

Module 2 Checkpoint Exercise

Name: YUAN Wenxuan
University ID: 3036292740

Instructions:

For each single question or each group of questions in the Checkpoint exercise, please type your answer right after the question in this Word document. Please refer to the example below.

Checkpoint 0:

What is the meaning of the command “date”?

Ans: The “date” command prints the current date of the current machine

Checkpoint 2.1 (Please answer in this document and submit to Moodle)

What is the output of the following shell script? Please explain your answer.

```
#!/bin/bash

a=100
b=99
if [ $a \> $b ]
then
    echo "$a is larger"
else
    echo "$b is larger"
fi

if [ $a -gt $b ]
then
    echo "$a is larger"
else
    echo "$b is larger"
fi
```

Note: If you are using your own Linux, you may need to grant execute permission to run it.

Ans: The output is “99 is larger” in the first line and “100 is larger” in the second line. The first part compares “100” and “99” in the form of a string, and “100” is sorted before “99”, thus “99 is larger”. The second part compares 100 and 99 in the form of numbers.

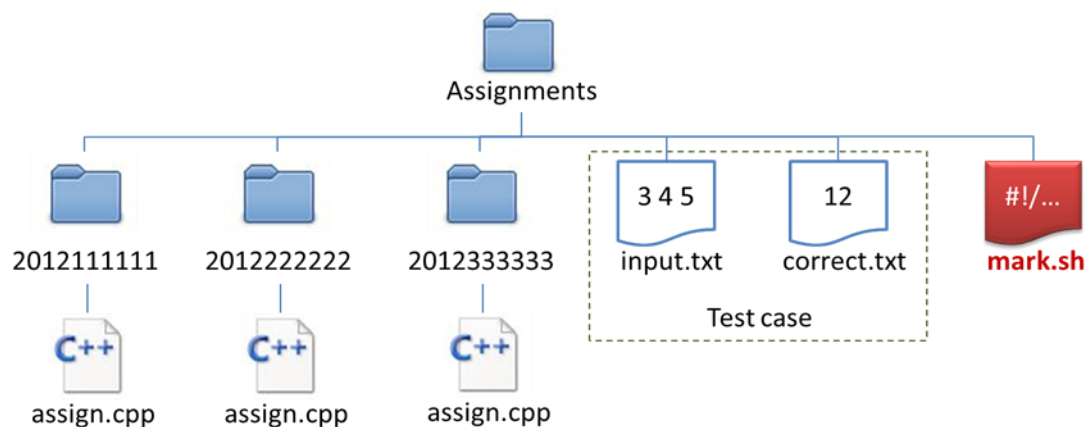
Checkpoint 2.2: Shell Script to mark all assignments (Please submit and evaluate your answer in VPL)

Let's upload the module files to the Linux server and browse to ~/Module2/Assignments/

```
$ cd ~/Module2/Assignments/  
$ ls  
2012111111  2012222222  2012333333  input.txt  correct.txt  mark.sh
```

Let's write a script to mark all the assignments in a class!

1. Suppose that the submission of each student is organized in the directory `Assignments/UID`, where UID is the University number of the student.
2. In this example we have three students with UID 2012111111, 2012222222 and 2012333333.
3. The test cases `input.txt` and `correct.txt` are located in the `Assignments` directory.
4. The shell script `mark.sh` is under the `Assignments` directory, to be updated by you.



Consider the unfinished shell script *mark.sh*

```
#!/bin/bash
allStudents=Q1
for student in $allStudents
do
    if Q2
    then
        cd $student
        if Q3
        then
            g++ assign.cpp -o assign.o 2>error.txt
            if Q4
            then
                ./assign.o < ../input.txt > output.txt
                result=`diff output.txt ../correct.txt`
                if Q5
                then
                    echo "$student: Wrong answer."
                else
                    echo "$student: Correct!"
                fi
            else
                echo "$student: Cannot be compiled."
            fi
        fi
        cd ..
    fi
done
```

Question1:

- What should be the code in **Q1**? (Reminder, spacing is critical!)
- Hints : First we need to get the names of the files and directories under our current directory, what is the command to do this? How we can execute shell command in shell script and store the result in the variable `allStudents`?

Question 2:

- However, the command in **Q1** also returns files under the Assignments directory, such as the test case files *input.txt* and *correct.txt*, as well as the shell script *mark.sh*.
- We need an if statement, what should be the condition in **Q2**?
- Hints: How can we check if `$student` is a directory?

Question 3:

- After we have identified the student's directories and browse into them one by one using the `cd $student` command, we need to check if the student has submitted *assign.cpp* or not.
- What is the condition to be filled in **Q3**?

