UNIVERSITY OF BIRMINGHAM

Introduction to Linux

March 2021

http://intranet.birmingham.ac.uk/bear

Housekeeping

- Today's workshop – a mixture of us talking and hands-on activities
- Ask questions as we go along via the Chat box or raise your hand
- Refreshment/comfort break during workshop 2
- Feedback – Zoom poll

Overview

- Section 1
  - Using Linux with BlueBEAR
  - Understanding the BlueBEAR workflow
  - Accessing BlueBEAR
- Section 2
  - Creating files
- Section 3
  - Basic Linux commands
  - File management

Workshop 1
Logging into BlueBEAR
(10 mins)

Workshop 2
Directories and files
(20 mins)

Break – 10 mins ~10:50 am

- Section 4
  - File permissions
  - Write and run a simple program
- Section 5
  - Next steps and other BEAR services

Workshop 3
Write & run a program
(20 mins)

Courses - Modular

First steps

- Intro to Linux (Canvas)
- Intro to Linux (f2f)
- LinkedIn courses

Intermediate

- Introduction to BlueBEAR
- Python
- R
- Git
- NVIDIA
- MATLAB

Further steps

- C ++

UNIVERSITY OF BIRMINGHAM

Introduction to BlueBEAR

- Section 1

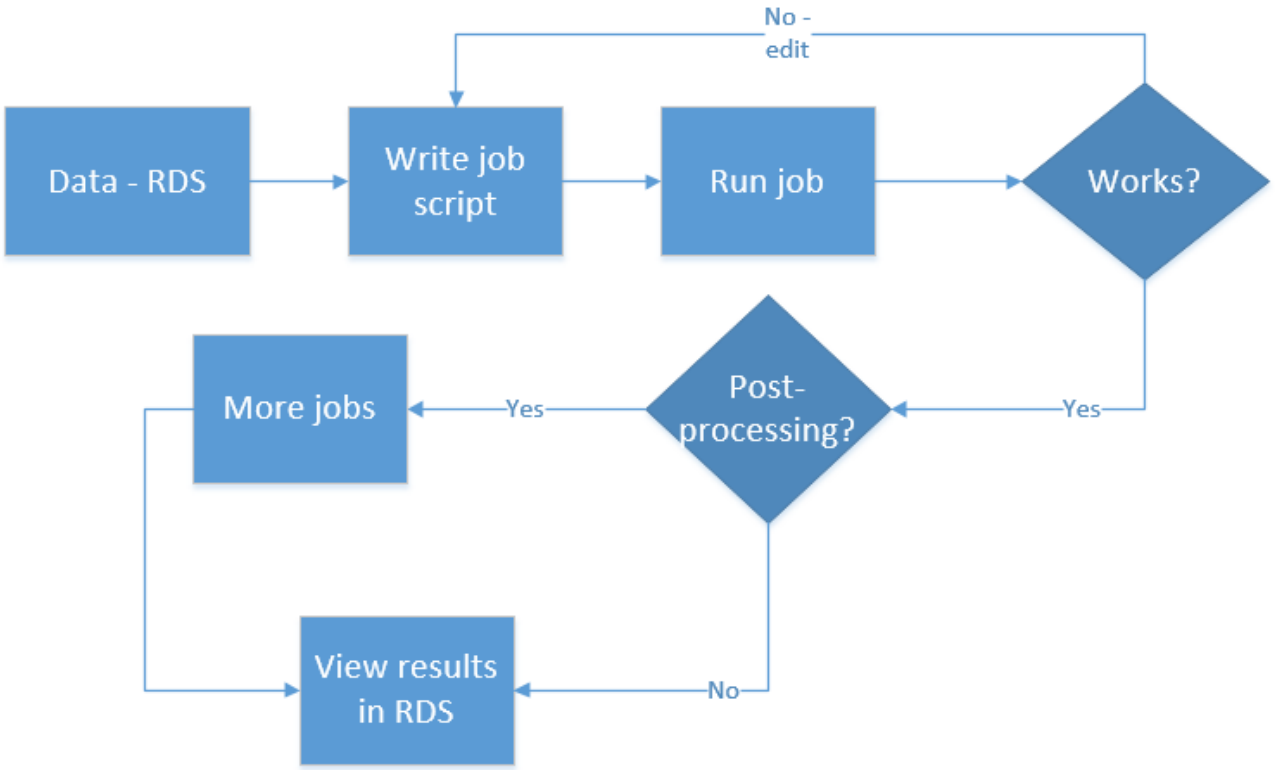
What is BEAR?

