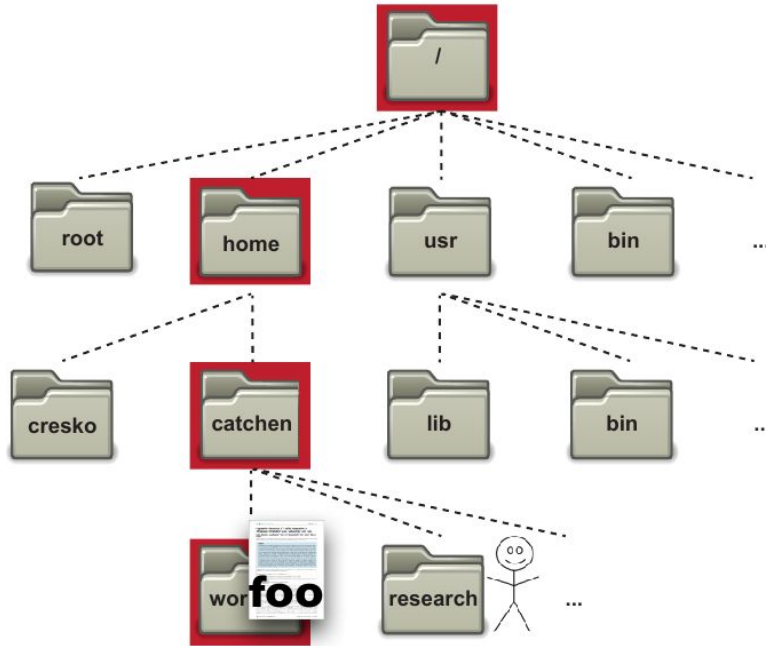


Some Bash course flashback



Absolute Path: `/home/catchen/working/foo`

Relative Path: `../working/foo`

## Relative and absolute paths

A shortcut to your 'home', tilde:

~

Moving through the filesystem:

cd

Knowing where you are:

pwd

% ls ~/

% cd ~/

% cd

% pwd

## Three variants to `ls`

<code>ls -l</code>	<code>ls -la</code>	<code>ls -lh</code>
provides a <i>long</i> listing	includes <i>all</i> files, even hidden files	displays file sizes in <i>human</i> readable numbers

# Important notes before

- Case-Sensitive
  - Tab completion (!?)
  - Single-tab and Double tab
- Spaces “ ” mean a lot!
  - Naming files without spaces!
  - File extensions

# Quoting

```
% echo '$USER'
```

```
% $USER
```

```
% echo "$USER"
```

```
% bagnoud
```

```
% echo $USER
```

```
% bagnoud
```

```
% echo \"
```

```
% echo \>
```

```
% >
```

```
% echo ">"
```

```
% >
```

```
% echo '>'
```

```
% >
```

```
% echo >
```

# Redirection

```
% echo hello > test.txt
```

# Write to file

```
% echo "hello" >> test.txt
```

# Append to file

```
% cat test.txt
```

# Print to screen

# Summary1

- `man` manual for bash commands
- `ls` list directory content
- `ls -l` list directory content with more detail
- `cd DIRNAME` change the directory to DIRNAME



# Four ways to view a text file

less	head	tail	cat
view a text file one screen full at a time	view top 10 lines of a file	view last 10 lines of a file	spit the whole file at once
space-bar: scroll q: quit	-n <number> controls the number of lines	-n <number> controls the number of lines	

# Summary2

- `cat` printing input to output
- `COMMAND > FILENAME` put the output of the COMMAND into a file named FILENAME
- `cp FILENAME1 FILENAME2` makes a copy of FILENAME1 called FILENAME2
- `mv FILENAME1 FILENAME2` renames FILENAME1 to FILENAME2
- `mv FILENAME1 PATH/FILENAME2` moves the file FILENAME1 to the specified PATH and names it to FILENAME2
- `rm FILENAME` removes FILENAME