Question 4:

- If the student has submitted *assign.cpp*, we then need to compile the *assign.cpp*, and report on screen if it cannot be compiled.
- We need to check if *assign.cpp* can be compiled or not.
- What is the condition to be filled in **Q4**?
- Hints: if the compilation is successful, what file will be created after executing the line `g++ assign.cpp -o assign.o 2> error.txt`?

Question 5:

- If the student has submitted *assign.cpp*, and *assign.cpp* can be compiled, we then need to check if the output file *output.txt* is the same as the test case *correct.txt* or not.
- The `result=`diff output.txt ../correct.txt`` command will store some value in the variable `$result` if *output.txt* is different from *correct.txt*.
- Therefore, if the length of the string `$result` is NOT EQUAL to 0, then there is something output by the `diff` command, that means *assign.o* returns wrong answer.
- What is the condition to be filled in **Q5**?

Upload your answer to the Moodle page “**Checkpoint 2.2: Shell Script to mark all assignments**” under Module 2 and **evaluate** it.

Please refer to “**Quick VPL Guide**” on Moodle if you are new to VPL.

Checkpoint 2.3: Shell Script to analyze web log (Please submit and evaluate your answer in VPL)

Suppose the web server of your company website that produces a log file which captures every page access in the following format:

ip_address date_YYYY/mm/dd time_HH:MM:SS page_accessed

Here is a sample content of such file:

192.168.83.173	2014-09-04	10:05:17	/products.html
192.168.130.175	2014-09-04	10:09:13	/products/296.html
192.168.130.175	2014-09-04	10:25:01	/index.html
192.168.138.244	2014-09-04	10:29:37	/products.html
...			

You are asked to write a shell script, **getProductAccess.sh**, which captures the product pages under **/products/** accessed by each IP address. The script takes one command line argument, which is the filename of a log file. It will read the content of the log file specified, find and print the product pages each IP address accessed. The output should be sorted by IP address and then by the product page.

- Sorting is done **alphabetically, ascendingly** for both IP address and product page, i.e., 192.168.0.11 will be ordered before 192.168.0.2.
- Only pages starting with **/products/** should be printed.
- You can assume that the input file will strictly follow the above format. I.e.:
 - Each page access is recorded on a separated line.
 - Each value (*ip_address*, *date*, *time* and *page_accessed*) are separated by tabs.
 - The values will not consist of spaces or tabs.
- The script does not need to handle cases where the input file is empty, or if there is not enough argument specified.

Sample Run

Assume we have *access-2014-09-04.log* in the current directory, with the following content:

192.168.83.173	2014-09-04	10:05:17	/products.html
192.168.130.175	2014-09-04	10:09:13	/products/296.html
192.168.130.175	2014-09-04	10:25:01	/index.html
192.168.138.244	2014-09-04	10:29:37	/products.html
192.168.83.173	2014-09-04	11:00:00	/products/004.html
192.168.130.175	2014-09-04	11:01:20	/products/296.html
192.168.138.244	2014-09-04	11:22:49	/products.html
192.168.138.244	2014-09-04	11:23:00	/index.html
192.168.83.173	2014-09-04	12:19:55	/products/560.html
192.168.130.175	2014-09-04	12:24:24	/products/004.html

If we run the script like this:

```
$ ./getProductAccess.sh access-2014-09-04.log
```

The script will print:

```
192.168.130.175    /products/004.html
192.168.130.175    /products/296.html
192.168.83.173     /products/004.html
192.168.83.173     /products/560.html
```

Note that IP 192.168.130.175 accessed /products/296.html multiple times, but there should only be one corresponding line in the output.

Hints:

- You may need to use the commands “cut”, “grep”, “sort” and/or “uniq” in your script.

Upload your answer to the Moodle page “**Checkpoint 2.3: Shell Script to analyze web**” under Module 2 and **evaluate** it.

Checkpoint 2.4

GitHub has an education plan to allow university students to apply for a professional account for free for at least two years. All students can join at <https://education.github.com/>.