- http://intranet.birmingham.ac.uk/bear
- Birmingham Environment for Academic Research
- BEAR is a collection of services: HPC, storage, fast networking, ...
- BlueBEAR refers to the Linux High Performance Computing (HPC) environment
- BEAR services are FREE at the point of use

BlueBEAR

- Users need to register to use the service
- Users are attached to (multiple) projects
- Projects are created by staff and are for 5 years
- Projects are used to account for time on the cluster
- Registrations are via: https://intranet.birmingham.ac.uk/it/teams/infrastructure/research/bear/bluebear/bluebear-registration.aspx

BlueBEAR Workflow



UNIVERSITY OF BIRMINGHAM

Accessing and using BlueBEAR

About Linux

- The interface used for BlueBEAR – how we connect to it and tell it what to do
- It's a huge subject; we are only covering the basics
- There are often multiple ways of doing things

Accessing BlueBEAR

- You must register for access to BlueBEAR
- You will need an SSH (Secure Shell) client (e.g. PuTTY); Mac users - Terminal
- You can connect to the cluster from the University network or via Remote Access Service (apply via ServiceDesk)
- Use your normal University (ADF) username and password
- Interface is command line
- X service for graphical applications, e.g. Exceed, XQuartz (Mac)

Logging in from a Mac

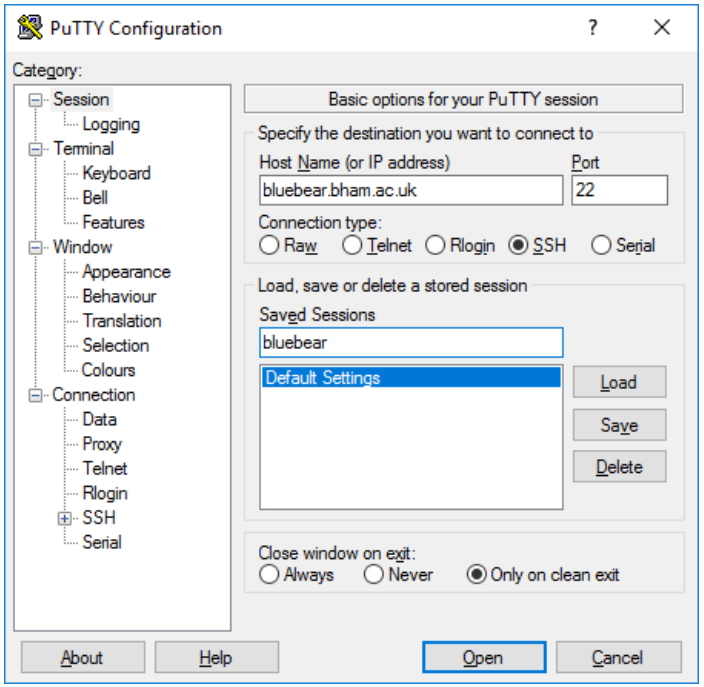
- Open iTerm or Terminal
  - Finder window – Applications>Utilities
- To connect to the training VM type:

```
ssh <username>@172.31.11.62
```

- For connecting to BlueBEAR after the course type:

```
ssh <username>@bluebear.bham.ac.uk
```

Setting up PuTTY



- **Host name**: 172.31.11.62 (training server)
- **Saved Sessions**: bluebear training
- Instructions here: https://intranet.birmingham.ac.uk/it/teams/infrastructure/research/bear/bluebear/accessing-bluebear-using-putty-and-exceed.aspx

Workshop 1 – setting up and logging in

Time: 10 minutes

- Log in to the Remote Access Service
- Open and set up PuTTY
- Log in to BlueBEAR training - 172.31.11.62

Section 2 -

Creating files

The Command Line

- Type in commands
- Commands, Files and programs are CaSe-SeNsItIvE
- Spaces are key – get them in the right place
- You type commands into a program called the “shell” – `bash` is the default
- Text can be copied from other sources, e.g. Notepad

Directories and files

- Unlike Windows there are no “drives” (i.e. no ‘C:\’, ‘D:\’, ‘U:\’ etc.)
- Everything exists under a directory, ‘/’, called the root directory.
- Useful directories to know:
  - ‘`/rds/homes/u/username`’ – each user usually has a directory here with their own files in it, known as their ‘home directory’. 20 GB. For settings, user environment files (files that start with a dot).
  - RDS (`/rds/projects/p/project_title`)
    - Should be used for all data, job scripts, output etc.

