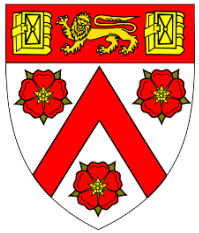
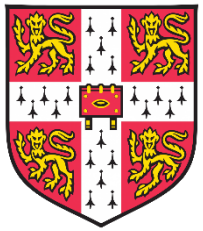




Introduction to Unix shell

Alexey Morgunov
Trinity College & MRC-LMB, Cambridge

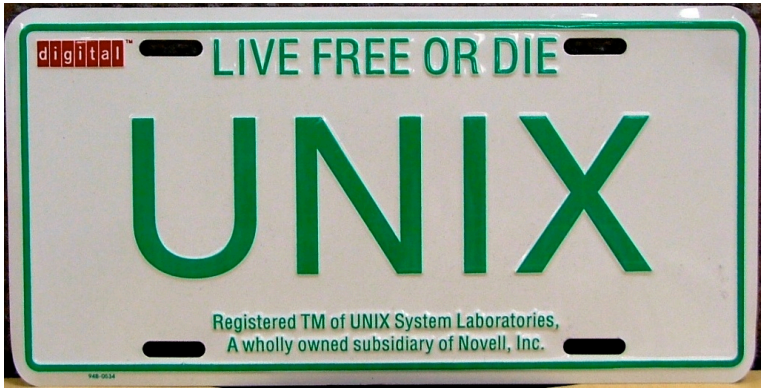
Part III Systems Biology
8 Feb 2017

Start here

github.com/alexeymorgunov/unixshellcourse2017

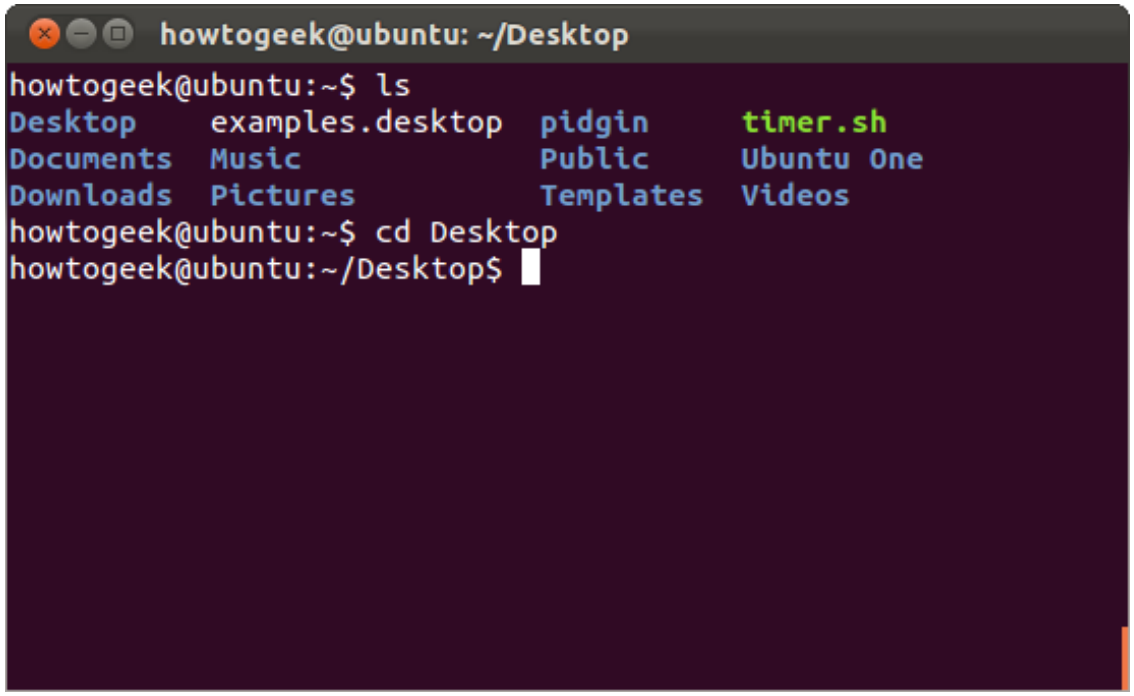
What is Unix?

