

CS 279 - Week 7 Lab Exercise

Deadline

Due by the end of lab on 2022-10-06.

How to submit

Submit the files specified below on <https://canvas.humboldt.edu>.

Purpose

To set up a `bin` directory and `.bash_profile` and `.bashrc` files, to modify your `PATH` to look in the `bin` directory for scripts, and to play around a bit with the `alias` command and with using `PS1` to modify your Bash shell prompt.

Important notes

- Remember: On the public course web site, at the end of the **References** section, there is a handout about how to use **ssh** to connect to `nrs-projects.humboldt.edu` and how to use **sftp** to transfer files to and from `nrs-projects.humboldt.edu`.
- This is a RARE individual lab, but PLEASE help one another! 8-)

Lab Exercise setup

- use `ssh` to connect to your account on `nrs-projects.humboldt.edu`

Part 1: create a `bin` directory

The traditional place for a Linux/UNIX user to put shell scripts and other programs they would like to use anywhere in their account is in a directory named `bin`, in their home directory.

Create this!

```
cd # start from your nrs-projects home directory
mkdir bin
chmod 700 bin
```

Part 2: create an initial `.bash_profile` file

(...if you do not already have one! 8-)

In your home directory, use `nano` to create an initial `.bash_profile` file:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]
then
```

```
    source ~/.bashrc
fi
```

```
# User specific environment and startup programs
export PATH=$PATH:$HOME/bin
```

If you would like, you can also add your current working directory to the end of your PATH:

```
export PATH=$PATH:$HOME/bin:..
```

Part 3: create an initial `.bashrc` file

(...if you do not already have one! 8-)

In your home directory, use nano to create an initial `.bashrc` file:

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]
then
    source /etc/bashrc
fi

# User specific aliases and functions
```

Part 4: practice with `alias`

Use the `alias` command to add at least one shell alias of your choice to the end of your `.bashrc` file, after the comment:

```
# User specific aliases and functions
```

Recall the basic syntax of the `alias` command:

```
alias desired_alias="cmd it is an alias for"
```

Now run:

```
source .bash_profile
source .bashrc
```

...and try out your alias; does it work?

Create a file `279alias-play.txt`, containing:

- your name
- paste in the `alias` command(s) you added to your `.bashrc`
- After each, PASTE an example using your alias, and also paste in its output (if it has any).

Part 5: experiment with `PS1`

Remember that the `PS1` environment variable lets you customize your bash prompt.

Run:

```
echo $PS1
```

...to see your current bash prompt.

You can put plain characters in your prompt as you wish, including spaces. But there are also a whole set of special characters available as well.

Play around with setting the PS1 environment variable, and see what you get!

Create a file `279prompt-play.txt`, containing:

- your name
- paste in at least three statements that set PS1 to 3 example prompts that you either liked or found particularly strange/interesting/etc.
- After each, PASTE an example of your resulting prompt after running that statement

Some prompt special characters

- `\h` is "The hostname, up to the first `.`. "
- `\W` is "The basename of \$PWD, with \$HOME abbreviated with a tilde. "
- `\u` is "The username of the current user. "
- `\$` can be used to escape special character \$ so the prompt ends with a literal dollar sign.
- `\a` - "a bell character" ...more like a THUMP on my Mac -- maybe demo?
- `\d` - "The date, in "Weekday Month Date" format (e.g., "Tue May 26"). "
- `\j` - "The number of jobs currently managed by the shell. "
- `\l` - "The basename of the shell's terminal device name. "
- `\n` - "A newline. "
- `\r` - "A carriage return. "
- `\s` - "The name of the shell, the basename of \$0 (the portion following the final slash)."
- `\t` - "The time, in 24-hour HH:MM:SS format. "
- `\T` - "The time, in 12-hour HH:MM:SS format. "
- `\@` - "The time, in 12-hour am/pm format. "
- `\A` - "The time, in 24-hour HH:MM format. "
- `\v` - "The version of Bash (e.g., 2.00) "
- `\V` - "The release of Bash, version + patchlevel (e.g., 2.00.0) "
- `\w` - "The current working directory, with \$HOME abbreviated with a tilde (uses the \$PROMPT_DIRTRIM variable). "
- `\W` - "The basename of \$PWD, with \$HOME abbreviated with a tilde. "

- \! - "The history number of this command. "
- \# - "The command number of this command. "
- \nnn - "The character whose ASCII code is the octal value nnn. "
- \\ - "A backslash."

Part 6: try out your bin directory

Choose one of your bash scripts (or write a new one, your choice) that prints something to standard output, and copy it into your new bin directory.

Now go into a completely different directory, and try to run the script you copied to your bash directory.

Run the following commands, from that other directory, to hopefully show me you have done this:

```
echo "bin contents:" >> 279bin-play.txt
```

```
ls -l ~/bin >> 279bin-play.txt
```

```
echo "" >> 279bin-play.txt
```

```
echo "running script your-selected-script from: " >> 279bin-play.txt
```

```
pwd >> 279bin-play.txt
```

```
echo "containing:" >> 279bin-play.txt
```

```
ls >> 279bin-play.txt
```

```
echo "" >> 279bin-play.txt
```

```
your-selected-script >> 279bin-play.txt
```

Submit these files to Canvas:

- .bash_profile
- .bashrc
- 279alias-play.txt
- 279prompt-play.txt
- 279bin-play.txt

Once you have submitted these lab exercise files, you may leave lab if you wish. Or, you can ask questions, read the course text, etc. But note that questions about today's lab exercise will get first priority.