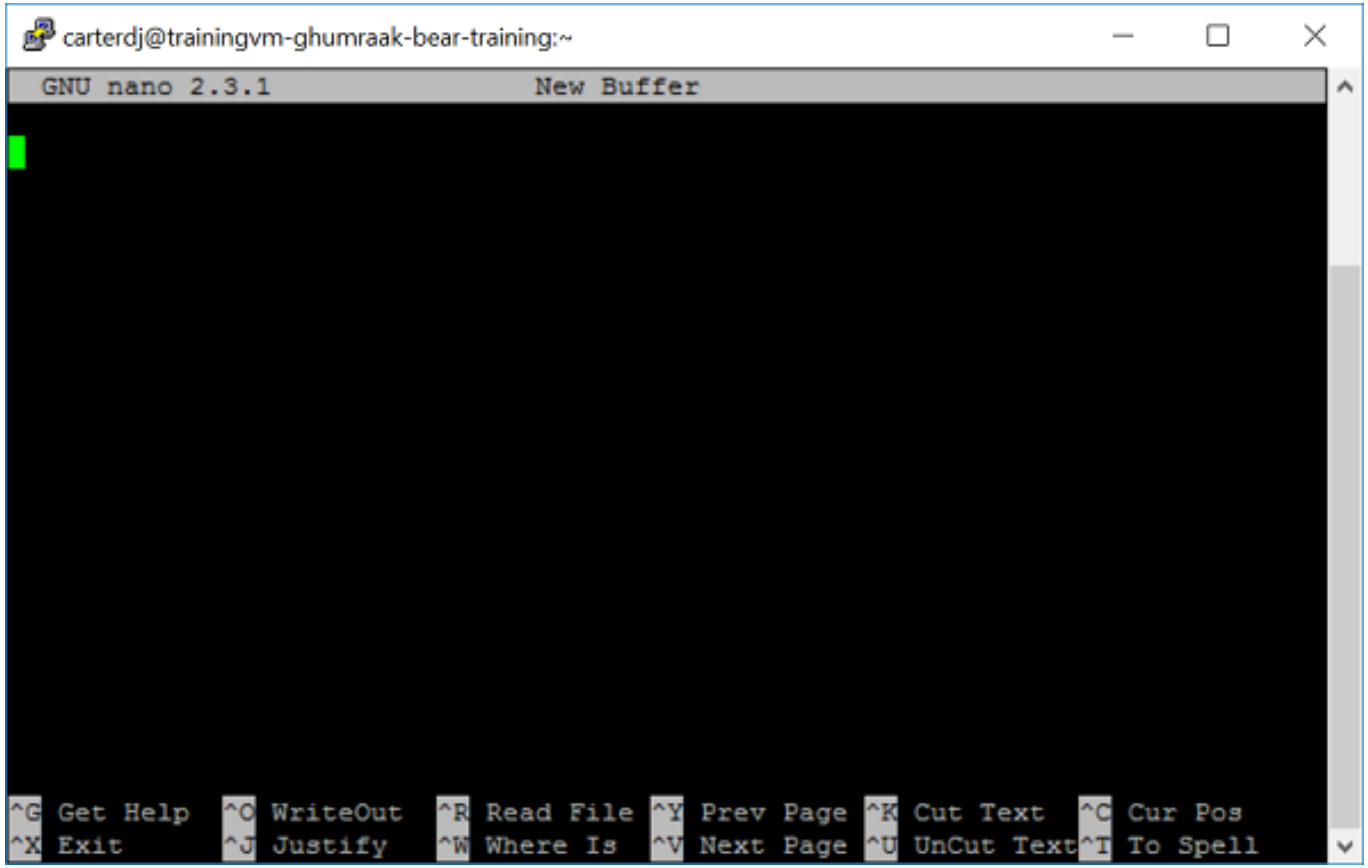
Finding Applications

- A path is where the system looks to find programs (Not where you walk!)
  `echo $PATH`
- These are already set up on BlueBEAR
- `Command not found` means the program isn't in your path

Creating a file - nano

- From the command prompt type `nano`
- Runs in the terminal window
- Enter text
- Commands are at the bottom of the screen – `ctrl+letter`, e.g. `ctrl+w` to find ("Where Is"), `ctrl+x` to exit
- `ctrl+o` to save ("WriteOut"), enter a file name, press enter
- Alphanumeric filenames
- See Canvas course for graphical interface - Exceed

nano


carterdj@trainingvm-ghumraak-bear-training:~
GNU nano 2.3.1 New Buffer
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