- Modularly designed operating system
- Simple tools each performing well-defined functions
- Unified file system, means of communication
- Shell scripting and command language to combine simple tools into complex workflows
- Examples: Linux (incl. Android), Mac OS X (incl. iOS)



What is shell?

- Program that accepts commands as text input
- Converts them to operating system functions
- Easy to automate via scripting (c.f. GUI)



Components of coding/scripting skills
(true for any language!)

- Logic
  - Syntax, how commands are structured, how the different components fit together − need to learn!

Components of coding/scripting skills
(true for any language!)

- Logic
  - Syntax, how commands are structured, how the different components fit together − need to learn!
- Awareness
  - Knowing what commands, methods and tricks exist − like having a LEGO brick inventory!

Components of coding/scripting skills
(true for any language!)

- Logic
  - Syntax, how commands are structured, how the different components fit together − need to learn!
- Awareness
  - Knowing what commands, methods and tricks exist − like having a LEGO brick inventory!
- Practice
  - Learning how to combine the bricks to solve increasingly more complex problems

Components of coding/scripting skills
(true for any language!)

- Logic
  - Syntax, how commands are structured, how the different components fit together − need to learn!
- Awareness
  - Knowing what commands, methods and tricks exist − like having a LEGO brick inventory!
- Practice
  - Learning how to combine the bricks to solve increasingly more complex problems
- Google and Stack Overflow
  - Don’t reinvent the wheel − someone somewhere is likely to have encountered the same problem before…

Command syntax

- *doSomething how toFiles*
- *doSomething how sourceFile destinationFile*
- *doSomething how < inputFile > outputFile*
- *doSomething how | doSomething how | doSomething how > outputFile*

```
[command] -[parameters] [file or folder]
```

Start here

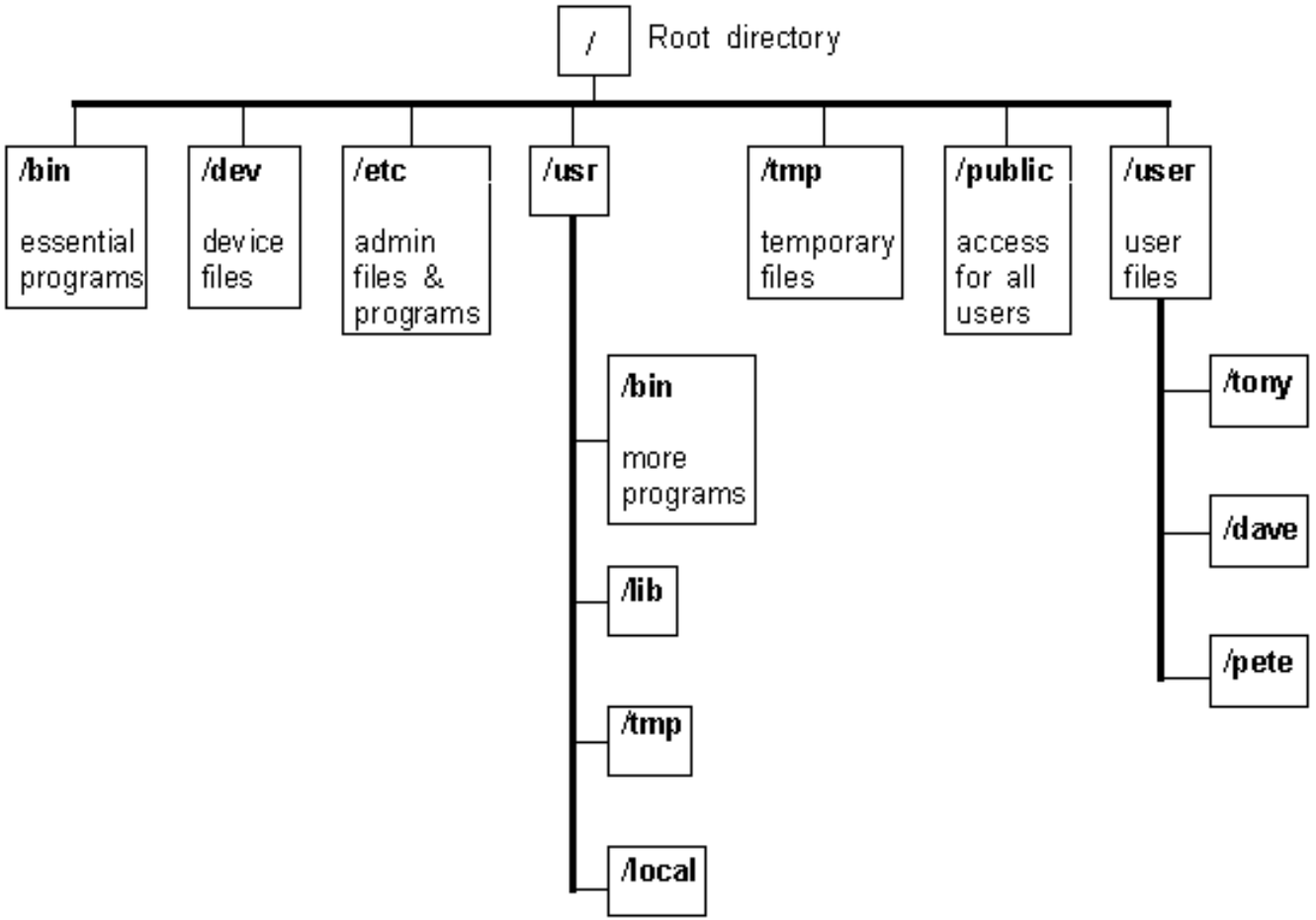
github.com/alexeymorgunov/unixshellcourse2017

