Intro to Linux Command-Line: View, Find, and Search

Introduction: Introduce how to work with, view, find and search the content of text files. The workshop is aimed at beginners with basic command-line experience of the Linux file system and will focus on hands-on exercises.

Course Goals

- View the content of text-based files.
- Search a file for a string.
- Search for a file/folder by name.
- Redirect output from commands and pipe commands together.
- Be very exercise based to allow practice of commands and concepts.

Sections

1. View and Search a File: Use various commands to view, inspect and search a text file.
2. Search for a File: How to search for a file, by name, across a hierarchy of folders.
3. Output Redirection and Pipes: Introduce how to pipe the output of one command into the input of another, and how to redirect the output of a command into a file.
4. Intermediate Features and Summary:

View and Search Text and Directories

Goal: Use various commands to view, inspect and search a text file.

- Setting Up
- View the content of files
- Exercises
- Search for a string within a text file (grep)
- Examples: Search a file
- Exercises
- Next Steps

Setting Up

Question:

- There is a folder called `intro_to_linux` within the `/project/arccatrain/` folder.
- How would you copy this folder into your home folder?

Setting Up: Answer(s)

Answer: There are a number of ways…

```
# Move to your home folder and copy int this location.
# The “.” marks the current working directory.
[arccanetrain]$ cd
[~]$ cp -r /project/arccanetrain/intro_to_linux/ .
cp: cannot open 'intro_to_linux/workshop_me.txt' for reading: Permission denied
[~]$ ls
Desktop  Documents  Downloads  intro_to_linux
# Move into the /project/arccatrain/ folder and copy from there into your home.
# The “~” is short for your home folder.
[~]$ cd /project/arccanetrain/
[arccanetrain]$ cp -r intro_to_linux/ ~
cp: cannot open 'intro_to_linux/workshop_me.txt' for reading: Permission denied
```

```
[arccanetrain]$ ls ~
Desktop  Documents  Downloads  intro_to_linux
# Why do we see the cp related permission denied?
# -rw-------  1 arcc-t05 arccanetrain      23 Oct  5 07:20  workshop_me.txt
# What happens to this file? It does not get copied.
```

View the content of files

Command	Description
cat	Usage: cat [OPTION]... [FILE]... Concatenate FILE(s) to standard output. -n, --number number all output lines
more	more [options] <file>... A file perusal filter for CRT viewing. more is a filter for paging through text one screenful at a time.
head	Usage: head [OPTION]... [FILE]... Print the first 10 lines of each FILE to standard output. -n, --lines=[-]NUM print the first NUM lines instead of the first 10; with the leading '-', print all but the last NUM lines of each file

View the content of files

Command	Description
tail	Usage: tail [OPTION]... [FILE]... Print the last 10 lines of each FILE to standard output. -f, --follow[={name|descriptor}] output appended data as the file grows; an absent option argument means 'descriptor' -n, --lines=[+]NUM output the last NUM lines, instead of the last 10; or use -n +NUM to output starting with line NUM

Exercises

```
[]$ cd ~/intro_to_linux/
[intro_to_linux]$ cat software.csv
[intro_to_linux]$ cat -n software.csv
# Press spacebar to scroll through.
# Press 'q' to quit at any time.
[intro_to_linux]$ more software.csv
[intro_to_linux]$ head software.csv
[intro_to_linux]$ head -n 5 software.csv
[intro_to_linux]$ tail software.csv
[intro_to_linux]$ tail -n 5 software.csv
```

Search for a string within a text file (grep)

Command	Description

grep	`Usage: grep [OPTION]... PATTERN [FILE]...`
	`Search for PATTERN in each FILE.`
	`Example: grep -i 'hello world' menu.h main.c`
	`...`
	` -i, --ignore-case  ignore case distinctions`
	`...`
	` -n, --line-number  print line number with output lines`
	`...`
	` -r, --recursive  like --directories=recurse`
	`...`
	`# grep is case-sensitive`

Examples: Search a file

```
# Remember: grep is case-sensitive
```

```
[intro_to_linux]$ grep NVIDIA software.csv
libraries and toolkits,cuDNN,cudnn,beartooth,The NVIDIA CUDA Deep...
libraries and toolkits,TensorRT,,beartooth,"NVIDIA TensorRT, an...
# Nothing is returned.
[intro_to_linux]$ grep nvidia software.csv
[intro_to_linux]$
# Neither of the above picked up “Nvidia”.
[intro_to_linux]$ grep -i NVidia software.csv
compiler,NVidia HPC SDK,nvhpc,"beartooth,teton"...
libraries and toolkits,cuDNN,cudnn,beartooth,The NVIDIA CUDA Deep...
libraries and toolkits,TensorRT,,beartooth,"NVIDIA TensorRT, an...
# Ignore the case of the word to search for.
[intro_to_linux]$ grep -n -i NVidia software.csv
145:compiler,NVidia HPC SDK,nvhpc,"beartooth,teton"...
152:libraries and toolkits,cuDNN,cudnn,beartooth,The NVIDIA CUDA Deep...
166:libraries and toolkits,TensorRT,,beartooth,"NVIDIA TensorRT, an...
```