Section 3 –

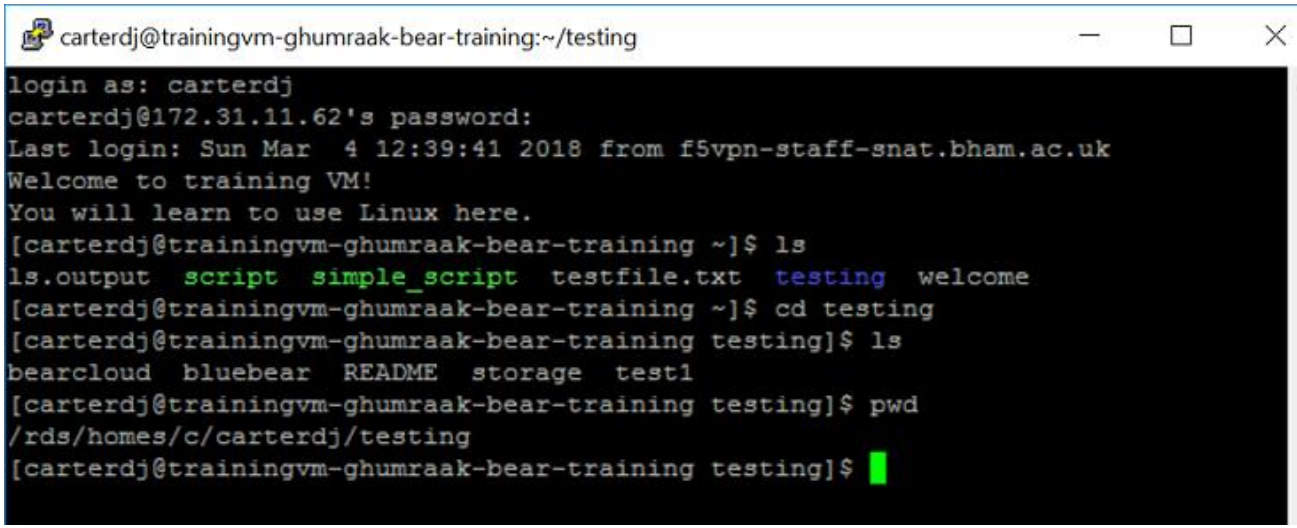
Commands

Basic file commands:

- `ls`

  list files

(don't delete ones starting with . !)



- `ls -l`

  list files with detail

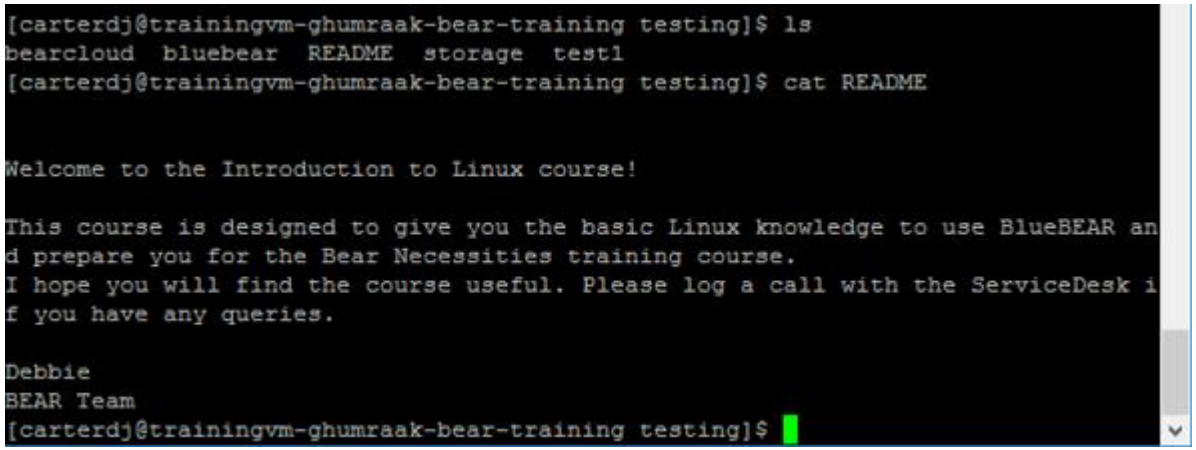
- `cd`

  change directory

- `pwd`

  print current (working) directory

Basic file commands:



- Basic file commands:

  `cat filename`

  view the contents of a file, better for smaller files

  `head` *`filename`*

  watch the start of file as it grows

  `tail -f` *`filename`*

  watch the end of file as it grows

Command syntax

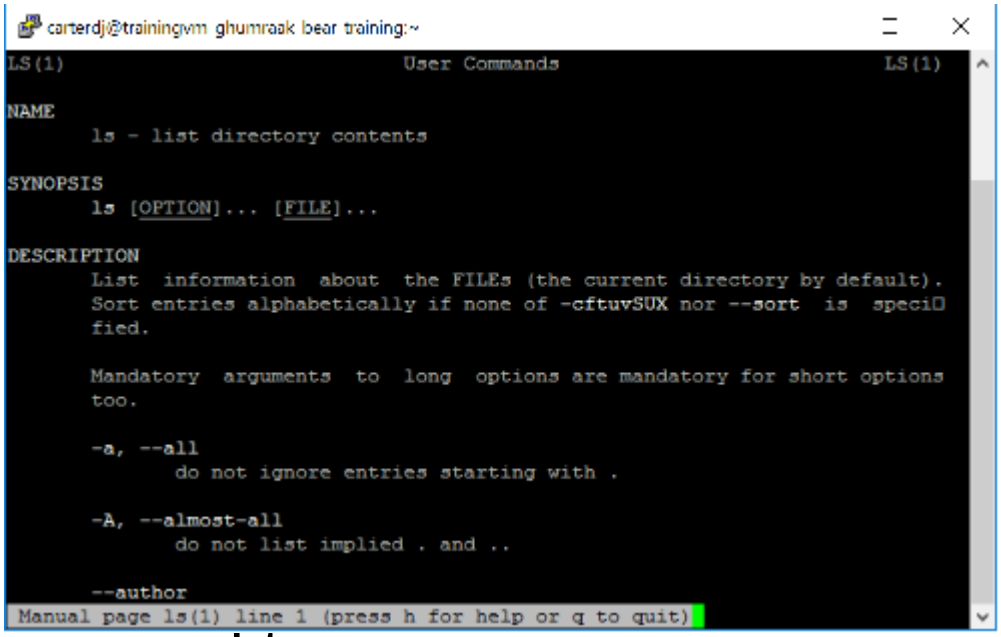
- Commands are used to tell the computer what you want it to do, e.g.

  ls [option(s)] [file(s)]

- Use the manual (man) or Google to see options

Command	Options	Arguments
What you want to do	Information that alters the behaviour of the command	File name or other data that is needed by the command
ls	-l	filename

Help? How do I use this command?



