

Practical Computing for Scientists

Armin Sobhani
CSCI 2000U
UOIT – Fall 2015

Checkpoint 2



- Please complete the *What is Science Survey Results*:
 - Blackboard > Course Content > Week 1 (Sept. 14-18) > Wednesday Sept. 14 > Checkpoint 2



The Unix Shell

Files and Directories

Created by Greg Wilson

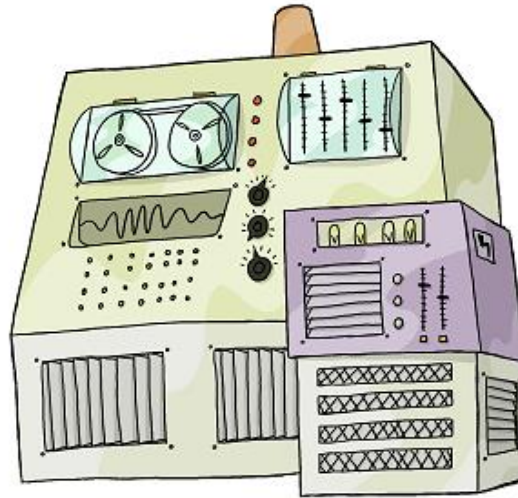


Copyright © Software Carpentry

This work is licensed under the Creative Commons Attribution License

See <http://software-carpentry.org/license.html> for more information.



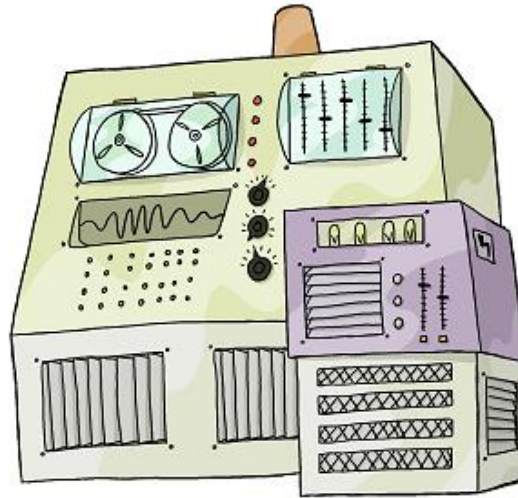


Run
Programs

Store
Data

Communicate
with each other

Interact
with us

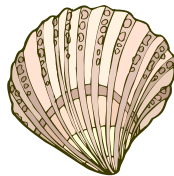


Run

Store

Programs

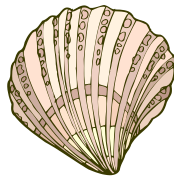
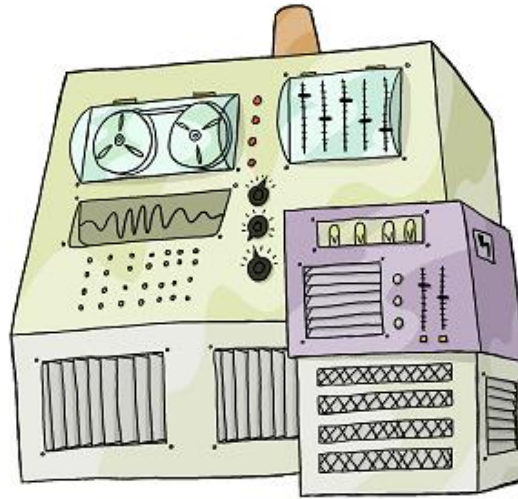
Data



shell

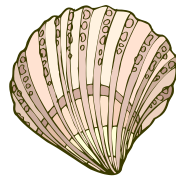
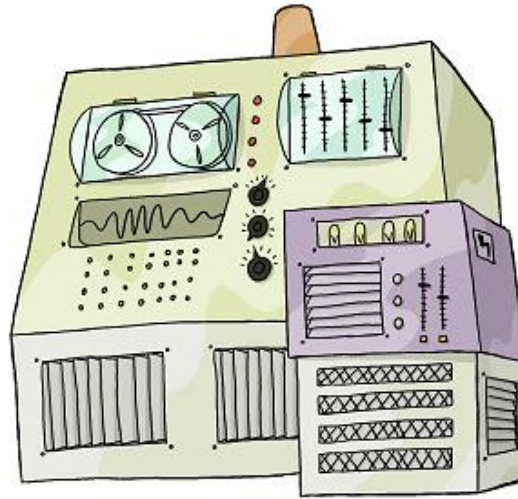
Communicate
with each other

Interact
with us



shell

Store
Data



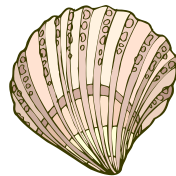
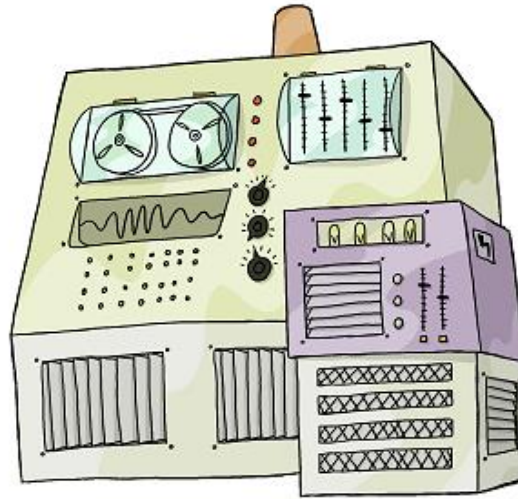
shell