Examples: Search folders and files

```
[intro_to_linux]$ cd clusters/
[clusters]$ grep -i nvidia *
beartooth.html:    .../788758554/NVidia+HPC+SDK">NVidia HPC SDK</a></td>
teton.html:    .../788758554/NVidia+HPC+SDK">NVidia HPC SDK</a></td>
[clusters]$ cd ..
[intro_to_linux]$ grep -i nvidia *
grep: clusters: Is a directory
software.csv:compiler,NVidia HPC SDK,nvhpc,"beartooth,teton"...
software.csv:libraries and toolkits,cuDNN,cudnn,beartooth,The NVIDIA CUDA Deep...
software.csv:libraries and toolkits,TensorRT,,beartooth,"NVIDIA TensorRT, an...
[intro_to_linux]$ grep -r -i nvidia *
clusters/teton.html:    .../788758554/NVidia+HPC+SDK">NVidia HPC SDK</a></td>
clusters/beartooth.html:    .../788758554/NVidia+HPC+SDK">NVidia HPC SDK</a></td>
software.csv:compiler,NVidia HPC SDK,nvhpc,"beartooth,teton"...
software.csv:libraries and toolkits,cuDNN,cudnn,beartooth,The NVIDIA CUDA Deep...
software.csv:libraries and toolkits,TensorRT,,beartooth,"NVIDIA TensorRT, an...
```

Exercises

```
# The software.csv file takes the form:
[intro_to_linux]$ head software.csv
Type,Name,Module,Cluster,Description
application,Alphafold,alphafold,"beartooth,teton",AlphaFold...
application,Astral,astral,wildiris,ASTRAL is a tool...
application,Augustus,augustus,beartooth,AUGUSTUS is a program...
application,Avizo,avizo,loren-pre202308,Avizo is a general-purpose...
application,ANGSD,angsd,"beartooth,teton",ANGSD: is a software...
application,ANSYS,ansys,teton,"ANSYS is a general-purpose software...
```

Questions:

1. Which *named* applications are related to the words “bayes”?
2. Which files contain reference to IPA?

Answers

```
[intro_to_linux]$ grep -i bayes software.csv
application,Bayescan,bayescan,beartooth,"BayeScan aims...
application,Beast1,beast1,wildiris,BEAST is a cross-platform program for Bayesian...
application,Beast2,beast2,beartooth,"BEAST 2 is a cross-platform program for Bayesian...
application,Freebayes,freebayes,beartooth,"freebayes is a Bayesian genetic...
application,Jags,jags,"beartooth,teton",Just Another Gibbs Sampler. It is a program for analysis of
Bayesian hierarchical...
application,RevBayes,revbayes,wildiris,Bayesian phylogenetic...
application,ROHan,rohan,teton,"ROHan is a Bayesian framework...
application,SourceTracker2,sourcetracker2,"beartooth,teton","SourceTracker, a Bayesian approach...
[intro_to_linux]$ grep -r IPA *
clusters/beartooth.html:    .../pages/1893597185/IPA">IPA</a></td>
software.csv:application,IPA,ipa,beartooth,Improved Phased Assembler (IPA) is...
```

Output Redirection and Pipes

Goal: Introduce how to pipe the output of one command into the input of another, and how to redirect the output of a command into a file.

- Output Redirection and Pipes
- Redirection of output: > vs >>
- Example: Using pipe "|" from a file
- Example: Pipe from ls command
- Exercises
- Next Steps

Output Redirection and Pipes

- Redirection of output: > vs >>
  - A *redirect* sends a channel of output to a file.
  - You can redirect a file as input to a command using < and << (not looked at).
- Using pipe "|'
  - A *pipe* passes *standard output* as the *standard input* to another command
- Examples of the form:
  - View a text file and pipe to grep.
  - Cat a list and sort by line.
  - Sort and then find unique items.
  - View folder contents and look for a specifically named name.