- Each command *should* have a manual page for it.
  - To view it type `man` *`command`*, e.g. to view the manual for the man command type:
    `man man`
- Press `q` to quit, use the arrow keys to scroll

Tips and tricks on the command line

- Press the **up arrow** and you will see your previous commands, so can use them again
- Type `history` at the command prompt and you will see the history of what you have typed and re-run commands
- The '**tab**' key is magic, pressing it:
  - once: complete the current command or filename if there is one match
  - twice: list all the matching options if there is more than one match
- `CTRL+c` or `q` will return you to the command ($) prompt
- `cd ~` returns to home directory
- `clear` clears screen and returns to $ prompt
- to **paste text into PuTTY**, copy it from the source location, then right click in PuTTY and the text will be pasted

File management:

- Basic directory management commands:

  `mkdir` *`newdirectory`*

  create directory
- `rmdir` *`directory`*

  remove directory (only works if directory is empty). Note there is no warning/confirmation message!

```
[carterdj@trainingvm-ghumraak-bear-training ~]$ ls
ls.output  script  simple_script  testfile.txt  testing  welcome
[carterdj@trainingvm-ghumraak-bear-training ~]$ mkdir matlab
[carterdj@trainingvm-ghumraak-bear-training ~]$
```

Organising files:

- Basic file management commands:
  - `touch` *`file`*

    create an empty file
  - `rm` *`file`*

    remove a file (there is no recycle bin!)
  - `cp` *`file newfile`*

    copy a file (creates a duplicate)
  - `mv` *`file newfile`*

    move a file (renames the file)

Editing files

- If a file has been created in Windows, you can use dos2unix to make it Linux-friendly
  - Windows files use different line endings

    `dos2unix ~/`*`filename`*
- Lots of editors under Linux
  - nano is a basic, easy to use one
  - Others – joe, vim, emacs

Editing files

- View and edit a file:
  - Check the contents of the file:
    `cat filename`
  - Edit the file in nano:
    try `CTRL+k, CTRL+u, CTRL+w` to see what they do

A useful tool

- **grep string filename –**

to search for alphanumeric or numeric characters in a specific file

  - Look for number of occurrences in the file e.g. bear

```
grep bear README
```

  - What does

```
grep -ic bear README
```

do?

A cautionary tale….

How Toy Story 2 almost got deleted!

Workshop 2 – directories and files

Time: 30 minutes (including break)

- Copy some existing directories/files using this:

```
cp -r /rds/projects/2017/ghumraak-bear-training/testing/ .
```

- View the example file **bearcloud** (in `testing` directory)
- Create a directory called **username_test**
- Create a file in nano or your preferred editor, add some content, save and close
- Edit the file
- List your files
- View the contents of the file(s) you created
- Advanced exercises – in Canvas

Section 4 –

File permissions

File permissions:

- Files (and directories) can have different permission sets for groups and users

