Linux and Beyond

Prepared By: Aatiz Ghimire, for Herald Center for AI.

Summer, 2025

1 Learning Objectives.

- Understand Linux filesystem navigation and user permissions.
- Gain hands-on experience in compiling software from source.
- Develop basic and advanced Bash scripts.
- Learn resource monitoring and lightweight system diagnostics.
- Build a script for real-time system data logging.

---

2 Follow Along

Folder Navigation

Command	Description
`sleep 60`	Sleep for 60 seconds.
CTRL + C	Interrupt the command.
`cd`	Switch to your home directory.
`pwd`	Print the current working directory.
`ls`	List directory contents.
`ls -a`	List all files including hidden ones.
`ls -l`	List files in long format.
`ls -la`	List all files in long format.
`mkdir shell-ex`	Create folder named `shell-ex`.
`cd shell-ex`	Enter the `shell-ex` directory.
`mkdir "delete me"`	Create a folder named `delete me`.
`rmdir delete me`	Tries to remove directories `delete` and `me` separately (will fail).
`rmdir "delete me"`	Correctly removes the folder named `delete me`.
`ls -a ..`	List all files in the parent directory.

Table 1:

Help Commands

Command	Description
`mkdir --help`	Display help for the `mkdir` command with usage and options.
`man mkdir`	Open the manual page for the `mkdir` command.
q	Quit the pager when reading a man page.
`man --help`	Show usage options for the `man` command itself.
`man -h`	Another form to get usage info for `man`.
`man man`	Open the manual for the `man` command.
`whatis man`	Show a short description of what `man` does.
`man 7 man`	Open section 7 of the manual for `man` (conventions, macros, etc.).

Table 2:

Permissions

Command	**Description**
`cd /shell-ex`	Navigate to shell-ex directory
`mkdir perm-ex`	Create a folder named perm-ex
`cd perm-ex`	Enter the perm-ex directory
`touch file.txt`	Create an empty file file.txt
`mkdir folder`	Create a subdirectory named folder
`ls -la`	List all files and folders with permissions
`chmod a-r file.txt`	Remove read permission for all users on file.txt
`chmod a-r folder`	Remove read permission for all users on folder
`ls -la`	Check updated permissions
`ls folder`	Try to list contents of folder (expected to fail)
`cat file.txt`	Try to read file.txt (expected to fail)
`touch folder/newfile.txt`	Still possible to create new files
`chmod a= file.txt`	Remove all permissions on file.txt
`chmod a= folder`	Remove all permissions on folder
`rm file.txt`	Try deleting file.txt (expected to fail)
`rmdir folder`	Try deleting folder (expected to fail)
`chmod u+r folder`	Add read permission back for user
`ls folder`	Attempt to list folder contents
`touch folder/file2.txt`	Try to create a new file (expected to fail)
`chmod u+w folder`	Add write permission to folder
`touch folder/file2.txt`	Try again to create a file (still fails)
`chmod u+x folder`	Add execute permission (needed for access)
`touch folder/file2.txt`	Successfully create file after execute permission

Table 3:

Nano Editor

Command	**Description**
`cd ~/shell-ex`	Navigate to the shell-ex directory
`mkdir nano-ex`	Create the nano-ex folder
`cd nano-ex`	Enter the nano-ex directory
`nano`	Start editing in a new nano buffer
`Write some text and one long line`	Compose content in the editor
CTRL + O	Save the file (name it `file.txt`)
CTRL + X	Exit nano
`cat file.txt`	View contents of the saved file
`nano file.txt`	Reopen the file for editing
`Make a change`	Edit the file content
CTRL + X	Attempt to exit nano
`Prompt:` y/n + ENTER	Respond to the save prompt on exit

Table 4:

File and Folder Operations

Command	**Description**
`cd ~/shell-ex`	Change to the shell-ex directory
`mkdir operations-ex`	Create directory named operations-ex
`cd operations-ex`	Enter operations-ex
`mkdir folder`	Create a subdirectory named folder
`touch file`	Create an empty file named file
`mv file folder`	Move file into the folder directory
`mv folder/file file.txt`	Move file out and rename it to file.txt
`cp file.txt folder`	Copy file.txt into folder
`cp folder folder2`	Attempt to copy a folder (fails without -R)
`cp -R folder folder2`	Recursively copy folder to folder2

Table 5:

Environmental Variables

Command	**Description**
`echo $HOME`	Print current user's home directory
`echo $PWD`	Print current working directory
`echo $PATH`	Display current PATH variable
`echo -e $PATH//:/:\n`	Show PATH entries each on a new line
`printenv`	List all environment variables
`export HELLO=hello`	Create an environment variable
`echo $HELLO`	Print the value of HELLO
`export HELLO="$HELLO world"`	Append to the variable HELLO
`echo "$HELLO"`	Print expanded value
`echo '$HELLO'`	Print literal string without expansion
`unset HELLO`	Remove the environment variable HELLO
`echo $HELLO`	Verify that HELLO has been unset
`nano ~/.bash_profile`	Open bash profile for editing
`Add: [[ -f ~/.bashrc ]] && . ~/.bashrc`	Ensure .bashrc is sourced from .bash_profile
`Save and exit nano & Save your edits and exit`	
`nano ~/.bashrc`	Open .bashrc for editing
`Add:` `export HELLO=hi`	Add a default HELLO variable
`Add:` `alias ll='ls -la'`	Create alias ll for detailed listing
`Save and exit nano`	Save your edits and exit
`source ~/.bashrc`	Apply the new changes
`ll`	Use the new alias
`echo $HELLO`	Verify HELLO variable from .bashrc

