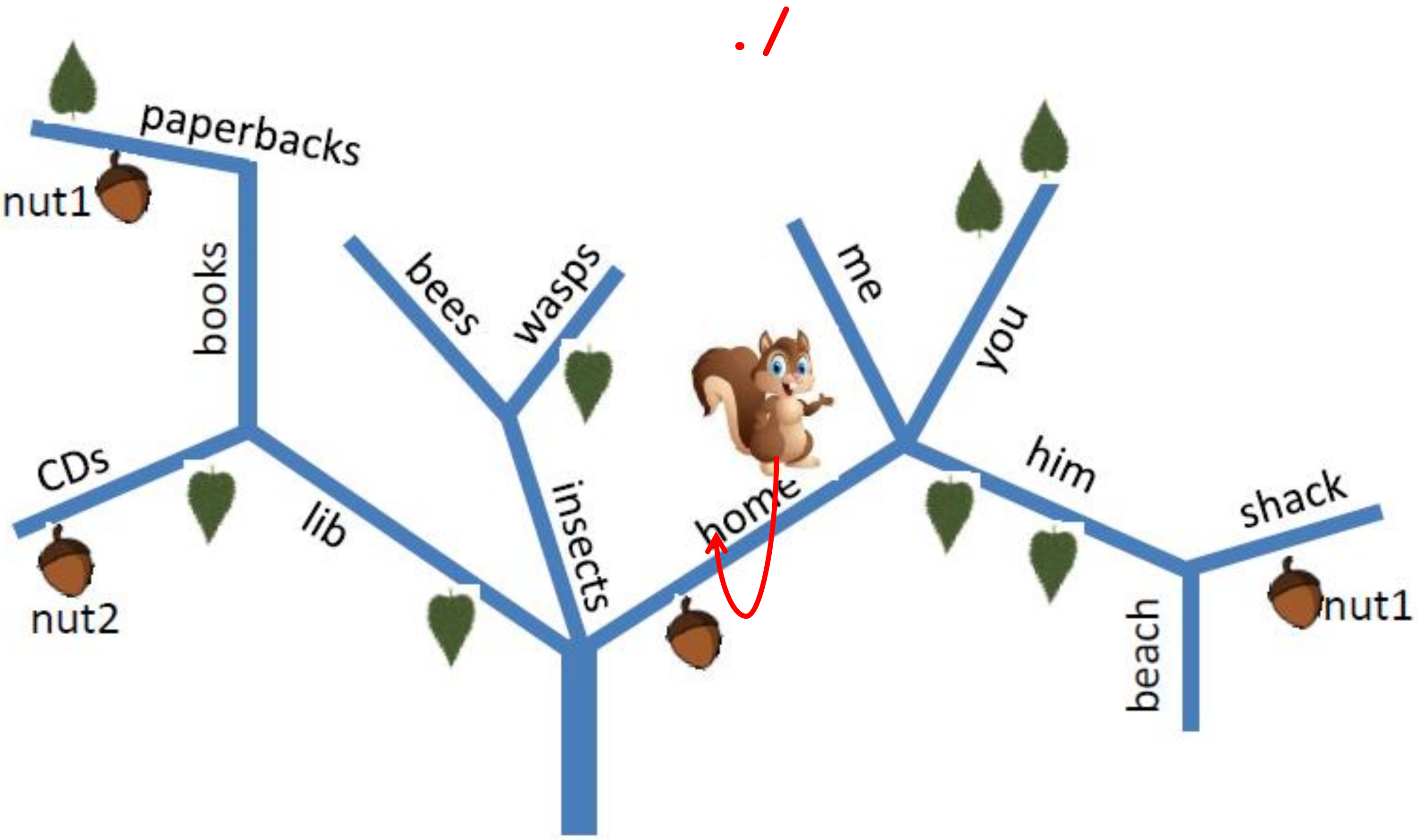