Store

Data

file system





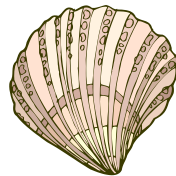
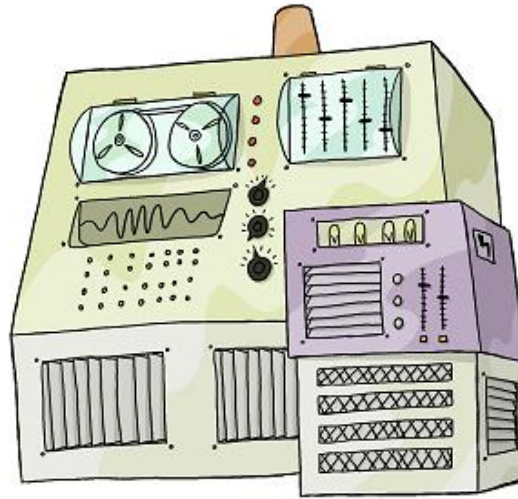
shell

Store

Data

file system

files



shell

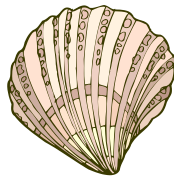
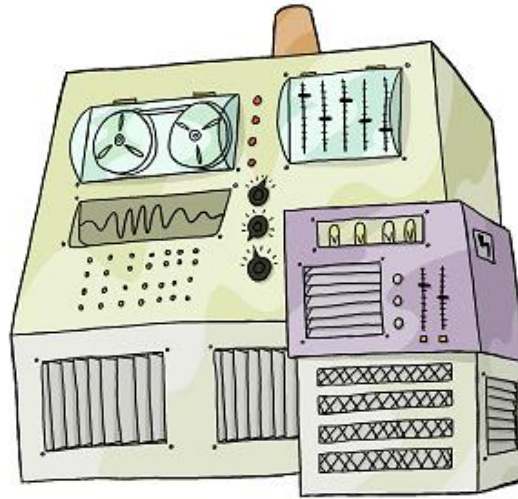
Store

Data

file system

files

directories



shell

Store

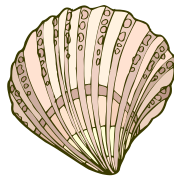
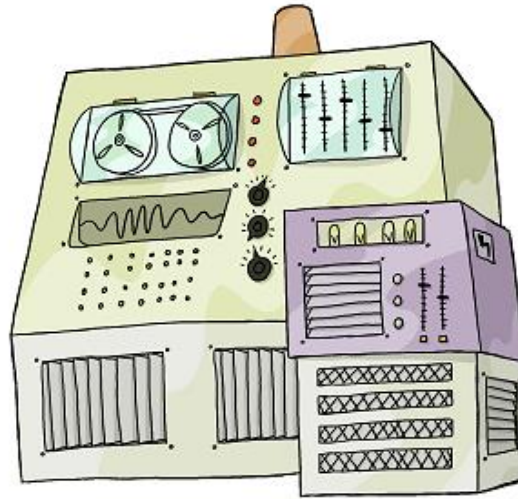
Data

file system

files

directories

Use the shell
to view and change
the file system



Use the shell
to run commands
to view what's in
the file system

shell

Store

Data

file system

files

directories

login:

login: ← computer prompt in **bold**

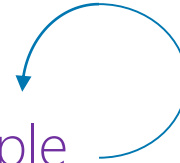
login:



computer prompt in **bold**



explanatory text in purple



login: `nelle` ← user input in green

login: nelle

password: **** ← password

login: nelle

password: ****

\$ ← shell prompt

login: nelle

password: ****

\$ ← shell prompt
like Python's >>> and ...

login: nelle

password: ****

\$ whoami ← check user ID

login: nelle

password: ****

\$ whoami ←

check user ID

shell finds the whoami program

login: nelle

password: ****

\$ whoami



check user ID

shell finds the whoami program

runs it

login: `nelle`

password: `*****`

\$ `whoami`

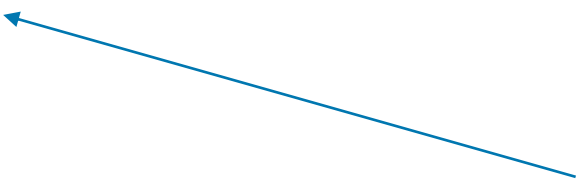
`nelle`

check user ID

shell finds the `whoami` program

runs it

prints its output



login: `nelle`

password: `*****`

\$ `whoami`

`nelle`

\$

check user ID

shell finds the `whoami` program

runs it

prints its output

displays a new prompt

login: nelle

password: ****

\$ whoami

nelle

\$ pwd ← what is the *working directory*

login: nelle

password: ****

\$ whoami

nelle

\$ pwd



what is the *working directory*
the directory used when no other
directory is explicitly specified