Redirection of output: > vs >>

```
# Writes out to the command line.
[intro_to_linux]$ grep -i bayes software.csv
# Redirects the output to a file called apps.txt
[intro_to_linux]$ grep -i bayes software.csv > apps.txt
[intro_to_linux]$ ls
apps.txt  clusters  data  software.csv
[intro_to_linux]$ cat apps.txt
# Overwrites any existing file called apps.txt
[intro_to_linux]$ grep -i IPA software.csv > apps.txt
[intro_to_linux]$ cat apps.txt
[intro_to_linux]$ rm apps.txt
# Overwrites existing apps.txt
[intro_to_linux]$ grep -i bayes software.csv > apps.txt
# Appends to the existing file.
[intro_to_linux]$ grep -i IPA software.csv >> apps.txt
```

Example: Using pipe “|” from a file

```
[intro_to_linux]$ cat fruits.txt
Gooseberry
Apple
Apricot
Avocado
Strawberry
...
[intro_to_linux]$ cat fruits.txt | wc -l
97
```

Example continued

```
# The order of items is the same as listed within the fruits.txt file.
[intro_to_linux]$ cat fruits.txt | grep berry
```

```
Gooseberry
Strawberry
Bilberry
Blackberry
Marionberry
Blueberry
Boysenberry
Gooseberry
Cloudberry
Elderberry
Goji berry
Honeyberry
Juniper berry
Cranberry
Cranberry
Marionberry
Gooseberry
Mulberry
Salmonberry
Huckleberry
Raspberry
Salal berry
```

Example continued

```
# Notice the duplicates.
[intro_to_linux]$ cat fruits.txt | grep berry | sort
Bilberry
Blackberry
Blueberry
Boysenberry
Cloudberry
Cranberry
Cranberry
Elderberry
Goji berry
Gooseberry
Gooseberry
Gooseberry
Honeyberry
Huckleberry
```

```
Juniper berry
Marionberry
Marionberry
Mulberry
Raspberry
Salal berry
Salmonberry
Strawberry
```

Example continued

```
# Duplicates have been removed leaving only the unique names.
[intro_to_linux]$ cat fruits.txt | grep berry | sort | uniq
Bilberry
Blackberry
Blueberry
Boysenberry
Cloudberry
Cranberry
Elderberry
Goji berry
Gooseberry
Honeyberry
Huckleberry
Juniper berry
Marionberry
Mulberry
Raspberry
Salal berry
Salmonberry
Strawberry
[intro_to_linux]$ cat fruits.txt | grep berry | sort | uniq | wc -l
18
```

Example continued

```
[intro_to_linux]$ cat fruits.txt | grep berry | sort | uniq > berries.txt
[intro_to_linux]$ cat berries.txt
Bilberry
Blackberry
Blueberry
```

```
Boysenberry
Cloudberry
Cranberry
Elderberry
Goji berry
Gooseberry
Honeyberry
Huckleberry
Juniper berry
Marionberry
Mulberry
Raspberry
Salal berry
Salmonberry
Strawberry
[intro_to_linux]$ cat berries.txt | wc -l
18
```

Example: Pipe from ls command

```
[intro_to_linux]$ ls -R
[intro_to_linux]$ ls -R | grep "Feb"
February
./data/2022/February:
Feb
./data/2023/Feb:
[intro_to_linux]$ ls -R | grep -i "Feb"
feb
./data/2021/feb:
february_01_2021.tx
February
./data/2022/February:
Feb
./data/2023/Feb:
```

Exercises

1. How does the `wc` command work? What are its options?
2. How does the `sort` command work? What are its options?
3. How does the `uniq` command work? What are its options?
4. How many unique varieties of beans are there in the `vegetables.txt` file?

Answers

4: How many unique varieties of beans are there in the vegetables.txt file?

- How do you deal with “soy beans” vs “Soy Beans”?
- What options does the `uniq` command provide?

```
[intro_to_linux]$ cat vegatables.txt | grep -i beans | sort | uniq -i | wc -l
12
```

Intro to Linux Command-Line: Intermediate Features and Summary

---

- More Intermediate Features
- Next Steps, Suggestions
- Further Trainings: UWYO LinkedIn
- Request an Account with ARCC
- Summary

- [ ] Next Steps

More Intermediate Features

- [ ] Environment Variables: Define the behavior of the environment: Try:
  - `echo $HOME`
  - `echo $USER`
  - `echo $SHELL`
  - `echo $PATH`
- [ ] File searching/manipulation
  - `sed: stream editor for filtering and transforming text`
  - `gawk: pattern scanning and processing language`
- [ ] Ability to update file permission and ownership: `chmod`/`chown`
  - User-case of sharing files/folders.