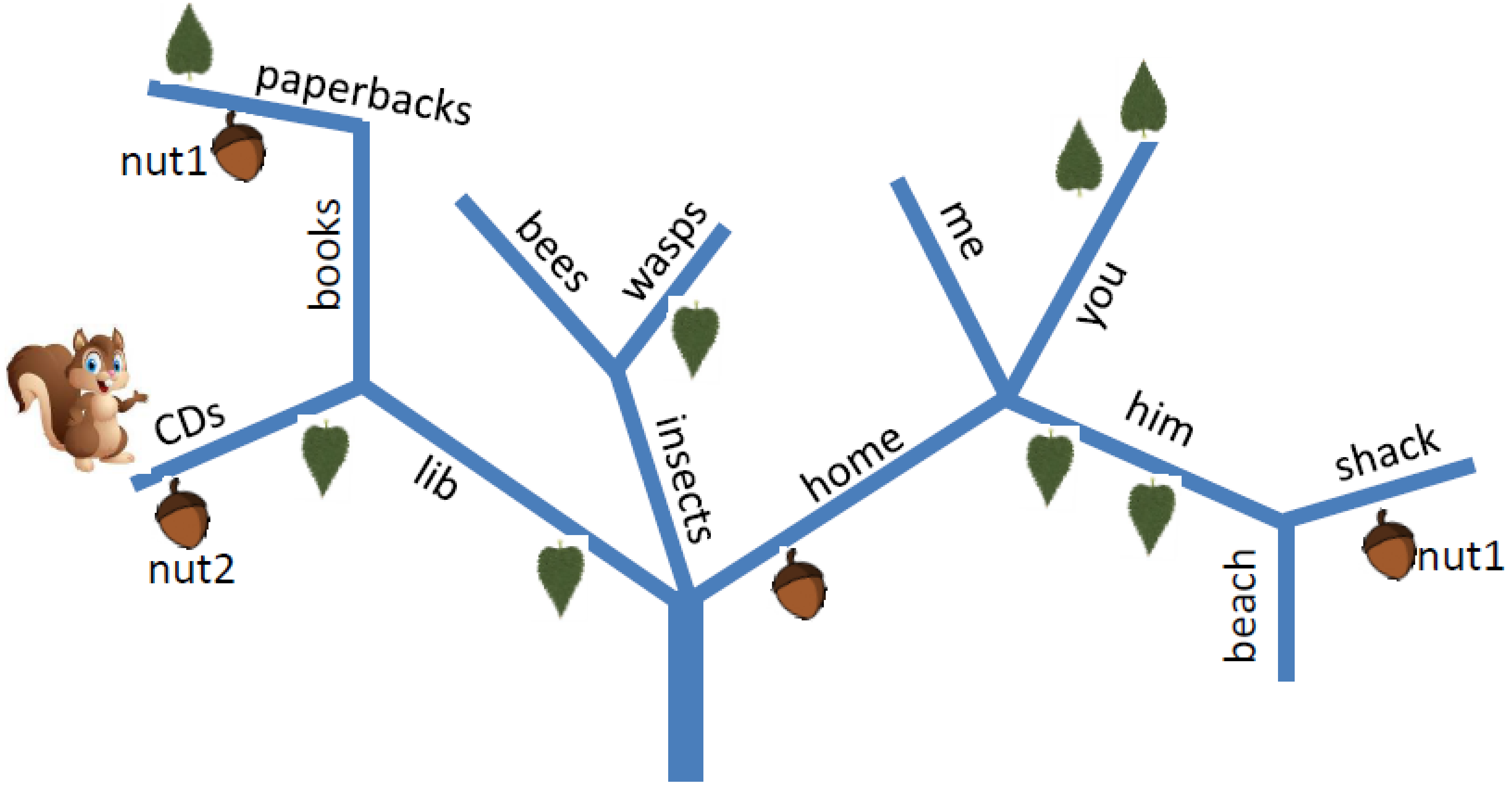


Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

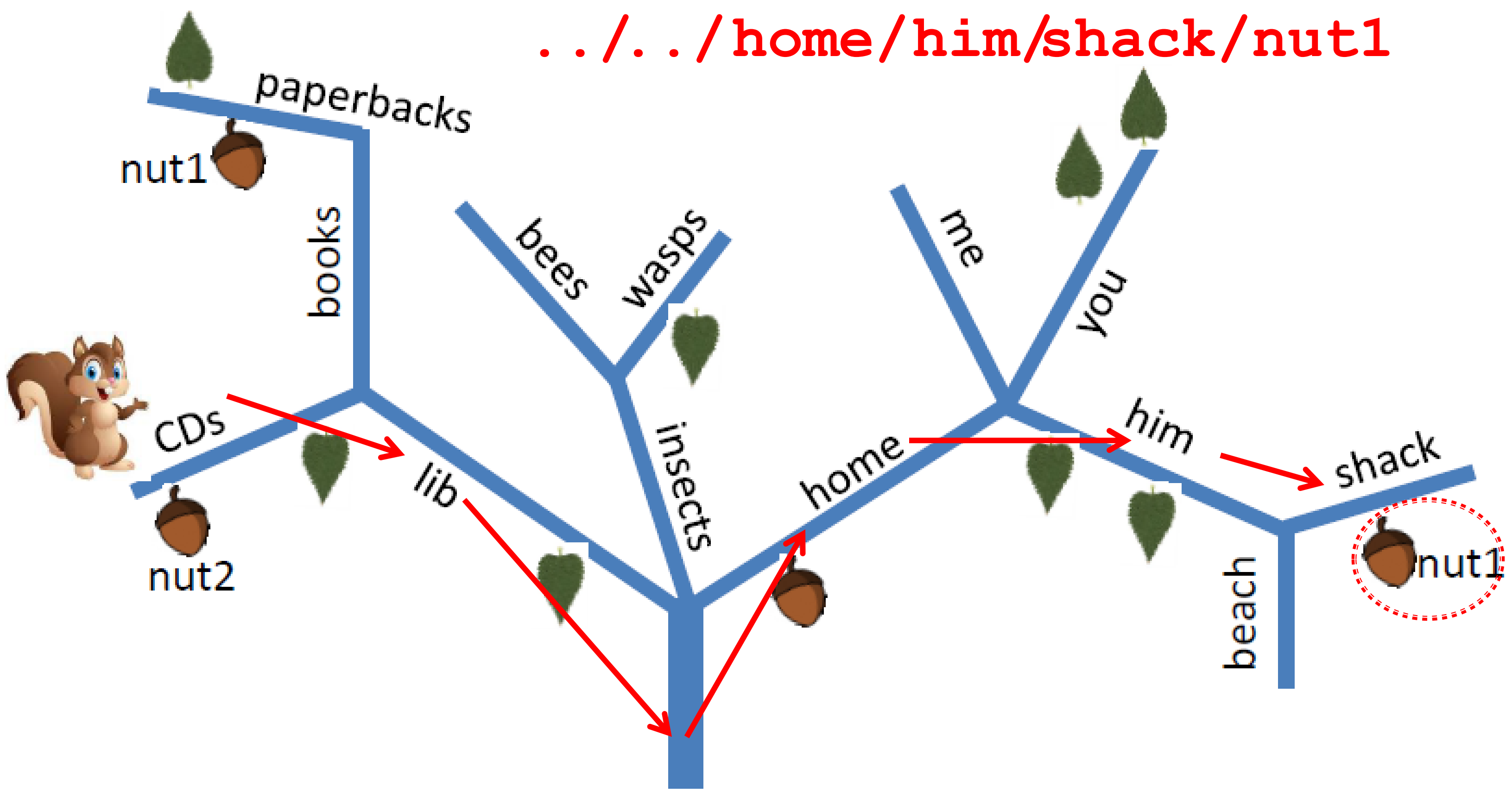
Linux directory tree



Assume squirrel sitting on CDs, we would like to direct him to nut1 on the right.