login: `nelle`

password: `*****`

\$ `whoami`

`nelle`

\$ `pwd`

`/users/nelle`

\$

login: `nelle`

password: `*****`

\$ `whoami`

`nelle`

\$ `pwd`

`/users/nelle`

\$



login: `nelle`

password: `*****`

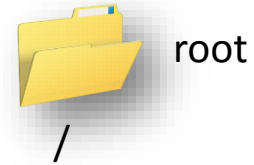
\$ `whoami`

`nelle`

\$ `pwd`

`/users/nelle`

\$



login: `nelle`

password: `*****`

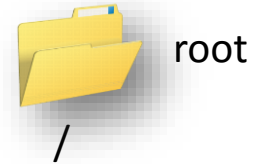
\$ `whoami`

`nelle`

\$ `pwd`

/ `users/nelle`

\$



login: `nelle`

password: `*****`

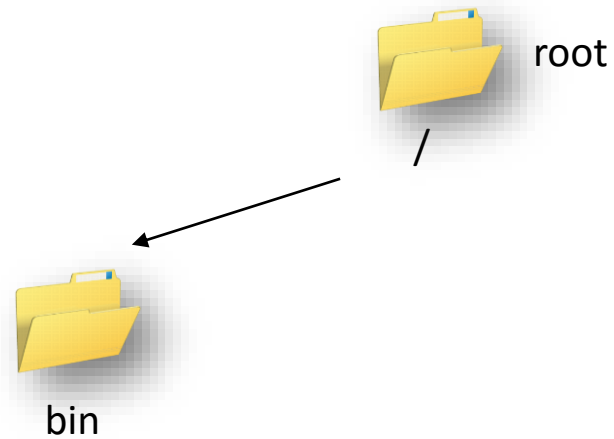
\$ `whoami`

`nelle`

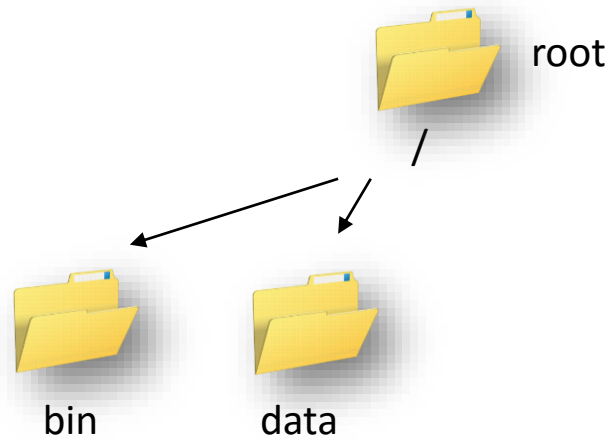
\$ `pwd`

`/users/nelle`

\$



```
login: nelle
password: ****
$ whoami
nelle
$ pwd
/users/nelle
$
```



login: `nelle`

password: `*****`

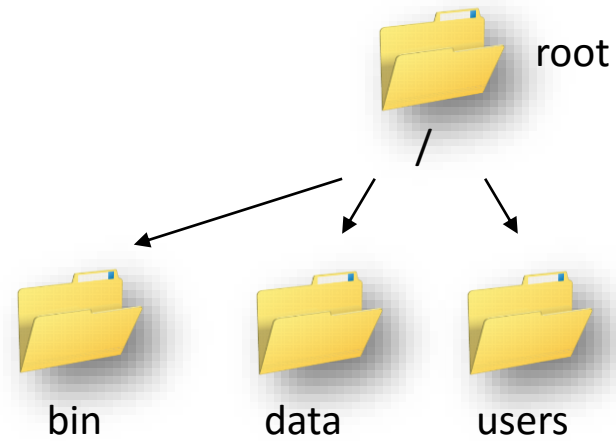
\$ `whoami`

`nelle`

\$ `pwd`

`/users/nelle`

\$



login: `nelle`

password: `*****`

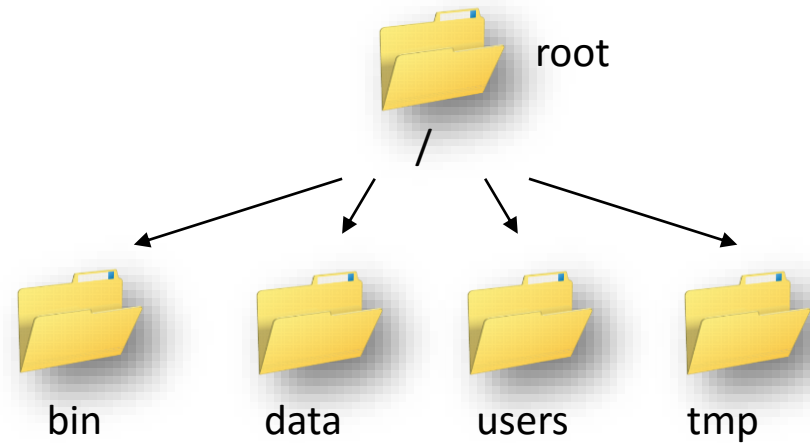
\$ `whoami`

`nelle`

\$ `pwd`

`/users/nelle`

\$



login: *nelle*

password: *******

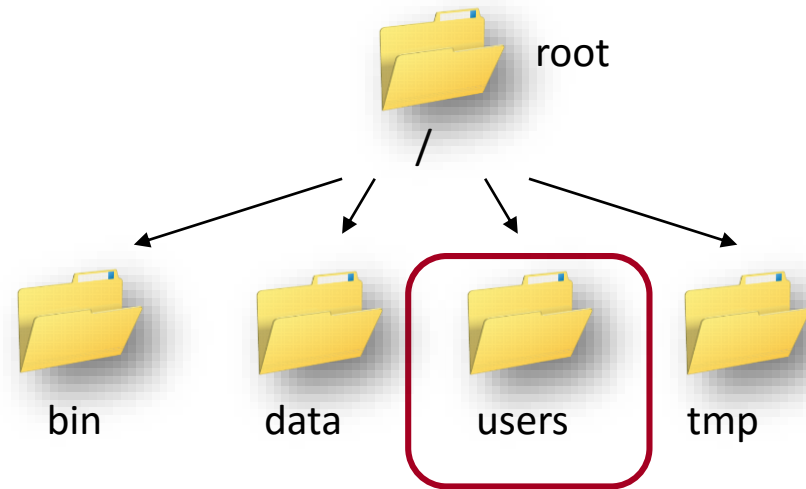
\$ *whoami*

nelle

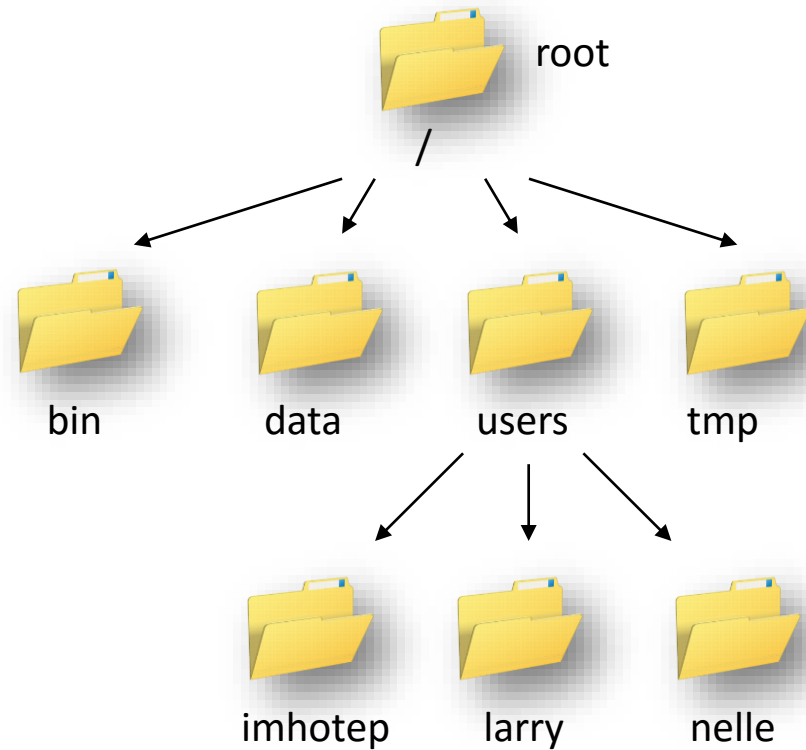
