

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG ĐIỆN – ĐIỆN TỬ**

---o0o---



**SEEE**

**BÁO CÁO BÀI TẬP LỚN**  
**HỆ ĐIỀU HÀNH**

**ĐỀ TÀI:**

**SHELL SIMULATOR USING C/C++**

Giảng viên hướng dẫn: ***PGS.TS Đào Trung Kiên***

Mã lớp: ***145601***

Nhóm sinh viên thực hiện: ***Nhóm 5***

***Nguyễn Tuấn Anh*** ***20200038***

***Hà Đình Đạt*** ***20203355***

***Nguyễn Phạm Trung Hiếu*** ***20203706***

***Nghiêm Văn Quang*** ***20203547***

***Hà Nội, tháng 1 năm 2024***

# BÁO CÁO BÀI TẬP LỚN – MÔN: HỆ ĐIỀU HÀNH SHELL SIMULATOR USING C/C++

---

Giảng viên hướng dẫn:	<b>PGS. TS Đào Trung Kiên</b>
-----------------------	-------------------------------

Mã lớp:	<b>145601</b>
---------	---------------

Nhóm sinh viên thực hiện:	<b>Nhóm 5</b>
---------------------------	---------------

<i>Nguyễn Tuấn Anh</i>	<i>20200038</i>
------------------------	-----------------

<i>Hà Đình Đạt</i>	<i>20203355</i>
--------------------	-----------------

<i>Nguyễn Phạm Trung Hiếu</i>	<i>20203706</i>
-------------------------------	-----------------

<i>Nghiêm Văn Quang</i>	<i>20203547</i>
-------------------------	-----------------

---

## MỤC LỤC

<b>1. Phân công công việc .....</b>	<b>1</b>
<b>2. Yêu cầu ứng dụng.....</b>	<b>1</b>
<b>3. Thực hiện mô phỏng.....</b>	<b>1</b>
<b>4. Thử nghiệm mô phỏng .....</b>	<b>2</b>

**Github**—[https://github.com/highquanglity/OS\\_course/tree/main/Project](https://github.com/highquanglity/OS_course/tree/main/Project).

**Giới thiệu**—Trong nội dung bài tập lớn này, các chức năng và câu lệnh của một Linux shell cơ bản sẽ được nhóm mô phỏng sử dụng ngôn ngữ lập trình C++.

## 1. PHÂN CÔNG CÔNG VIỆC

MSSV	Thành viên	Công việc
20200038	Nguyễn Tuấn Anh	Tìm hiểu về external command, exeExternalCmd(), handle_cd(), find_special_character(), expand_wcard(), expandWildcardTokens(), find_and_erase(), IsInsideQuotes(), làm báo cáo.
20203355	Hà Đình Đạt	Tìm hiểu về internal command, get_current_dir(), display_promt, handle_pwd(), handle_help(), trim(), wcard_check(), handle_redirect_check(), InItalicCmd_check().
20203706	Nguyễn Phạm Trung Hiếu	Viết test case, handle_cat(), exeInternalCmd(), SplitCmd_char(), get_delimiter_positions(), exeCmdOutput(), SplitCmd_pos(), README.md, làm báo cáo.
20203547	Nghiêm Văn Quang	Chia module, viết test case, doc cho hàm, ghép chương trình, single_cmd_process(), handle_redirection(), exeBackgndCmd(), exePipeCmd(), exeComplexCmds(), GetItalicCommaCommandOutput(), README.md.

## 2. YÊU CẦU ỨNG DỤNG

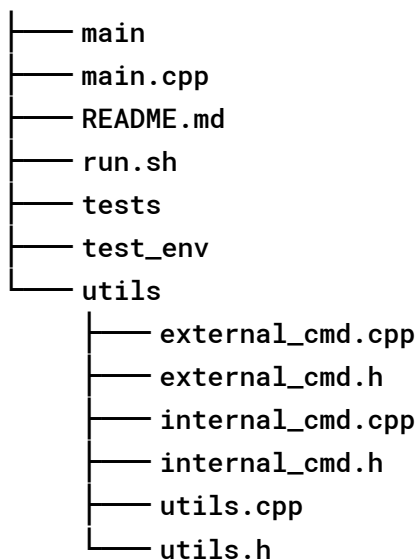
- C++ 11, C++ 14, C++ 17
- g++/gcc
- Ubuntu > 16.04

## 3. THỰC HIỆN MÔ PHỎNG

Đầu tiên, git clone toàn bộ Project về máy, dùng câu lệnh:

```
git clone https://github.com/highquanglity/OS\_course.git  
cd Project
```

Dưới đây là cấu trúc thư mục **Project**:



**run.sh** là bash script để chạy biên dịch toàn chương trình, sau đó chạy **./main** để vào chương trình. Các file phụ trợ header file và source file nằm trong thư mục **utils**. Folder **tests** chứa các chương trình C++ để kiểm tra chức năng từng hàm. Folder **test\_env** là thư mục riêng biệt chứa các file và folder được tạo ra để test chương trình hoạt động như thế nào, không gây ảnh hưởng đến các folder khác.