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

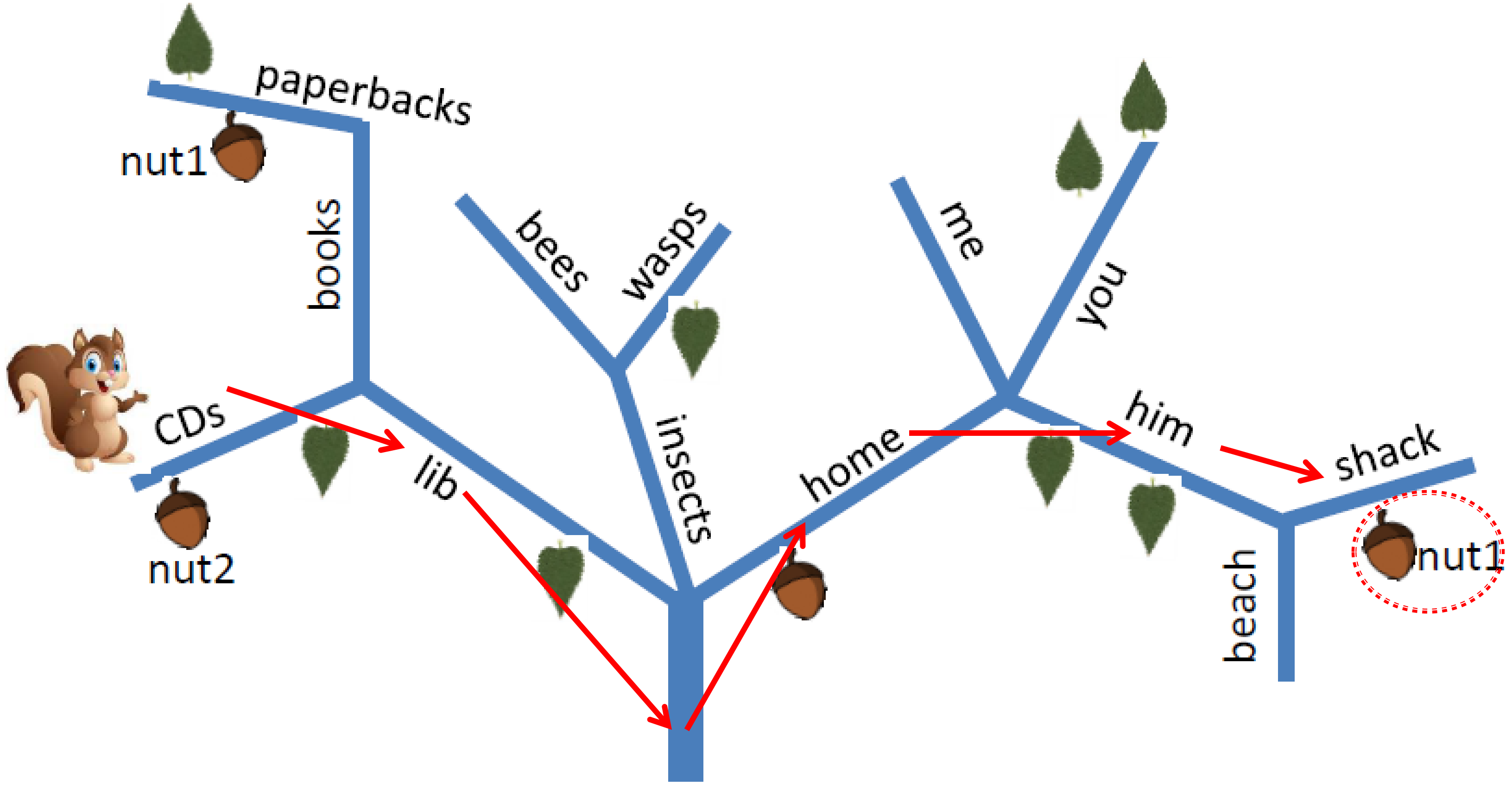
Linux directory tree



../../home/him/shack/nut1

Photo credit: Robert Bukowski, Linux for Biologists- Part 1.

Traversing directory tree



Directories (create, remove & copy)

`mkdir Bio`	Make directory Bio.
`rm -rf Bio`	Remove directory Bio recursively (force remove).
`rm -ri Bio`	Remove directory Bio (prompt).
`rmdir Bio`	Remove directory Bio.
`cp -r Bio1 Bio2`	Copy directory Bio1 to Bio2 recursively.

Directories (listing)

`ls`	List items in current directory.
`ls Bio`	List all items in directory Bio.
`ls -l`	List items in current directory and show in long format to see permissions, size, and modification date.
`ls -a`	List all items in current directory, including hidden files.
`ls -F`	List all items in current directory and show directories with a slash and executables with a star

Directories (listing)

`ls -t`	sort by time & date.
`ls -S`	sort by file size.
`ls -la`	list long format including hidden files
`man ls`	ask for help!, q for quite

Thanks!

// | ?