\$ *pwd*

/users/nelle

\$



```
login: nelle
password: ****
$ whoami
nelle
$ pwd
/users/nelle
$
```



login: `nelle`

password: `*****`

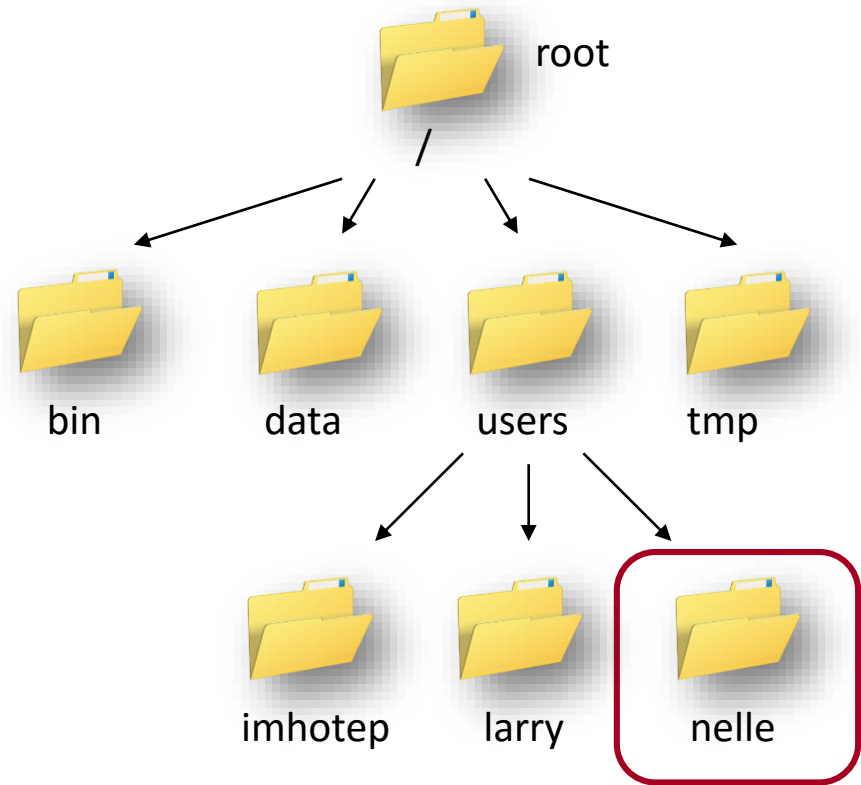
\$ `whoami`

`nelle`

\$ `pwd`

`/users/nelle`

\$



login: *nelle*

password: *******

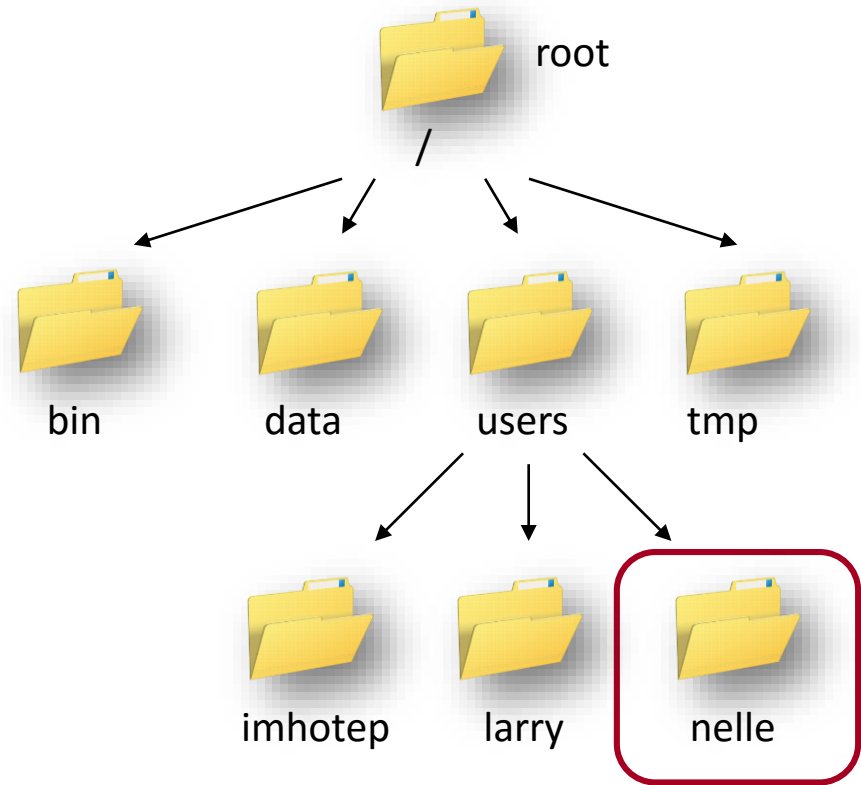
\$ *whoami*

nelle

\$ *pwd*

/users/nelle

\$



login: nelle

password: ****

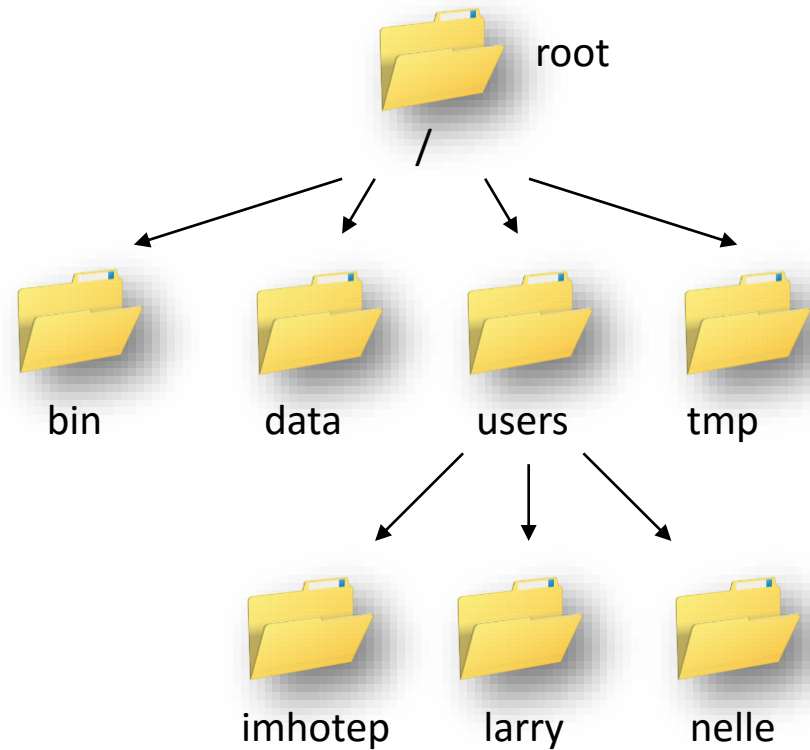
\$ whoami

nelle

\$ pwd

/users/nelle

\$



login: `nelle`

password: `*****`

`$ whoami`

`nelle`

`$ pwd`

`/users/nelle`

`$ ls`  stands for "listing"

login: `nelle`

password: `*****`

\$ `whoami`

`nelle`

\$ `pwd`

`/users/nelle`

\$ `ls` ←

stands for "listing"
sadly more memorable than
most command names

login: `nelle`

password: `*****`

\$ `whoami`

`nelle`

\$ `pwd`

`/users/nelle`

\$ `ls`

<code>bin</code>	<code>data</code>	<code>mail</code>	<code>music</code>
<code>notes.txt</code>	<code>papers</code>	<code>pizza.cfg</code>	<code>solar</code>
<code>solar.pdf</code>	<code>swc</code>		