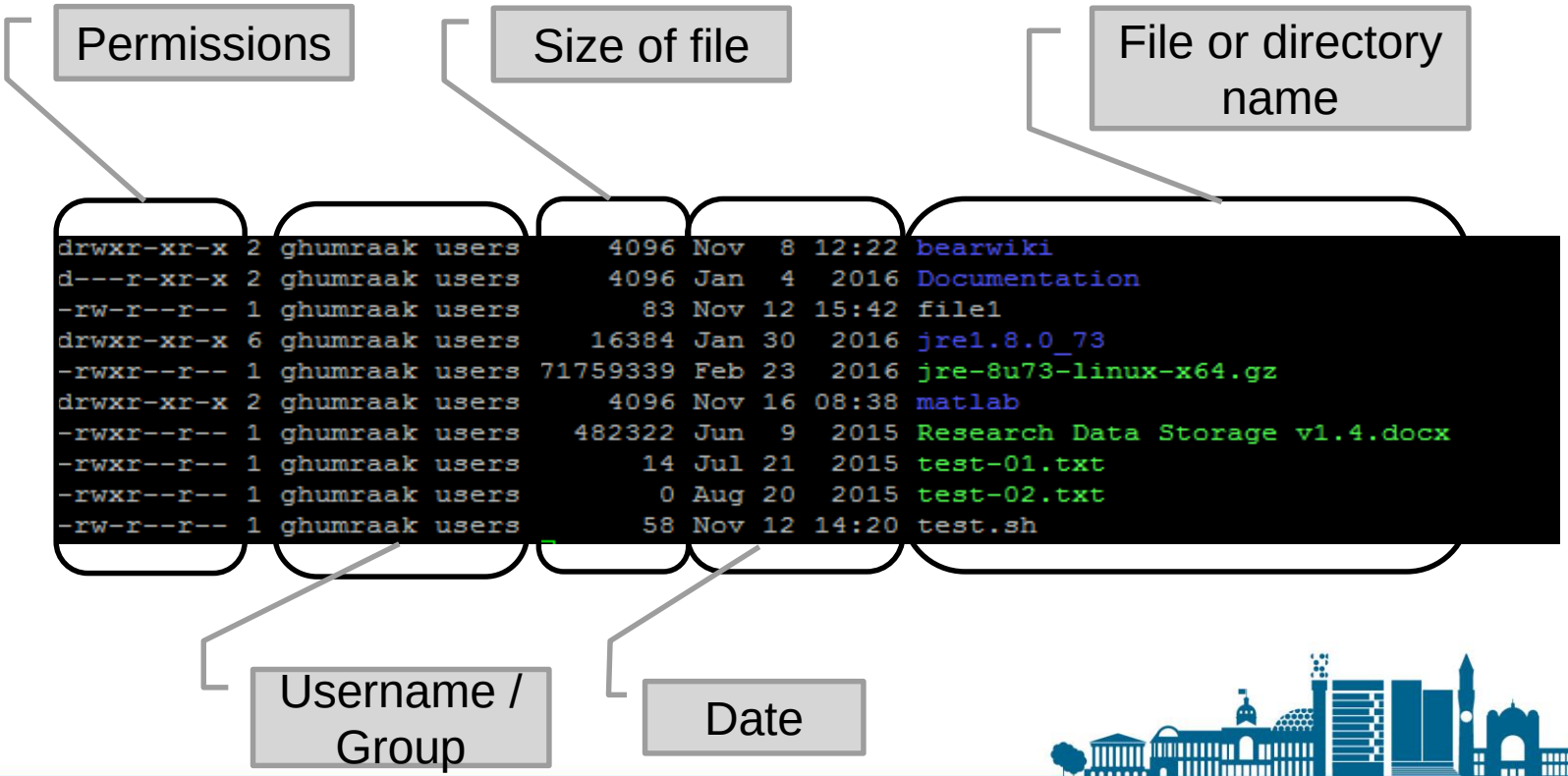
Files	Directory
r – read the file	r – list contents of directory
w – write to the file	w – create new files/folders
x – execute (run)	x – traverse (e.g. cd to directory)

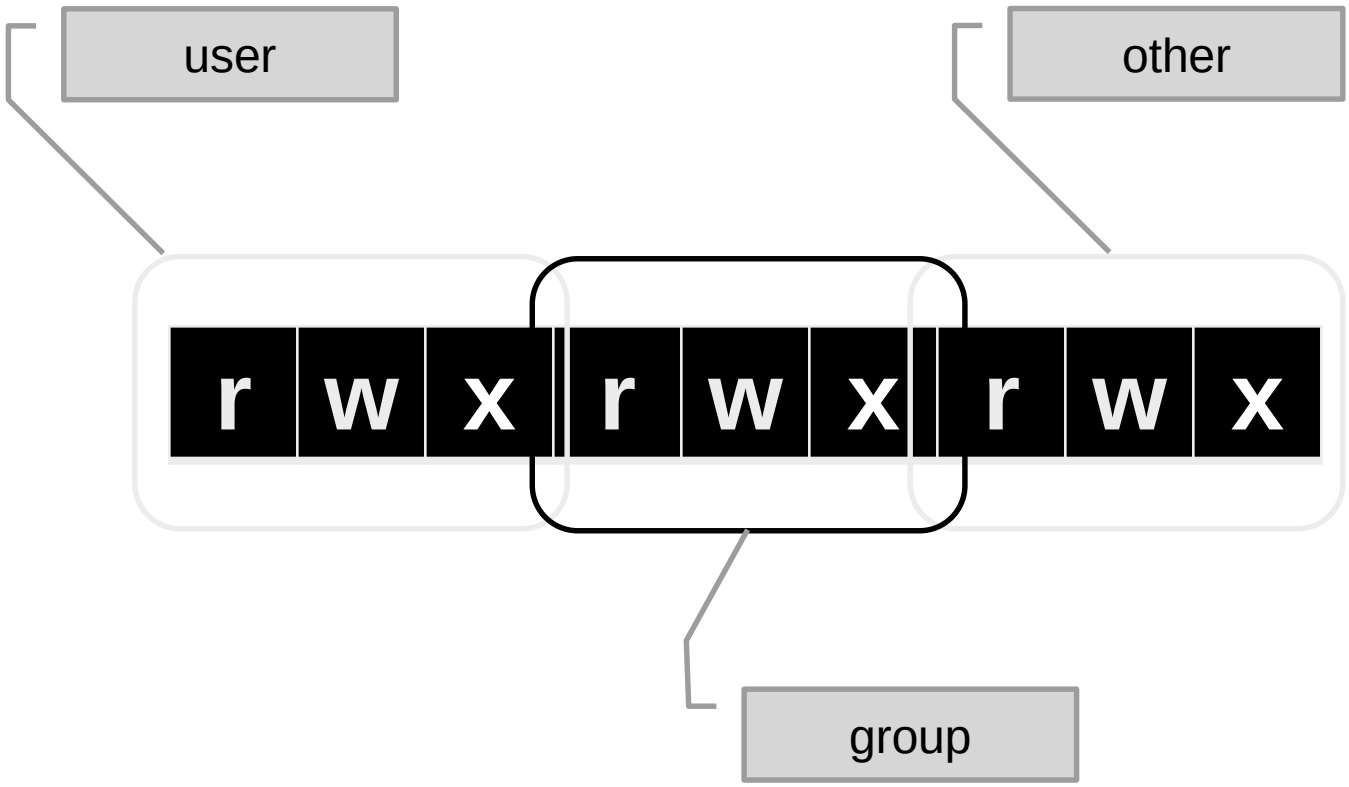
Example file permissions:

