NHR FAU

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

PTfS-CAM

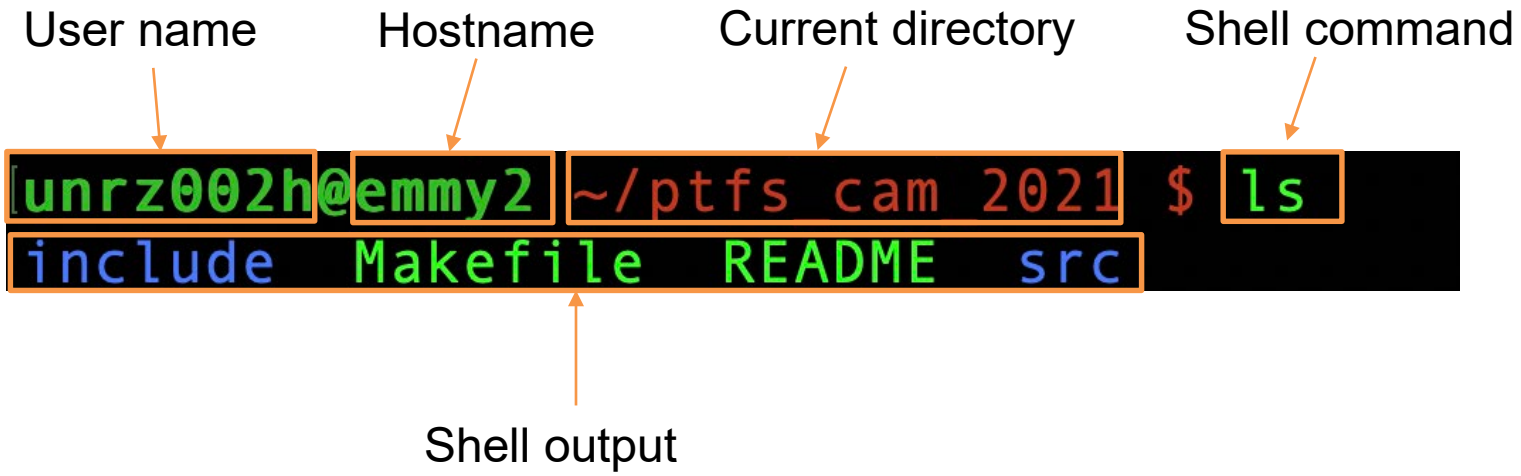
Shell basics



Shell

What is a shell?

- Text oriented (command line) mediator between user and system.
- Waits for commands, interprets them and starts programs.
- Supervises and controls program execution (e.g., scripts).



Shell commands

Some **basic** shell commands you should know

- `echo` –prints the string it is being passed
- `ls` – list all files and folders
- `cd` – change directory
- `cp` – copy files and folders (use `-r` for folders)
- `mv` – move files and folders
- `mkdir` – making a new directory
- `rm` – remove files and folders (use `-r` for folders)
- `ssh` – connect and execute commands on remote host
- `scp` – copy files to or from remote host (use `-r` for folders)

There are many many commands available. Google is your friend.

Environment variables

- Name values pairs in a program's environment.
- They can control the program's behaviour.

```
unrz002h@emmy2 ~/ptfs_cam_2021 $ echo $HOME
/home/hpc/unrz/unrz002h
```

- See defined environment variables using `env` shell command.
- To set environment variables use:
  - `export VARNAME=<value>`
  - `setenv VARNAME <value>` (on csh shell)

Scripts

- Shell scripts are a sequence of commands written to a file.
- Using scripts allows you to automatize tasks.

Example file script.sh:

```
#!/bin/bash
echo "Your home is ${HOME}"
echo "Your second argument is $2 and first argument is $1."
```

- Now convert the script.sh file to an executable using `chmod` command.

  `chmod u+x` script.sh
- Run the script. Pass in arguments if required.

  `./script.sh abc 1`

Scripts

- Use scripts wisely.
- Use scripts to run and collect results. This avoids human errors and documents your run settings.
- There are control flow loops/statements like `for, while, if`
- **Always validate the correctness of your script.**
- Output redirection (> and >>) can be handy.
- You can use pipe operator (|) to pass the output of one command to another command.
- Some of my favorite shell commands for string parsing : `grep, cat, cut, find, head, tail, bc.`

Hands on: running and collecting results

- Copy the program (square) from /home/hpc/unrz/unrz002h/ptfs_cam_2022/shell/square
- The program squares a number and returns the results.

```
unrz002h@emmy1 ~/ptfs_cam_2021 $ ./square 4
Lots of
lines
of text
Square = 16
```

- Now use shell script to collect the squares for 1…10 and report the result in a csv file format such that:

```
#val, square
   1, 1
   2, 4
   3, 9
…
```

Hands on: Solution

```
#!/bin/bash
echo "#val, square" > out.csv
for(( i=1; i<=10; i++ )); do
        ./square $i > tmp.txt
        result=$(cat tmp.txt | grep "Square =" | cut –d"=" -f2)
        echo "$i, ${result}" >> out.csv
done
rm tmp.txt
```

Coming up next

Basic on C/C++ programming