<code>\$</code>			

```
login: nelle
```

```
password: ****
```

```
$ whoami
```

```
nelle
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls -F
```

*an argument or flag modifying
the command's behavior*

```
bin/
```

```
data/
```

```
mail/
```

```
music/
```

```
notes.txt
```

```
papers/
```

```
pizza.cfg
```

```
solar/
```

```
solar.pdf
```

```
swc/
```

```
$
```

```
login: nelle
```

```
password: ****
```

```
$ whoami
```

```
nelle
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls -F
```

```
bin/
```

```
notes.txt
```

```
solar.pdf
```

```
$
```

```
data/
```

```
papers/
```

```
swc/
```


```
mail/
```

```
pizza.cfg
```

```
music/
```

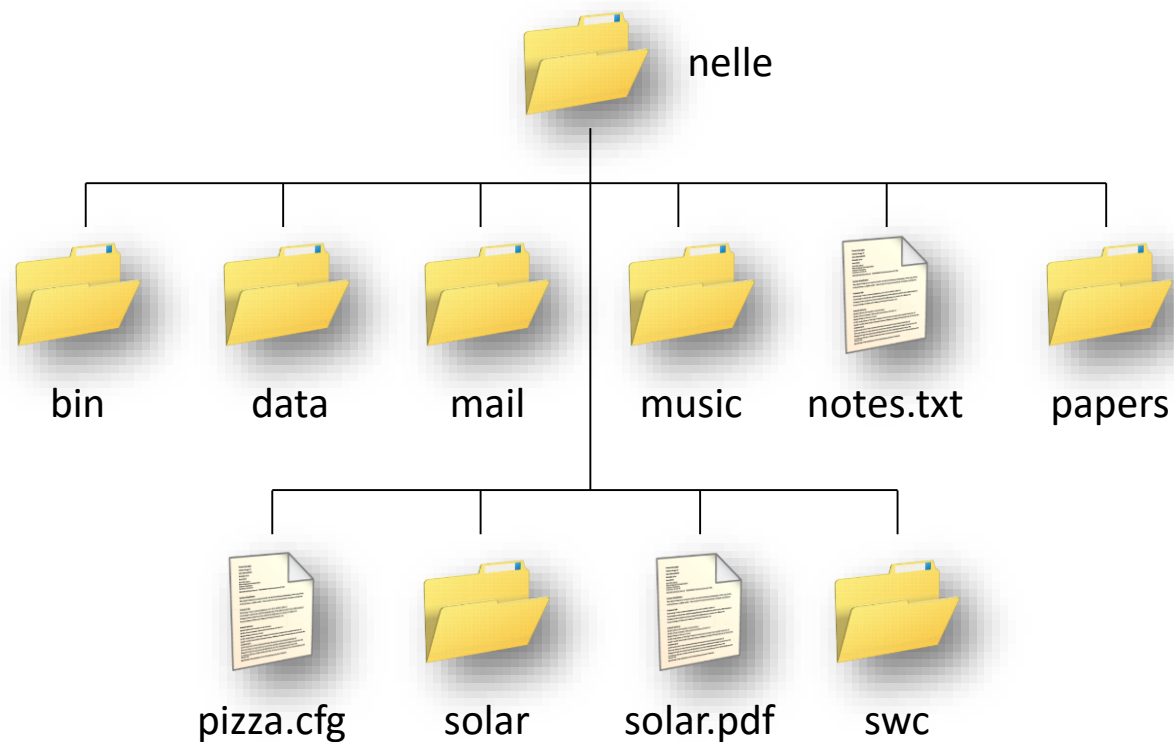
```
solar/
```

adds a trailing '/' to
directory names



```
$ ls -F
```

```
bin/          data/      mail/      music/  
notes.txt    papers/   pizza.cfg  solar/  
solar.pdf    swc/
```



```
$ ls -F
```

```
bin/          data/          mail/          music/  
notes.txt     papers/        pizza.cfg      solar/  
solar.pdf     swc/
```

