

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Homework Assignment No. 01:

Cloud Computing

submitted to:

Professor Joseph Picone
ECE 1111: Engineering Computation I
Temple University
College of Engineering
1947 North 12th Street
Philadelphia, Pennsylvania 19122

August 26, 2025

prepared by:

Damien Ortiz
Email: tut62308@temple.edu

ECE 1111: Engineering Computation I

Homework No. 1: Cloud Computing

Goal: Demonstrate that you can log into Amazon AWS, traverse the file system, execute simple Unix commands, and manipulate your environment. An important part of this assignment is to understand the difference between the Unix bashrc and bash_profile files. Also, you will learn how to cut and paste text into a textbox, and use textboxes in MS Word.

Description: The tasks are:

1. Log into your ece-000 account. List the files in your home directory using “ls -l” or the alias “d”. Capture the screen output into a Word document by cutting and pasting the text from your terminal window – not including a screenshot (there are reasons for this). Your Word document should not include screenshots as images, but instead include textboxes that contain the text.
2. Demonstrate the use of command completion for both Linux commands and for filenames that match what you have typed from the command line. For example, type “nedc_” and hit tab. Explain the output that you see. Then change directories to your home directory using “cd \$HOME”. Follow this by typing “ls -l .ssh/” and hit tab. Explain what you observe.

Again, submit a Word document where what you see is inserted as text in a textbox, not a screenshot.

3. Explain the function of your .bash_profile and .bashrc. Use the more command to view these. Include a copy of these in your report. Show the significant pieces of these files in a textbox.
4. Using the text editor emacs, run using our local version, isip_e, modify your .bash_profile to set an environment variable named “ECE_1111” to the current date and time:

ECE_1111_profile=`date`;

Explain what this command does and what is the function of the backquote characters (“”).

In your .bashrc, place this command:

ECE_1111_bashrc=`date`;

From the command line, execute these commands:

printenv | grep ECE_1111

sleep 5

bash

printenv | grep ECE_1111

Cut and paste the output into a textbox. In your Word document, explain the output that you see and what the values of these environment variables differ.

Similarly, modify your .bashrc so that every time you create a new shell, the environment variable is updated to show the current date and time using this useful command:

5. Demonstrate the use of the "ls -l" command by changing your directory to your local ece_1111 directory (\$HOME/ece_1111) and displaying the contents of your .ssh directory (\$HOME/.ssh). Explain what information is shown in the output. Be sure to address the file permissions of each file.

Submit these five results in a single pdf document located here:

ece-000:/data/courses/ece_1111/current/homework/hw_01/<lastname_firstname>/hw_01.pdf

Use the MS Word template located here for this assignment:

https://www.isip.piconepress.com/courses/temple/ece_1111/resources/templates/lastname_firstname_hwxx.docx

Substitute “hw01” for “hwxx” in the filename. **Make sure you compress your pdf files using Adobe Acrobat’s “reduce file size” feature.**

Follow these instructions carefully. An important part of this course is learning how to conform to requirements. Programming involves a process of setting requirements and then implementing code that meets those requirements. Requirements gathering is an important part of the programming, or software engineering, process.

A. BRIEF DESCRIPTION OF YOUR CODE

```
total 0  
drwxr-xr-x. 2 tut62308 ece_1111  
130 May 13 22:44 login
```

When we use the ls -l command we receive the long form of content within our current directory.

When we hit tab after typing nedc_ we are shown every function that begins with nedc_ within the server.

```
ece-000_[5]: nedc_  
  
nedc_awstats_convert      nedc_edf_pyprint_duration  
nedc_cardio_check_ids     nedc_edf_pyprint_header  
nedc_cardio_find_ann      nedc_edf_pyprint_signal  
nedc_cardio_make_edf      nedc_edf_pyresample  
nedc_cardio_pymake_csv    nedc_edf_pystream  
nedc_cardio_pymake_edf    nedc_edf_split_signal  
nedc_disk_usage           nedc_edf_stream  
nedc_docs_compare          nedc_eeg_check_annotation_times  
nedc_dpath_add_image_dimensions nedc_eeg_convert_ann  
nedc_dpath_check_ann       nedc_eeg_convert_events_to_terms  
nedc_dpath_convert_ann     nedc_eeg_eval  
nedc_dpath_deidentify      nedc_eeg_gen_feats  
--More--
```

When we enter ls -l .ssh we see the long form of each file within the .ssh directory. We can see the permissions of the file, the parent directory, and when the file was last changed.

```
ece-000_[5]: ls -l .ssh
total 16
-rw----- 1 tut62308 ece_1111 673
Aug 25 17:59 authorized_keys

-rw----- 1 tut62308 ece_1111 2602
Aug 25 17:30 id_rsa

-rw-r--r-- 1 tut62308 ece_1111 570
Aug 25 17:30 id_rsa.pub

-rw-r--r-- 1 tut62308 ece_1111 198
Aug 25 17:58 known_hosts
```

The .bashrc script handles all customizations through bash after logging in and when creating new shells. For example, the ./data/isip/tools/GET_ENV.sh; line within my .bashrc imports all the custom functions from the server. Without it, I could not use things like our local version of emacs isip_e. My .bash_profile script runs .bashrc when I first log in to the server otherwise .bashrc would only run when I create a new shell. Both scripts can also be used to create environmental variables and functions I would like to use while logged in to the server. Within these scripts we used the backquotes around date to call the date function built into bash.

