LINUX

AN INTRODUCTION TO THE BASH SHELL

LINUX

AN INTRODUCTION TO THE BASH SHELL

also known as the "command-line "

USER INTERFACES

The shell is an interface that gives you access to the operating system's services.

USER INTERFACES

The shell is an interface that gives you access to the operating system's services.

1. Command-Line Interface (CLI or shell)

USER INTERFACES

The shell is an interface that gives you access to the operating system's services.

1. Command-Line Interface (CLI or shell)
2. Graphical User Interface (GUI)

WHY USE THE COMMAND-LINE?

WHY USE THE COMMAND-LINE?

- Some high-performance platforms (e.g. NeSI) provide no GUI

WHY USE THE COMMAND-LINE?

- Some high-performance platforms (e.g. NeSI) provide no GUI
- Point-and-click/drag-and-drop does not scale.

WHAT WE'LL COVER

WHAT WE'LL COVER

- navigation

WHAT WE'LL COVER

- navigation
- basic file access

WHAT WE'LL COVER

- navigation
- basic file access
- customizing your linux session

Other sessions covering linux commandline topics:

1. Bash scripting for researchers Thu, November 26, 9:30 AM – 11:00 AM
2. Bash shell - some basic tools Thu, November 26, 4:00 PM – 5:00 PM

THE UNIX PHILOSOPHY

- simple tools
- each doing one job well
- compose them in a pipeline and you have a powerful language.

THE SHELL PROMPT

The bash shell prompt is typically

```
$
```

If you see that prompt in these notes,
type *what follows* at the command-line on your terminal.
Then press enter.

THE SHELL PROMPT

The bash shell prompt is typically

```
$
```

If you see that prompt in these notes,
type *what follows* at the command-line on your terminal.
Then press enter.

```
$ ls
```

THE SHELL PROMPT

The bash shell prompt is typically

```
$
```

If you see that prompt in these notes,
type *what follows* at the command-line on your terminal.
Then press enter.

```
$ ls
```

```
$ pwd
```

COMMENTS

COMMENTS

The character "#" denotes the beginning of a comment.

COMMENTS

The character "#" denotes the beginning of a comment.

```
$ ls   # list contents of current directory
```

COMMENTS

The character "#" denotes the beginning of a comment.

```
$ ls   # list contents of current directory
```

```
$ pwd  # print present working directory
```

COMMENTS

The character "#" denotes the beginning of a comment.

```
$ ls   # list contents of current directory
```

```
$ pwd  # print present working directory
```

Don't type the # sign or what follows.

GETTING SET UP

```
$ cd
$ mkdir -p resbaz_2020/commandline
$ cd resbaz_2020/commandline
```

Hint: Use the tab key to autocomplete pre-existing names.

CREATING/REMOVING DIRECTORIES

```
$ mkdir                         # make directory
$ rmdir                         # remove empty directory
```

CHANGING DIRECTORIES

```
$ cd                            # change directory
```

EXAMPLE

WORKING WITH DIRECTORIES

```
$ cd ~/resbaz_2020/commandline
$ mkdir new_dir                 # make directory
$ cd new_dir                    # change directory
$ touch new_file                # create empty file
$ ls
$ cd ..                         # change directory up one level
$ ls -lR                        # list all files recursively
$ rm new_dir/new_file           # remove file
$ rmdir new_dir                 # remove empty directory
$ ls -lR                        # list all files recursively
```

WORKING WITH FILES

```
$ touch filename   # create empty file
$ editor filename  # (create file and) edit it
$ rm filename      # remove file
```

EXAMPLE

WORKING WITH FILES

```
$ cd ~/resbaz_2020/commandline
$ mkdir dir_a      # create dir_a
$ ls               # directory listing
$ cd dir_a         # move to dir_a
$ editor file_b    # (create and) edit file_b
$ cat file_b       # show contents of file_b
$ ls               # directory listing
$ ls -lt           # full directory listing
$ cd ..            # cd back one directory
```

SYSTEM DOCUMENTATION

SYSTEM DOCUMENTATION

```
$ man ls      # read the manual on ls
```

SYSTEM DOCUMENTATION

```
$ man ls      # read the manual on ls
```

```
$ ls --help  # alternative help source on ls
```