- Try doing

  `ls -ld ~/`

- The command (ls -l) shows permissions on home directory




user
other
r w x r w x r w x
group

Scripts

- Allow series of commands to be repeated
- Can pass arguments in, use variables etc.
- Need to be “executable” to run from command line
- First line shows the “interpreter” (or shell) to use, e.g.

```
#!/bin/bash
```

- Objective is to get it to run!

Scripts

1. Use a text editor, e.g. nano to create a new script file:
   - Set the shell in the first line `#!/bin/bash`
   - Use the echo command to print some text on screen eg. `echo "Hello Linux world"`
   - Save the file
2. Make it executable (`chmod`) – to get the right permissions to run it – e.g. `chmod u+x` *`scriptname`*
3. Run the script `./`*`scriptname`*

Workshop 3 – writing a program, running a script

Time: 20 minutes

- `cd /rds/projects/2017/ghumraak-bear-training/`
- View the example program **simple_script**
- Return to home directory `cd ~`
- Write a basic program to say “Hello ‘[your name]’, welcome to the Introduction to Linux workshop”
- Run the program
- Look at file permissions for the program

Example script – input1

```
#!/bin/bash

read -p 'Enter Your Name:' name
echo  "Hello $name, Welcome to the
Introduction to Linux (for using
BlueBEAR) workshop"
```

Example script – input2

```
#!/bin/bash

echo -e "What do you think of Bash?"
read reply
echo "You said $reply"
echo
```

UNIVERSITY OF BIRMINGHAM

Section 5 -

Other BEAR services and further information

Next steps – further Linux

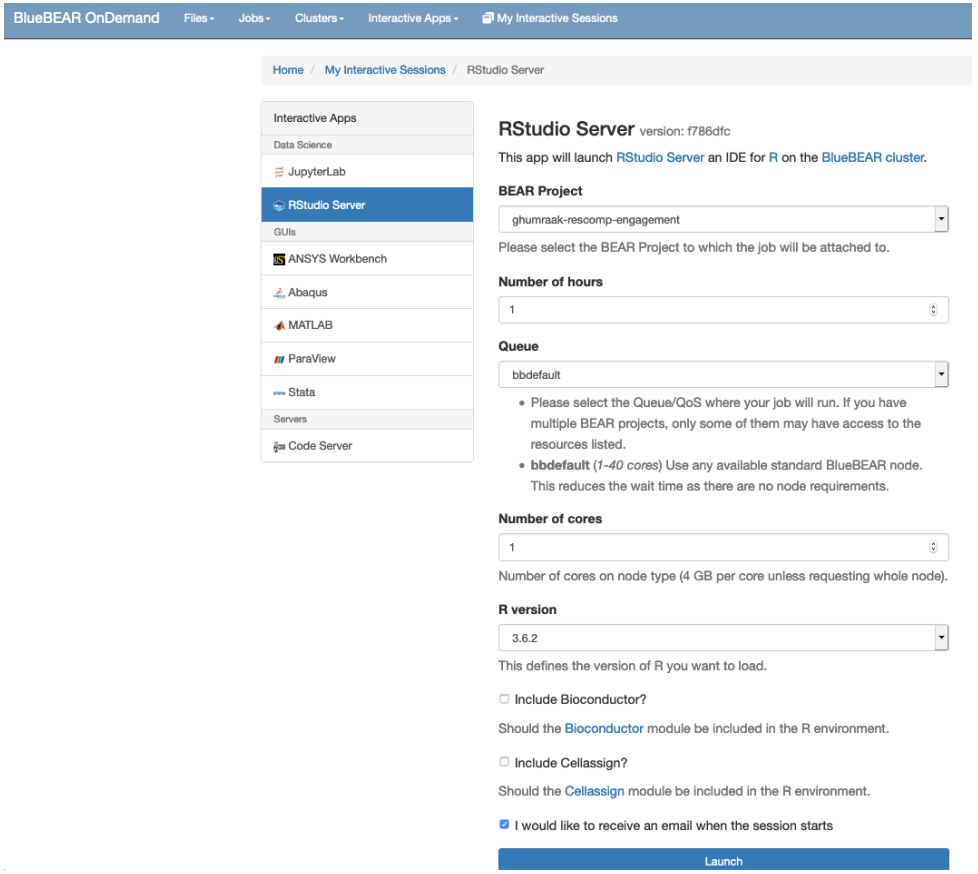
- Online resources - LinkedIn Learning Linux:
  - https://www.w3resource.com/linux-system-administration/linux-commands-introduction.php
- Software Carpentries Linux course:
  https://bham-carpentries.github.io/shell-novice/
- Introduction to BlueBEAR:
  https://intranet.birmingham.ac.uk/it/teams/infrastructure/research/bear/bear-training/necessities.aspx
- BEAR Training Page:
  https://intranet.birmingham.ac.uk/bear-training