.bashrc

```
export ECE_1111_bashrc=`date`;
# define the location of isip tools
#
./data/isip/tools/GET_ENV.sh;
ece1111(){
    cd
    ../../data/courses/ece_1111/current
}
```

.bash_profile

```
export ECE_1111_profile=`date`;
.
. $HOME/.bashrc
```

When we use the printenv command and grep it for ECE_1111 we see that the environmental variables we created to store time in .bashrc and .bash_profile are the same. However, if we use the bash command to open up another shell we see that the time variable for .bashrc continues to change while the time variable for .bash_profile does not change. This is because .bash_profile only changes with the first bash shell you open. Any bash shells beyond this will work off of the .bashrc script without running the .bash_profile script unless told to do so.

```
ece-000_[5]: printenv | grep ECE_1111
rep ECE_1111
ECE_1111_profile=Tue Aug 26
07:54:11 EDT 2025
ECE_1111_bashrc=Tue Aug 26
07:54:11 EDT 2025
ece-000_[5]: sleep 5
ece-000_[5]: bash
ece-000_[6]: printenv | grep ECE_1111
ECE_1111_profile=Tue Aug 26
07:54:11 EDT 2025
ECE_1111_bashrc=Tue Aug 26
08:00:49 EDT 2025
```

```
ece-000_[5]: ls -l .ssh
total 16
-rw----- 1 tut62308 ece_1111 673
Aug 25 17:59 authorized_keys
-rw----- 1 tut62308 ece_1111 2602
Aug 25 17:30 id_rsa
-rw-r--r-- 1 tut62308 ece_1111 570
Aug 25 17:30 id_rsa.pub
-rw-r--r-- 1 tut62308 ece_1111 198
Aug 25 17:58 known_hosts
```

When using the ls -l command on .ssh we see the long form of everything within the .ssh directory. For both authorized_keys, and id_rsa only I can view and edit the files. The authorized_keys file contains all public keys that can be used to login under my account. The id_rsa file contains my private ssh key. The id_rsa.pub and known_hosts files can be viewed or edited by me and viewed by anyone else. The id_rsa.pub contains my public ssh key. The known_hosts file contains a list of known servers I've trusted.

B. SUMMARY

In this assignment I learned how to create environmental variables inside bash scripts, and I learned how to alter bash scripts to customize my Linux machine. I practiced using the ls -l command to find the long form of files, and I also practiced using the printenv command to see all environmental variables.

C. APPENDIX

```
total 0
```

```
drwxr-xr-x. 2 tut62308 ece_1111  
130 May 13 22:44 login
```

```
ece-000_[5]: nedc_
```

nedc_awstats_convert	nedc_edf_pyprint_duration
nedc_cardio_check_ids	nedc_edf_pyprint_header
nedc_cardio_find_ann	nedc_edf_pyprint_signal
nedc_cardio_make_edf	nedc_edf_pyresample
nedc_cardio_pymake_csv	nedc_edf_pystream
nedc_cardio_pymake_edf	nedc_edf_split_signal
nedc_disk_usage	nedc_edf_stream
nedc_docs_compare	nedc_eeg_check_annotation_times
nedc_dpath_add_image_dimensions	nedc_eeg_convert_ann
nedc_dpath_check_ann	nedc_eeg_convert_events_to_terms
nedc_dpath_convert_ann	nedc_eeg_eval
nedc_dpath_deidentify	nedc_eeg_gen_feats

--More--

```
ece-000_[5]: ls -l .ssh
```

```
total 16
-rw----- 1 tut62308 ece_1111 673
Aug 25 17:59 authorized_keys

-rw----- 1 tut62308 ece_1111 2602
Aug 25 17:30 id_rsa

-rw-r--r-- 1 tut62308 ece_1111 570
Aug 25 17:30 id_rsa.pub

-rw-r--r-- 1 tut62308 ece_1111 198
Aug 25 17:58 known_hosts
```

.bashrc

```
export ECE_1111_bashrc='date';

# define the location of isip tools
#
./data/isip/tools/GET_ENV.sh;

ece1111(){
    cd
    ../../data/courses/ece_1111/current
}
```

.bash_profile

```
export ECE_1111_profile='date';

. $HOME/.bashrc
```

```
ece-000_[5]: printenv | grep
ECE_1111
```

```
rep ECE_1111
ECE_1111_profile=Tue Aug 26
07:54:11 EDT 2025
```

```
ECE_1111_bashrc=Tue Aug 26
07:54:11 EDT 2025
```

```
ece-000_[5]: sleep 5
```

```
ece-000_[5]: bash
```

```
ece-000_[6]: printenv | grep
ECE_1111
```

```
ECE_1111_profile=Tue Aug 26
07:54:11 EDT 2025
```

```
ECE_1111_bashrc=Tue Aug 26
08:00:49 EDT 2025
```

```
ece-000_[5]: ls -l .ssh
```

```
total 16
```

```
-rw----- 1 tut62308 ece_1111 673
Aug 25 17:59 authorized_keys
```

```
-rw----- 1 tut62308 ece_1111 2602
Aug 25 17:30 id_rsa
```

```
-rw-r--r-- 1 tut62308 ece_1111 570
Aug 25 17:30 id_rsa.pub
```

```
-rw-r--r-- 1 tut62308 ece_1111 198
Aug 25 17:58 known_hosts
```