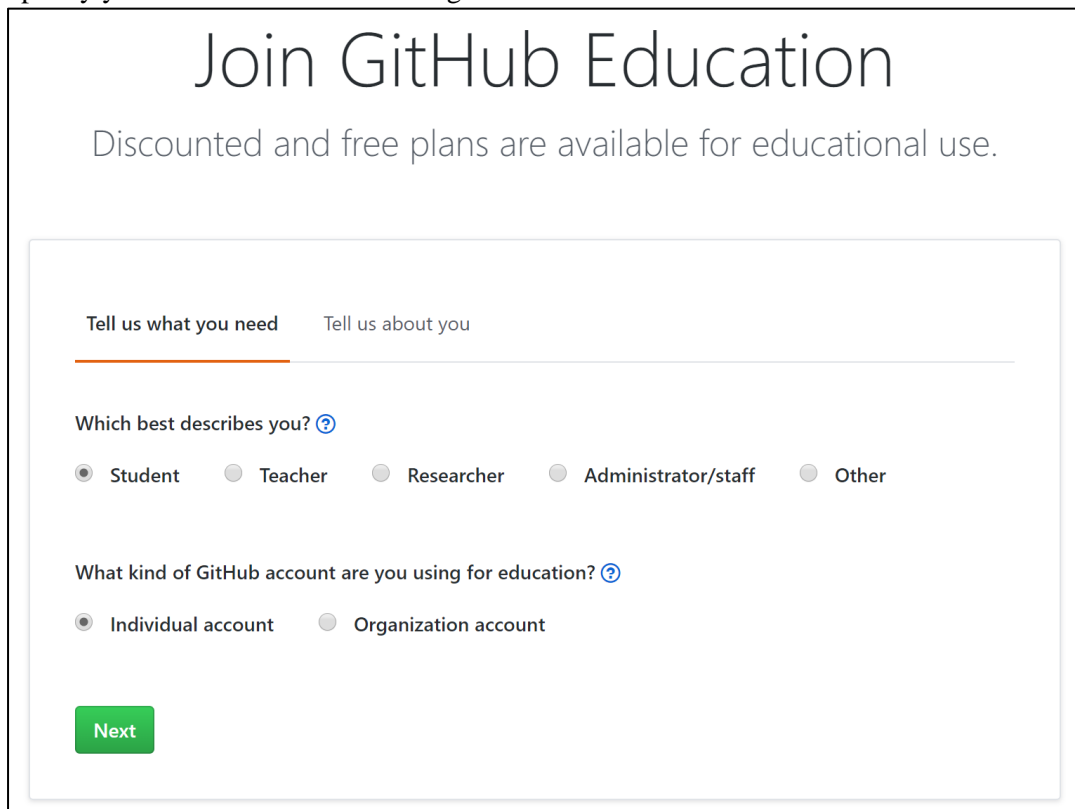
Later for student projects, we plan to request students to commit their code and changes to GitHub mandatorily.

To apply for the professional account, you must:

1. Prepare a digital copy of your Student ID card.
2. Create a free GitHub account at <https://github.com/>.
3. Join the “GitHub Education” at <https://education.github.com/>.

Here are some steps to join the GitHub Education:

1. Specify you are a **student** and creating an **individual account**. Then click “Next”.



The screenshot shows the 'Join GitHub Education' page. At the top, it says 'Join GitHub Education' and 'Discounted and free plans are available for educational use.' Below this is a form with two tabs: 'Tell us what you need' (active) and 'Tell us about you'. The first question is 'Which best describes you?' with five radio button options: Student (selected), Teacher, Researcher, Administrator/staff, and Other. The second question is 'What kind of GitHub account are you using for education?' with two radio button options: Individual account (selected) and Organization account. At the bottom left of the form is a green 'Next' button.

2. Add your HKU email address and then click “add”. A verification email will be sent to your HKU mailbox. Click the link in the email to verify it. After that, refresh the current page and upload the photo of your student ID card to the space provided. Then click “Submit request”

Tell us what you need

Tell us about you

Name

What e-mail address do you use for school?
⚡ **Pro-tip:** Selecting a school-issued email address gives you the best chance of a speedy review.

☐

How do you plan to use GitHub?

Please note, your request cannot be edited once it has been submitted, so please verify your details for accuracy before sending them to us.

3. You will be notified your application result within 5 days

Back to GitHub.com

GitHub Support

Contact GitHub

GitHub Education

StudentsTeachersSchoolsEvents

Thanks for submitting!

Be sure to check your email. If you don't hear from us within the hour, you should receive an email from us in fewer than **5 days**. Have an Octotastic day!

GitHub
© 2019

Features
Code review
Project management
Integrations
Community
Documentation
Code hosting

Platform
Atom
Electron
GitHub Desktop
Developers

Community
Personal
Open source
For Business
For Education
Sponsorships

Company
About
Blog
Customers
Careers
Press
Shop

Resources
Contact
Help
Status
Terms
Privacy
Security
Training

Note: There is no need to submit anything for this checkpoint.