By convention, use *filename extension* to indicate file type

```
$ ls -F
```

```
bin/          data/          mail/          music/  
notes.txt     papers/        pizza.cfg      solar/  
solar.pdf     swc/
```

By convention, use *filename extension* to indicate file type
.txt for text, .pdf for PDF, .cfg for configuration file, etc.

```
$ ls -F
```

```
bin/          data/          mail/          music/
notes.txt     papers/        pizza.cfg      solar/
solar.pdf     swc/
```

By convention, use *filename extension* to indicate file type
.txt for text, .pdf for PDF, .cfg for configuration file, etc.

But this is only a convention, not a guarantee

```
$ ls -F data
```



```
$ ls -F data
```

```
amino_acids.txt
```

```
elements/
```

```
morse.txt
```

```
pdb/
```

```
planets.txt
```

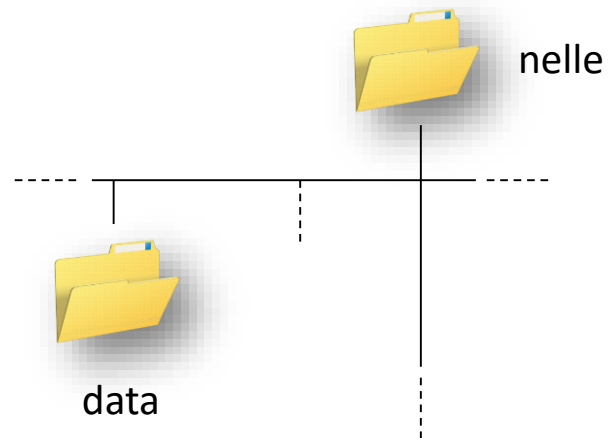
```
sunspot.txt
```

```
$
```

```
$ ls -F data
amino_acids.txt
pdb/
$
```

a relative path

```
elements/
planets.txt
morse.txt
sunspot.txt
```



```
$ ls -F data
```

```
amino_acids.txt
```

```
elements/
```

```
morse.txt
```

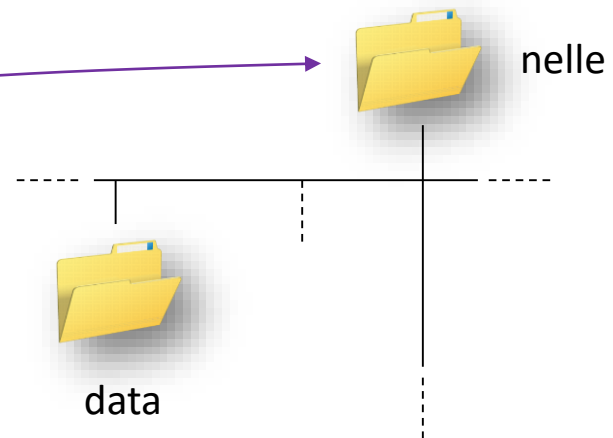
```
pdb/
```

```
planets.txt
```

```
sunspot.txt
```

```
$
```

a relative path
relative to
current working directory



```
$ ls -F /data
```

```
access.log      backup/         hardware.cfg
```

```
network.cfg
```

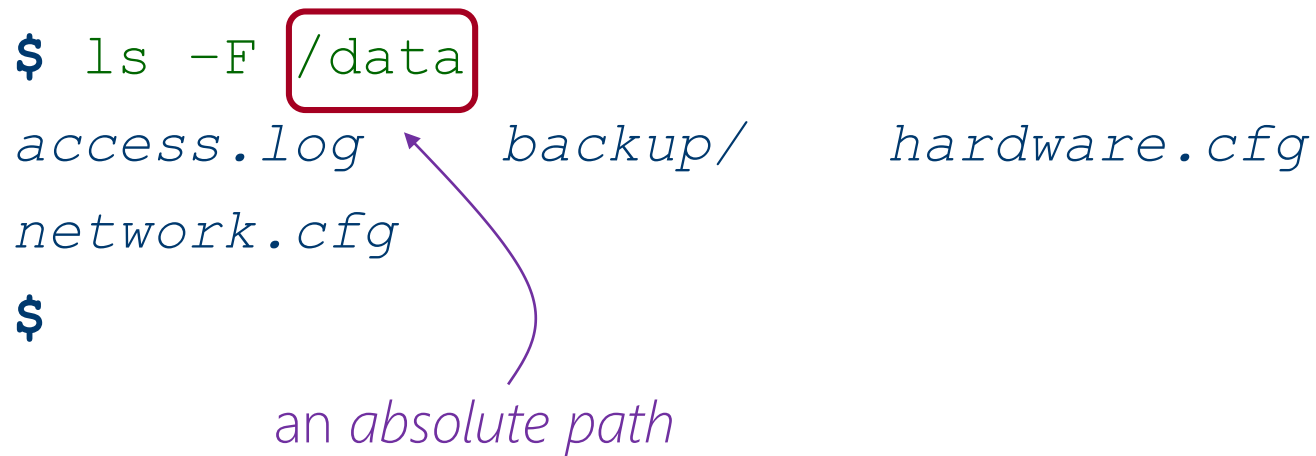
```
$
```

```
$ ls -F /data
```

access.log *backup/* *hardware.cfg*
network.cfg

\$

an absolute path



```
$ ls -F /data  
access.log      backup/         hardware.cfg  
network.cfg  
$
```



an absolute path

leading '/' means "from root"

```
$ ls -F /data
```

```
access.log    backup/    hardware.cfg
```