Table 6:

Processes

Command	**Description**
`top`	Get an overview of current resource usage
`htop`	Get a better overview (with UI) of current usage
`ps`	List all your currently running processes
`ps -ef`	List all system-wide running processes
`sleep 60 &`	Run `sleep` in background
`kill PID`	Kill a process by PID (returned from `ps` or `sleep`)
`export HELLO2=hi2 && echo $HELLO2 && unset HELLO2 && echo $HELLO2`	Chain multiple commands using `&&`

Table 7:

Reading and Searching

Command	**Description**
`cd ~/shell-ex`	Change to the shell-ex directory
`mkdir read-search-ex`	Create read-search-ex directory
`cd read-search-ex`	Move into the new directory
`man man > man.txt`	Redirect the manual page of `man` to a text file
`head man.txt`	View the first 10 lines of `man.txt`
`tail man.txt`	View the last 10 lines of `man.txt`
`head -n 20 man.txt`	View the first 20 lines of `man.txt`
`grep manual man.txt`	Show all lines containing "manual"
`grep -c manual man.txt`	Count occurrences of "manual"
`grep -wc manual man.txt`	Count occurrences of "manual" as a whole word
`cp man.txt man2.txt`	Copy `man.txt` to `man2.txt`
`nano man2.txt`	Open `man2.txt` for editing
Edit and save file	
`diff man.txt man2.txt`	See differences between original and edited file
`wc man.txt`	Show line, word, and byte count of `man.txt`

Table 8:

Redirection and Piping

Command	**Description**
`mkdir ~/shell-ex/redirect-ex && cd ~//shell-ex/redirect-ex`	Create and move into directory `redirect-ex`
`ps -ef > p.txt`	Redirect output of `ps` to file `p.txt`
`echo $HOME >> p.txt`	Append `$HOME` value to file `p.txt`
`tail p.txt`	View the last lines of `p.txt`
`ps -ef | grep ssh`	Pipe `ps` output to `grep` to search for `ssh`
`ps -ef | grep -wc root`	Count the number of `root` processes
`ps -ef | grep root | sort -nk 2 | head`	Show first 10 root processes sorted by PID
`ps -ef | head -1; ps -ef | sort -r -nk 3 | head -15`	Show top 15 CPU-consuming processes
`!!`	Re-run the last command
`echo "alias bycpu='!!'" >> ~/.bashrc`	Create alias `bycpu` for previous command
`source ~//.bashrc`	Reload bash configuration
`bycpu`	Test the new alias `bycpu`

Table 9:

Bash History

Command	Description
`history`	View your command history
`history | grep ps`	Find all commands including `ps`
`history | less`	Open history in a pager
`!NUMBER`	Execute a specific command from history using its number
`!ps`	Run the last used command starting with `ps`
`!?grep`	Run the last used command containing `grep`

Table 10:

wget & curl

Command	Description
`mkdir ~/shell-ex/wget-curl-ex && cd !# 1`	Create directory and switch to it; `!# 1` evaluates to the second word of previous command
`wget heraldcollege.edu.np`	Downloads the HTML content into `index.html`
`wget -O herald.html heraldcollege.edu.np`	Saves the downloaded content as `herald.html`
`curl heraldcollege.edu.np`	Prints the content to the shell (does not follow redirects)
`curl -L heraldcollege.edu.np`	Follows redirects automatically
`curl -Lo herald2.html heraldcollege.edu.np`	Follows redirect and saves to `herald2.html`
`lynx heraldcollege.edu.np -dump | less`	Uses text-based browser to view content in a pager
`tar -cvzf herald.html.tar.gz herald.html`	Create tarball archive
`rm herald.html`	Remove original HTML file
`tar -xvzf herald.html.tar.gz`	Extract tarball archive
`zip herald.html.zip herald.html`	Create ZIP archive
`rm herald.html`	Remove HTML file again
`unzip herald.html.zip`	Extract ZIP archive
`ls -alh`	Show file sizes in human-readable form

Table 11:

Further Reading

- *Advanced Programming in the UNIX Environment*, 3rd Edition, by R. Stevens and S. Rago

Task 2: Bash Scripting Basics

1. `cd` *Switch to home directory*
2. `mkdir script-ex && cd !#:1` *Create a folder for the exercises*
3. `nano first.sh`

```
#!/usr/bin/bash
echo "Hello World!"
```

4. Save and exit nano.
5. `chmod u+x first.sh` *Add execution permission*
6. `./first.sh` *Run it*

Further Reading: `https://linuxhint.com/30_bash_script_examples/`

Task 3: System Script

Create a bash script that gives an overview of the current system and its resource usage. Add the script as an alias to your `.bashrc`.

Include Outputs From:

- `hostname`
- `uptime`
- `uname -r`
- `arch`
- `w`
- `free`
- `hostnamectl`
- `lscpu`
- `hostname -I`

Hints:

- Use `https://www.shellcheck.net/` to validate syntax.
- `echo -e` enables escape characters like `\t`.
- `echo -e "Date: 'date'"` executes inline commands.
- `w` cut -d ' ' -f1— lists users.
- Colored output example:

```
RED='\033[0;31m'
NC='\033[0m'
echo -e "Default ${RED}colored text${NC}Blank text"
```

- Functions:

```
function name(){
  # Function code
}
$(name)
```

The - End