Khi đã trong folder **Project**, thực hiện bash script sau để biên dịch và chạy chương trình cho lần đầu hoặc có chỉnh sửa source:

---

```
sh run.sh
```

---

Từ lần mô phỏng tiếp theo, không cần biên dịch lại nữa, chỉ cần chạy chương trình:

---

```
./main
```

---

Sau đó, thao tác với chương trình trong giao diện dòng lệnh như thao tác với shell bình thường, *chú ý hiện tại chưa thể sử dụng các arrow keys trong terminal*.

#### 4. THỬ NGHIỆM MÔ PHỎNG

Dưới đây là danh sách các câu lệnh được dùng làm test case trong quá trình thực hiện thử nghiệm chương trình mô phỏng:

Test case	Đầu ra trên Ubuntu Terminal	Kết quả chương trình
<code>cd test_env</code>	Chuyển thành công vào <b>test_env</b>	Chuyển thành công vào <b>test_env</b>

<code>pwd</code>	/home/quang/Desktop/OS_course /Project/test_env	/home/quang/Desktop/OS_course /Project/test_env
<code>echo date</code>	date	date
<code>cat hehe.py</code>	print("dung roi ne")	print("dung roi ne")
<code>ls *.ipynb</code>	bed_11.ipynb bed_21.ipynb mnist_1_core.ipynb mnist_5_core.ipynb mobile_ss.ipynb	bed_11.ipynb bed_21.ipynb mnist_1_core.ipynb mnist_5_core.ipynb mobile_ss.ipynb
<code>ls bed*.ipynb</code>	bed_11.ipynb bed_21.ipynb	bed_11.ipynb bed_21.ipynb
<code>mkdir "quang'hieu"</code>	Tạo thành công thư mục có tên <b>quang'hieu</b>	Tạo thành công thư mục có tên <b>quang'hieu</b>
<code>mkdir "quang'dat'"</code>	Tạo thành công thư mục có tên <b>quang'dat'</b>	Tạo thành công thư mục có tên <b>quang'dat'</b>
<code>mkdir 'quang"tanh'</code>	Tạo thành công thư mục có tên <b>quang"tanh"</b>	Tạo thành công thư mục có tên <b>quang"tanh"</b>
<code>mkdir "quang'linh'"</code>	Tạo thành công thư mục có tên <b>quang'linh''</b>	Tạo thành công thư mục có tên <b>quang'linh''</b>
<code>mkdir "quang"'"</code>	shell vào chế độ đợi lệnh kép tiếp theo	Invalid command, chương trình bị break
Vào trong thư mục yolo, thực hiện: <code>ls   head -3</code>	yolo3.yaml yolov4.yaml yolov5.yaml	yolo3.yaml yolov4.yaml yolov5.yaml
Vào trong thư mục yolo, thực hiện: <code>ls   head -3   tail -1</code>	yolov5.yaml	yolov5.yaml
<code>echo `date`</code>	Thứ bảy, 06 Tháng 1 năm 2024 18:47:44 +07	Thứ bảy, 06 Tháng 1 năm 2024 18:47:44 +07
<code>ls &gt;&gt; file.txt</code>	Ghi thành công output khi thực hiện câu lệnh <b>ls</b> vào <b>file.txt</b>	Ghi thành công output khi thực hiện câu lệnh <b>ls</b> vào <b>file.txt</b>

<code>pwd &gt; file.txt</code>	Xóa toàn bộ nội dung <b>file.txt</b> đã có và ghi vào file này output của câu lệnh <b>pwd</b>	Xóa toàn bộ nội dung <b>file.txt</b> đã có và ghi vào file này output của câu lệnh <b>pwd</b>
<code>mkdir "hello&amp;&amp;world" &amp;&amp; pwd</code>	Tạo thành công thư mục có tên <b>"hello&amp;&amp;world"</b> đồng thời thực hiện câu lệnh <b>pwd</b> và in ra terminal	Tạo thành công thư mục có tên <b>"hello&amp;&amp;world"</b> đồng thời thực hiện câu lệnh <b>pwd</b> và in ra terminal
<code>ls &amp; pwd &amp; echo hi</code>	Chạy ngầm và in ra <b>pid</b> của <b>ls</b> , <b>pwd</b> , <b>echo hi</b> chạy xong đầu tiên, tiếp theo là <b>pwd</b> và cuối cùng là <b>ls</b>	Chạy ngầm và in ra <b>pid</b> của <b>ls</b> , <b>pwd</b> , <b>echo hi</b> chạy xong đầu tiên, tiếp theo là <b>pwd</b> và cuối cùng là <b>ls</b>
<code>cat hehe.txt &gt; o.txt</code>	Ghi thành công nội dung của file <b>hehe.txt</b> vào file <b>o.txt</b>	Ghi thành công nội dung của file <b>hehe.txt</b> vào file <b>o.txt</b>