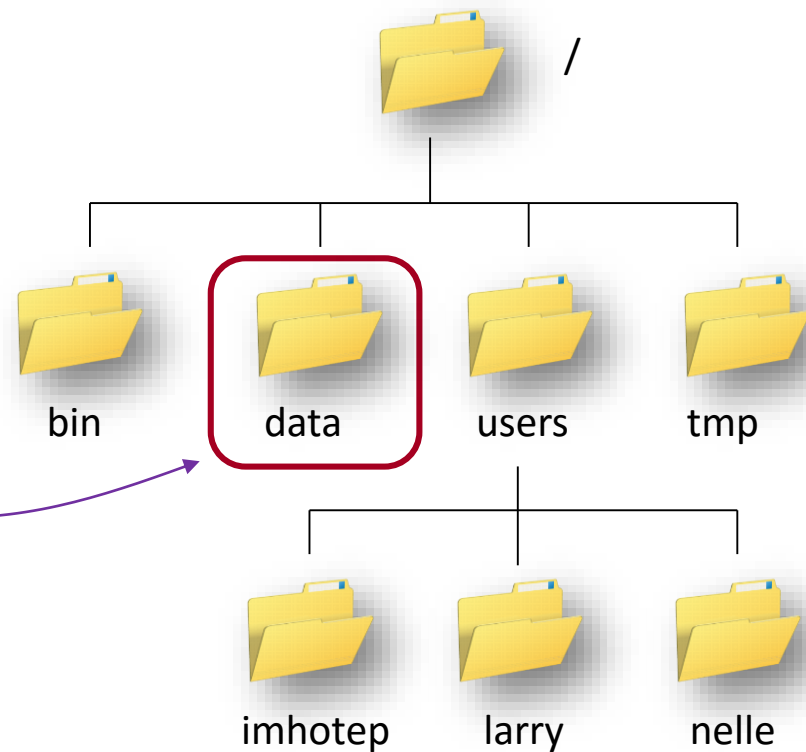
```
network.cfg
```

```
$
```

an absolute path

leading '/' means "from root"

so it always refers to
this directory



```
$ pwd
```

```
/users/nelle
```

```
$
```



```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/          data/          mail/          music/  
notes.txt     papers/        pizza.cfg      solar/  
solar.pdf     swc/
```

```
$
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/          data/          mail/          music/  
notes.txt     papers/        pizza.cfg      solar/  
solar.pdf     swc/
```

```
$ cd data
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/          data/          mail/          music/
```

```
notes.txt     papers/        pizza.cfg      solar/
```

```
solar.pdf     swc/
```

```
$ cd data ← change directory
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/          data/          mail/          music/
```

```
notes.txt     papers/        pizza.cfg      solar/
```

```
solar.pdf     swc/
```

```
$ cd data ← change directory  
           actually doesn't change the directory
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/          data/          mail/          music/  
notes.txt     papers/        pizza.cfg      solar/  
solar.pdf     swc/
```

```
$ cd data ← change directory
```

actually doesn't change the directory
changes the shell's idea of
which directory we are in

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/          data/          mail/          music/  
notes.txt     papers/        pizza.cfg      solar/  
solar.pdf     swc/
```

```
$ cd data
```

```
$ pwd
```

```
/users/nelle/data
```

```
$
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/          data/          mail/          music/  
notes.txt     papers/       pizza.cfg     solar/  
solar.pdf    swc/
```

```
$ cd data
```

```
$ pwd
```

```
/users/nelle/data
```

```
$ ls
```

```
amino_acids.txt  elements/      morse.txt  
pdb/             planets.txt    sunspot.txt
```

```
$
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/          data/          mail/          music/
notes.txt     papers/        pizza.cfg      solar/
solar.pdf     swc/
```

```
$ cd data
```

```
$ pwd
```

```
/users/nelle/data
```

```
$ ls
```

```
amino_acids.txt  elements/      morse.txt
pdb/             planets.txt    sunspot.txt
```

because we're now "in"
this directory




```
$ pwd
```

```
/users/nelle/data
```

```
$
```

```
$ pwd
```

```
/users/nelle/data
```

```
$ cd ..
```

```
$ pwd
```

```
/users/nelle/data
```

```
$ cd .. ←
```

the directory above the current one

```
$ pwd
```

```
/users/nelle/data
```

```
$ cd .. ←
```

the directory above the current one
its parent directory

```
$ pwd
/users/nelle/data
$ cd ..
$ pwd
/users/nelle
$
```

```
$ pwd
```

```
/users/nelle/data
```

```
$ cd ..
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/
```

```
data/
```

```
mail/
```

```
music/
```

```
notes.txt
```

```
papers/
```

```
pizza.cfg
```

```
solar/
```

```
solar.pdf
```

```
swc/
```

```
$
```

```
$ pwd
```

```
/users/nelle/data
```

```
$ cd ..
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

<i>bin/</i>	<i>data/</i>	<i>mail/</i>	<i>music/</i>
<i>notes.txt</i>	<i>papers/</i>	<i>pizza.cfg</i>	<i>solar/</i>
<i>solar.pdf</i>	<i>swc/</i>		

```
$ ls -F -a
```

<i>./</i>	<i>../</i>	<i>bin/</i>	<i>data/</i>
<i>mail/</i>	<i>music/</i>	<i>notes.txt</i>	<i>papers/</i>
<i>pizza.cfg</i>	<i>solar/</i>	<i>solar.pdf</i>	<i>swc/</i>

```
$ pwd
```

```
/users/nelle/data
```

```
$ cd ..
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/          data/          mail/          music/
notes.txt     papers/        pizza.cfg      solar/
solar.pdf     swc/
```