More Intermediate Features

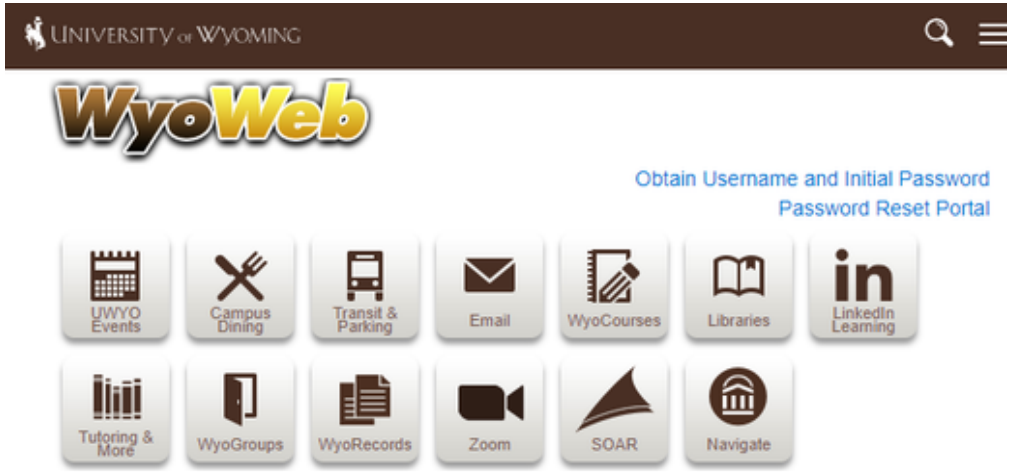
- [ ] Aliases in `.bashrc`.
  - Create short-cuts of popular/frequently used commands.
- [ ] Text editors: vi/vim/nano
  - `vimtutor`
  - `touch`
- [ ] Remote access with ssh.

Next Steps, Suggestions

- [ ] Next Steps on using Linux:

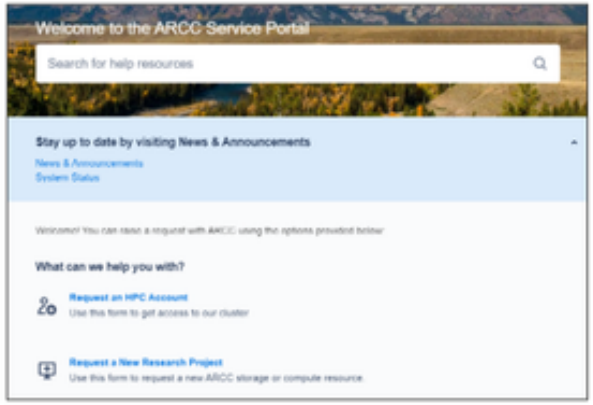
    - Practicing using Linux online.
        - Tutorialspoint: Online Linux Terminal
        - Best Online Linux Terminals and Online Bash Editors
- ☐ Dual boot a Windows machine with Linux.
- ☐ Run a container image.
    - Run Linux containers on Windows
    - How to dual-boot Linux and Windows
- ☐ UW Researcher? Create a project on the Beartooth cluster with your PI.

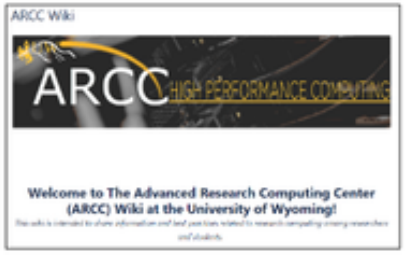
Further Trainings: UWYO LinkedIn



- ☐ Introduction to Linux
- ☐ Learning Linux Command Line
- ☐ Linux: Files and Permissions
- ☐ Linux: Over and Installation
- ☐ Learning Linux Shell Scripting

Request an Account with ARCC





- Web: https://www.uwyo.edu/arcc/
- Wiki: https://arccwiki.atlassian.net/wiki/spaces/DOCUMENTAT
- Portal: https://arccwiki.atlassian.net/servicedesk/customer/portals

Summary

In this workshop we have:

- How to search for a string within a file.
- How to find a file.

- How to redirect the output of a command into a file.
- How to use pipes to direct the output of one command as the input into another command.

Next Steps

Previous	Workshop Home
Output Redirection and Pipes	Intro to Linux CLI: View, Find, and Search

Use the following link to provide feedback on this training: https://forms.gle/6hRPPabA87CWncEu8 or use the QR code below.