Let’s start simple

- `pwd`

Let's start simple

- `pwd`
- File system



Looking and moving around

- `ls`
- `cd`
- `ls -l`
- `man ls`
- `ls -lht`

Create, copy, move, delete…

- `mkdir`
- `echo 'Hello, world!' > hello.txt`
- `cp`
- `mv` -> not just move, used for renaming
- `rm`
- `rmdir`
- `rm -r`
- `find . -name '*.txt' -delete`

Redirection & pipes

- `cat > file.txt   # <CTRL-D>`
- `cat file.txt`
- `cat hello.txt >> file.txt`
- `cat hello.txt file.txt > long.txt`

Redirection & pipes

- `cat > file.txt   # <CTRL-D>`
- `cat file.txt`
- `cat hello.txt >> file.txt`
- `cat hello.txt file.txt > long.txt`
- `sort < long.txt > sorted.txt`
- `cat long.txt | sort > sorted2.txt`

`echo`, variables, `\` and brackets

- `echo ls`
- `echo “ls”`
- `echo ` `ls` `

`echo`, variables, `\` and brackets

- `echo ls`
- `echo “ls”`
- `echo ` `ls` `
- `echo $HOME`
- `echo ${HOME}`
- `echo \$HOME`
- `echo \\$HOME`
- `echo \\\$HOME`

`echo`, variables, `\` and brackets

- `echo ls`
- `echo "ls"`
- `echo \`ls\``
- `echo $HOME`
- `echo ${HOME}`
- `echo \$HOME`
- `echo \\$HOME`
- `echo \\\$HOME`
- `echo '$HOME | ls'`
- `echo \`$HOME | ls\``
- `echo "$HOME | ls"`
- `echo f{oo,ee,e}d`
- `echo $((42+42))`

Looking at text files

- `cat long.txt`
- `less long.txt`
- `gedit long.txt`
- `head -3 long.txt`
- `tail -3 long.txt`
- `tail -n +3 long.txt`
- `wc long.txt`

what options does `wc` have?

Sorting

- `cat > animals.txt` `# <CTRL-D>`
- `cat > numbers.txt` `# <CTRL-D>`
- `sort animals.txt`
- `sort numbers.txt`
- `sort -ur animals.txt`
- `sort -n numbers.txt`
- `uniq -c`

what other options does `sort` have?

`grep` and `sed`

- `grep "o" animals.txt`
- `grep -v "o" animals.txt`
- `sed "s/o/O/g" animals.txt`
- `sed -f script.sed animals.txt`

Browse the links in the Notes about `sed` to see many-many more uses of this function!

Operations on strings

- `rev`
- `cut`
- `join`
- `paste`
- `tr`

Investigate what the above functions can do and then attempt the exercises!

If you have time left…

Check out shell scripting in the “more advanced” section of the notes on GitHub!

github.com/alexeymorgunov/unixshellcourse2017

Thank you!

Have any questions, comments?
Email me: asm63@cam.ac.uk