"show all"

```
$ ls -F -a
```

```
./          ../          bin/          data/
mail/       music/       notes.txt     papers/
pizza.cfg   solar/       solar.pdf     swc/
```



```
$ pwd
```

```
/users/nelle/data
```

```
$ cd ..
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/
```

```
data/
```

```
mail/
```

```
music/
```

```
notes.txt
```

```
papers/
```

```
pizza.cfg
```

```
solar/
```

```
solar.pdf
```

```
swc/
```

parent directory

```
$ ls -F -a
```

```
./
```

```
../
```

```
bin/
```

```
data/
```

```
mail/
```

```
music/
```

```
notes.txt
```

```
papers/
```

```
pizza.cfg
```

```
solar/
```

```
solar.pdf
```

```
swc/
```

```
$ pwd
```

```
/users/nelle/data
```

```
$ cd ..
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/
```

```
data/
```

```
mail/
```

```
music/
```

```
notes.txt
```

```
papers/
```

```
pizza.cfg
```

```
solar/
```

```
solar.pdf
```

```
swc/
```

```
$ ls -F -a
```

```
./
```

```
../
```

```
bin/
```

```
data/
```

```
mail/
```

```
music/
```

```
notes.txt
```

```
papers/
```

```
pizza.cfg
```

```
solar/
```

```
solar.pdf
```

```
swc/
```

parent directory

/users

```
$ pwd
```

```
/users/nelle/data
```

```
$ cd ..
```

```
$ pwd
```

```
/users/nelle
```

```
$ ls
```

```
bin/
```

```
data/
```

```
mail/
```

```
music/
```

```
notes.txt
```

```
papers/
```

```
pizza.cfg
```

```
solar/
```

```
solar.pdf
```

```
swc/
```

this directory
itself

```
$ ls -F -a
```

```
./
```

```
../
```

```
bin/
```

```
data/
```

```
mail/
```

```
music/
```

```
notes.txt
```

```
papers/
```

```
pizza.cfg
```

```
solar/
```

```
solar.pdf
```

```
swc/
```

Things are different on Windows

Things are different on Windows

```
C:\Users\nelle
```

Things are different on Windows

`C:\Users\nelle`

Drive letter



Things are different on Windows

`C:\Users\nelle`



Drive letter

Each drive is a separate file system

Things are different on Windows

C:\Users\helle

Backslash \ as separator

A diagram illustrating the use of the backslash character as a directory separator in Windows. It shows the path 'C:\Users\helle' with red boxes around the backslashes. Below the path, the text 'Backslash \ as separator' is written, with two blue arrows pointing from the backslashes in the path to the backslash in the text.

Things are different on Windows

`C:\Users\helle`

Backslash `\` as separator

Unix uses `\` to escape special characters
in names like `my\ files.txt`

Things are different on Windows

```
C:\Users\nelle
```



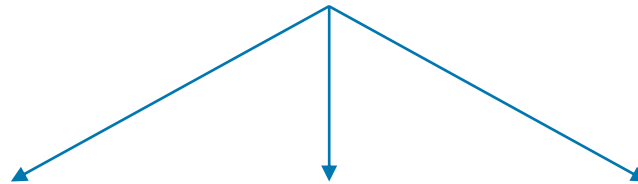
Case insensitive

Things are different on Windows

`C:\Users\nelle`



Case insensitive



`c:\users\nelle`

`C:\USERS\nelle`

`C:\uSeRs\nelle`

Things are different on Windows

```
C:\Users\nelle
```

```
Cygwin: /cygdrive/c/Users/nelle
```



Map drive letters to "directories"



Things are different on Windows

```
C:\Users\nelle
```

```
Cygwin: /cygdrive/c/Users/nelle
```



Map drive letters to "directories"

And use / instead of \

Things are different on Windows

```
C:\Users\nelle
```

```
Cygwin: /cygdrive/c/Users/nelle
```



Map drive letters to "directories"

And use / instead of \

But still case insensitive

Things are different on Windows

```
C:\Users\nelle
```

```
Cygwin: /cygdrive/c/Users/nelle
```



Map drive letters to "directories"

And use / instead of \

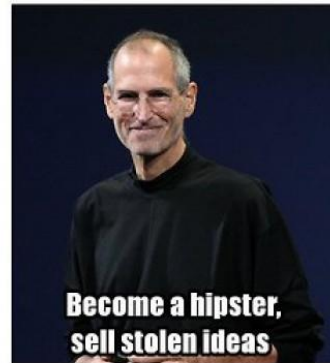
But still case insensitive

Can't put `backup.txt` and `Backup.txt` in a directory

<code>pwd</code>	print working directory
<code>cd</code>	change working directory
<code>ls</code>	listing
<code>.</code>	current directory
<code>..</code>	parent directory

Dennis Ritchie vs. Steve Jobs

Steve Jobs
(b. 24 Feb 1955)



**Become a hipster,
sell stolen ideas**

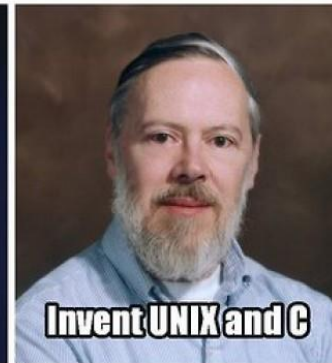
DIED 05 OCT 2011

Without Steve Jobs, we would have:

- No iProducts**
- No over-expensive laptops**

**Praised by media as Jesus
of computing**

Dennis Ritchie
(b. 09 Sep 1941)



Invent UNIX and C

DIED 12 OCT 2011

Without Dennis Ritchie, we would have:

- No Windows**
- No UNIX**
- No C**
- No programs**
- A large setback in computing**
- No generic text languages**
- We would all read in Binary**

Ignored

**They died in the same year, but it seems only few notice
the death of Dennis Ritchie compared to Steve Jobs**

FAQTS – the Game

- Frequently Asked Questions with Tiny Sentences
- Both Q and A with least possible words
- The ideal word count for answers is *two*
- Our second round:

What is  ?

Tutorials

- *Jamil* will help you getting Linux installed on your laptop in case you haven't done that yet

