Introduction to the command line with Bash: Learn useful fundamentals and make your own scripts!

interactive intro workshop

2024-11-06

Jakob Berg Jespersen, Data Science Platform at DTU Biosustain
datascience@biosustain.dtu.dk

agenda

- motivation to learn how to use command line tools
- terminal & shell
- text editor
- commandline tools
- file permissions
- transfer of files
- installing software
- customizing work environment

Motivation & Justification

- understand how the computer "thinks"
- decide the tradeoff, when do you do something manual, and when do you automate it?
- what you do in the terminal could be made into a script that can be saved for reproducibility.
- there is a plethora of open source software developed by other scientists.

show of hands ?

- Bioengineering/Biosustain/anything else?
- windows/mac/anything else
- windows users with WSL?
- who has US english or EU english keyboard layout?

How to get started with BASH if you are not on a linux machine

THE BOURNE-AGAIN SHELL

git bash / WSL2 + Ubuntu

24.04 LTS

Windows Terminal

Multi platform

MacPorts

port / brew

Mac built in Terminal / iTerm2

Inside the terminal

```
bash-3.2$ bash --help
bash-3.2$ bash --version
bash-3.2$ man bash
```

command line argument:

Unless there is a particular reason, most software is designed so that -- prefixes a word and - prefixes a single character.

i.e.: `ls -ltrh  == ls -l -t -r -h`

Inside the terminal

```
bash-3.2$ which man

bash-3.2$ man less

bash-3.2$ less --help | less
```

Anatomy of a man page

NAME

The name of the command or function, followed by a one-line description of what it does.

SYNOPSIS

In the case of a command, a formal description of how to run it and what command line options it takes. For program functions, a list of the parameters the function takes and which header file contains its declaration.

DESCRIPTION

A textual description of the functioning of the command or function.

EXAMPLES

Some examples of common usage.

SEE ALSO

A list of related commands or functions.

characters, what are their names?

```
`        ( )        !
~        [ ]        @
"        { }        #
'         -         $
/         *         %
\         .         ^
|         ,         &
:         ;         +
<         >         =
```

characters

Backtick	`	Round Brackets	( )	Exclamation	!
Tilde	~	Square Brackets	[ ]	At	@
Double quote	"	Curly Brackets	{ }	Pound/Hash	#
Single quote	'	Dash/Minus	-	Dollar	$
Slash	/	Asterisk	*	Percentage	%
Backslash	\	Dot	.	Hat/Circumflex	^
Pipe	|	Comma	,	Ampersand	&
Colon	:	Semicolon	;	Plus	+
Less than	<	Greater than	>	Equal	=

whitespace characters and line breaks

```
Space
tab
newline
carriage return
```

lets make a folder and enter it

pwd - print working directory

ls – list files in directory

cd - change directory (you should now be in ~ your home directory

mkdir bashworkshop – make a directory named bashworkshop

cd bashworkshop – enter the directory (try hitting tab during typing for autocomplete)

lets say hello to the person next to us

- echo 'hej'
- name=Albert
  - echo hej $name
  - echo "hej $name"
  - echo 'hej $name'
- What happens if we had used backticks?

lets make a file

- echo "hej $name" > greeting.txt
- cat greeting.txt
- echo "hej $name" >> greeting.txt
- What is the difference
- touch emptyfile.txt
- cat, | , > grep, sort, …
- ls

lets make a file

- Try to redefine the $name variable and append a new line to the greeting.txt file
  - i.e. echo "hej Jakob" >> greeting.txt

you should have different lines in the file

```
cat greeting.txt

cat greeting.txt | sort

cat greeting.txt | grep hej

cat greeting.txt | grep Jakob

cat greeting.txt | grep fejwlkjbfg

cat greeting.txt | grep Jakob | wc -l
```

lets rename the file

- mv greeting.txt greetings.txt
  (moving the content from one file to another)
- ls -l
  what do we see in the ls –l output?

lets copy the file

- cp greetings.txt emptyfile.txt
  (copying the content from one file to another)
- ls -l

what exactly did we do?