SYSTEM DOCUMENTATION

```
$ man ls      # read the manual on ls
```

```
$ ls --help  # alternative help source on ls
```

E.G. TO READ UP ON FILE EDITORS:

```
$ editor --help
$ vim --help
$ man vim
```

THE MANUAL IS YOUR FRIEND

- when you know what you're looking for, use the manual
- when you don't, use "apropos"

```
$ man apropos
apropos - search the manual page names and descriptions

DESCRIPTION
 Each manual page has a short description available within it.
 apropos searches the descriptions for instances of keyword.
```

Use apropos to find the command you need. E.g. what editors are available?

```
$ apropos editor
```

PIPES - CAPTURING OUTPUT

The pipe "|" takes the output of one command and sends it as input to the next.

```
$ ls -lt         | head                  # long listing of newest files
$ cat long_file | head                  # first 10 lines of long_file
$ cat long_file | tail                  # last 10 lines of long_file
$ cat long_file | head -n 1 > header    # first line of long_file in new fi
$ history | tail                        # 10 most-recent commands
```

USING BASH HISTORY

```
$ history                    # show command history
$ history | grep filename    # search command history for string "filename"
```

USING BASH HISTORY

```
$ history                    # show command history
$ history | grep filename    # search command history for string "filename"
```

- To page through all recent commands, use the up-arrow key

USING BASH HISTORY

```
$ history                      # show command history
$ history | grep filename  # search command history for string "filename"
```

- To page through all recent commands, use the up-arrow key
- To find most recent commands containing certain text, type

```
$ Ctrl-r your_text_here
```

Continue typing Ctrl-r to page through these commands.

SHORT CUTS

```
$ alias                                 # what aliases you have
$ alias scut ="long linux command" # alias "scut" for this sessio
```

THE SHELL - CUSTOMIZING YOUR LINUX ENVIRONMENT

- The shell uses many environment variables
- Their names are upper-case
- Their values are accessed by putting a $ in front.

```
$ env                 # see your environment variables
$ echo $USER          # your login name
$ echo $PATH          # where your system looks for your commar
$ cat .bashrc         # where you can configure your shell (lir
$ cat .bash_profile # where you can configure your shell (mac
$ alias               # what aliases are registered
```

The history command tells you what commands you've issued in the past:

```
$ history | tail -n 5  # for the last 5 commands used
```

The history command tells you what commands you've issued in the past:

```
$ history | tail -n 5  # for the last 5 commands used
```

Configure the history command with:

```
$ export HISTTIMEFORMAT="%F %T "
```

The history command tells you what commands you've issued in the past:

```
$ history | tail -n 5  # for the last 5 commands used
```

Configure the history command with:

```
$ export HISTTIMEFORMAT="%F %T "
```

then do some stuff ..., and check your history again

```
$ history| tail -n 5
```

The format for history you will then see is:

```
$ history |tail -n 5
2020-11-23 13:39:52 man vim
2020-11-23 13:40:58 apropos edit
2020-11-23 13:41:22 apropos editor
2020-11-23 13:41:46 vi index.html
2020-11-23 13:55:21 history|tail
```

The format for history you will then see is:

```
$ history |tail -n 5
2020-11-23 13:39:52 man vim
2020-11-23 13:40:58 apropos edit
2020-11-23 13:41:22 apropos editor
2020-11-23 13:41:46 vi index.html
2020-11-23 13:55:21 history|tail
```

You get a timestamp associated with each command. Very handy!

ASIDE

To make this configuration permanent, append the command

```
$ export HISTTIMEFORMAT="%F %T "
```

to your bash configuration file

- ~/.bashrc (linux)
- ~/.bash_profile (MacOS) , and source that file to activate the setting:

```
$ source ~/.bashrc #or source ~/.bash_profile
```

ASIDE CONTINUED

The settings in this configuration file are applied to every terminal shell you open. Add more settings as you like.

Many utilities/programs you install have their own configuration files. They are usually *hidden* in your home-directory - they start with a dot, and aren't listed by default. To see them:

```
$ ls -lat       # everything
$ ls -lat .*    #just hidden (dot) files
```

These are your personal configuration files.

RESOURCES:

- the linux documentation project

THE END!!

YOU'RE INVITED

- Lunch and Listen Sessions .
- Attendance is free, but you must enrol.