Other BEAR Services

- **Research Data Store (RDS):** FREE storage for research projects (up to 3TB per project)
- **BEAR DataShare:** file synchronisation and sharing service – like Dropbox
- **BEAR Large Data Transfer**: Ability to share and receive large amounts of data
- **BEAR GitLab**: version control
- **BEAR Cloud:** local high-performance cloud computing integrated with campus services
- **BEAR Software:** free advice/help from BEAR Research Software Engineers
- **Training**: Software Carpentry – Python, R, MATLAB; deep learning, C++
- ... and more at https://intranet.birmingham.ac.uk/bear

BlueBEAR Portal

- https://intranet.birmingham.ac.uk/bear/portal
- Pilot service allowing web-based, graphical interface access to a limited but expanding no. of applications including:
  - JupyterLab
  - RStudio
  - MATLAB
  - Stata
- Still need access to a BlueBEAR project
- Needs Remote Access VPN when off-campus



Special Interest Groups

- SIGs for (currently):
  - CCB Seminar (Bioinformatics)
  - Computational Fluid Dynamics (CFD)
  - Finite Element Method (FEM)
  - MATLAB
  - Stata
  - Materials Simulation and Modelling

https://intranet.birmingham.ac.uk/collaboration/hpc-research/index.aspx

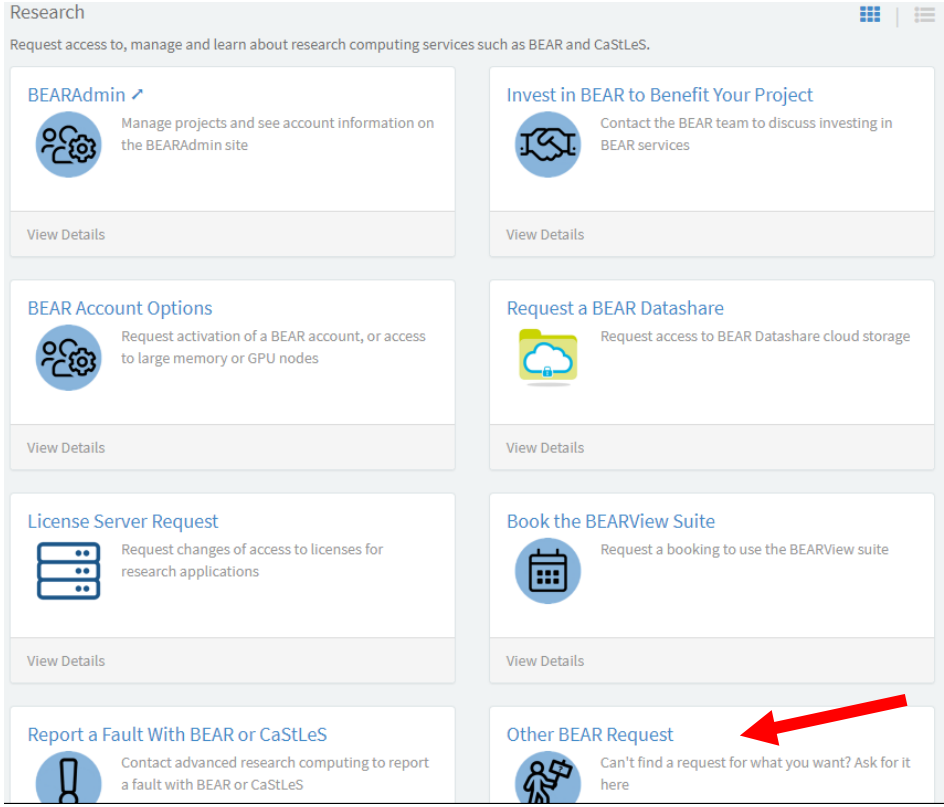
Help is available

If you're having any problems visit the IT Service Desk in a web browser:
https://universityofbirmingham.service-now.com/

Canvas course – self-register –
https://canvas.bham.ac.uk/enroll/6MPPJW

Regular drop-in sessions –
https://intranet.birmingham.ac.uk/bear-drop-in

Join our mailing list –
Email bearinfo@contacts.bham.ac.uk



Questions?