- history

text editors (.txt)

- on most systems you will encounter bash as the shell, some text editors are almost always installed:
  nano
  vi
- vi can be customized. Some people prefer to install emacs or neovim
- outside the terminal you have many more choices, i.e. vscode
- if your put your text files in an office program, you risk your - will be converted to – as well as text getting Auto Capitalized etc.

lets make our first script in nano or vim

- make a file called hello.sh
- it should say hello to someone at your table when you run it.

- bash ./hello.sh
- ./hello.sh
  (we need to add a line pointing to the program needed=bash)
  hashbang/shebang: #!/bin/bash
  which bash

nano – simple text editor, you can just type

- hotkeys listed in bottom, ^ = Ctrl

General Terminal hotkeys:

- copy and paste in terminal is different from other programs
- Copy = Shift + Ctrl + C / Cmd + C / selecting text with cursor
- Paste = Shift + Ctrl + V / Cmd + V / middle click with cursor

What does Ctrl + C do ?????

How about Ctrl + Z ???

vim – powerful text editor

- omnipresent, it is almost always on the server you visit.
- old, origins traces back to before arrow keys
- notoriously steep learning curve

file owner and permissions

- chmod
- sudo

Downloading

- scp / sftp (command line vs interactive) i.e. filezilla
- most systems have curl or wget (download files)
- some have git (download software projects/repositories)

Software:

- mac: port / brew
- ubuntu/debian: apt / snap
- fedora: dnf / flatpak
- DTU HPC: dont install software, do ` module load` instead

some websites will ask you to install software

- curl URL | bash
- Please download first to look at the code (keep the code in case you need to know what you did)
- curl –O URL
- less file
- bash file

some websites will ask you to install software

- curl | bash
  please be very careful if you ever have to use sudo for these types of setups
- icanhazip.com

https://major.io/p/a-new-future-for-icanhazip/

Downloading with curl or wget

- curl and wget are both useful for downloading files.
  will insert example download here

  go to github.com/biosustain/dsp_workshop_bash
  copy the URL:
  https://raw.githubusercontent.com/biosustain/dsp_workshop_bash/refs/heads/main/Readme.md

- curl

Downloading a repository with git

- git clone https://github.com/biosustain/dsp_workshop_bash

get / make a script

- #!/usr/bin/env bash
- #!/bin/bash

- # comment in bash

- echo 'code in bash'

Example: `script.sh`

```
#!/bin/bash
# Albert 2024-09-25
# Script to add sample names to the gene calls

# Loop through all .fna files in the current directory
for file in *.fna; do
    echo "Working on $file"

    # Extract filename without extension
    filename=$(basename "$file" .fna)

# Use sed to insert the filename at the start of each line beginning with '>'
    sed -i '' "s/^>/>${filename}_/" "$file"
done

echo "Sample name inserted into lines starting with '>' in all .fna files."
```

Example: `script.sh`

```
#!/bin/bash
# Albert 2024-09-25
# Script to add sample names to the gene calls

# Loop through all .fna files in the current directory
for file in *.fna; do
    echo "Working on $file"

    # Extract filename without extension
    filename=$(basename "$file" .fna)

# Use sed to insert the filename at the start of each line beginning with '>'
    sed -i '' "s/^>/>${filename}_/" "$file"
done

echo "Sample name inserted into lines starting with '>' in all .fna files."
```

ssh

- get on **DTUsecure wifi** and connect to login1.hpc.dtu.dk with ssh
- try to run the command **whoami** in your terminal
- ssh DTUUSERNAME@ login1.hpc.dtu.dk

https://www.hpc.dtu.dk/?page_id=4377

scp

Upload:

- scp file.txt DTUUSERNAME@transfer.gbar.dtu.dk:~/

Download:

- scp DTUUSERNAME@transfer.gbar.dtu.dk:~/file.txt ~/

You can also access the directory with an sftp browser like filezilla

https://www.hpc.dtu.dk/?page_id=4377

Takehome messages

- The best terminal is the one you already have
- There are certain unix tools you will find on almost all servers
  - bash, nano, vim, grep, sed, sort, uniq, head, tail, cat, less …
- dont just paste/pipe things you find on the web
- use a proper text editor, not Office programs
- give your files/folders the permissions they need
- learn about git in our next workshop

Upcoming events

- November 13th 2024 - Git and Github
- November 20th 2024 - Data Visualization
- November 27th 2024 - Nextflow

`datascience@biosustain.dtu.dk`