|  |
| --- |
|  |

|  |
| --- |
| **Operating System** |
|  |
|  |
| PROJECT 1: SIMPLE SHELL   |  |  | | --- | --- | | Tên thành viên | MSSV | | Nguyễn Hoàng Việt | 1712905 | | Trần Thúy Tuyền | 1712891 | | Phạm Thị Tuyết Vy | 1712927 |  I.Tiến trình xử lý: ***1. Nhập lệnh command.***  Kiểm tra tính hợp lệ,  chuẩn hóa sau (nếu được).  Còn không, báo lỗi và lặp lại bước này.  ***2. Kiểm tra và xử lý các trường hợp:***  Dấu “&” nằm cuối cùng command.  User nhập “exit” để thoát chương trình.  Command có ký tự “<” hoặc “>”.  Command có ký tự “|”  ***3. Tính toán và xử lý liên quan đến History Feature:***  User nhập “history” để có danh sách 10 command sử dụng gần nhất.  Chạy hàm history\_computation:   * Xử lý command “!!” * Xử lý command “![%d]”   với [%d] là một con số.   * Lưu command vào lịch sử.   ***4. Xử lý pipe.***  Chạy hàm pipe (ở bước 2)  ***5. Xử lý input/output.***  Chạy hàm input hoặc output. (ở bước 2)  ***6. Thực thi command bằng fork()(tạo tiến trình con) và gọi execvp() trong child proc.***  ***7. Quay lại bước 1.***  Đặt tiến trình xử lý vào vòng lặp để user nhập lệnh command tiếp theo.  II.Thiết kế chương trình:  ***1.Phần 1:*** *Khai báo các thư viện,biến và hàm cần dùng.* ***2. Phần 2:*** *Hàm main() chứa các hàm theo yêu cầu của project, gồm các phần sau:* (1) Command line:  - Đọc được commands từ users nhập vào, lưu vào biến cmd\_line và đảm bảo tính hợp lệ của lệnh.  - Lấy được arguments (args – đối số) của một command line và tiến hành xử lý để thực thi được command.  - Sau khi nhập vào một dòng lệnh và nhấn enter để chương trình thực thi, chương trình trả kết quả về màn hình console và sẵn sàng cho users nhập dòng lệnh tiếp theo.  - Cuối cùng, chương trình kết thúc khi users nhập vào “exit”. (2) Redirecting output/input:  - Cài đặt hàm output, sử dụng dup2() để chuyển hướng con trỏ vào file <output> và in kết quả lên file sau khi command được thực thi.  - Cài đặt hàm input, bắt lỗi không tìm thấy file. (3) Communication via a Pipe:  - Khởi tạo biến pipeFlag=0, (4) History feature:  - Truy cập được danh sách 10 command sử dụng gần nhất của users (tính từ khi bắt đầu chương trình đến khi kết thúc chương trình): > “history”  - Gọi hàm xử lý history\_computation và trả về con trỏ argsPtr trỏ vào vùng nhớ lưu lệnh cần thực thi trước khi được bỏ vào hàm execvp().  - Một số xử lý khác: Thực hiện các hàm xét ở 3 trường hợp:  +Xét trong trường hợp user nhập vào "!!" hoặc "![%d]" mà lệnh đó là lệnh yêu cầu xử lý chức năng redirecting input + Xét trong trường hợp user nhập vào "!!" hoặc "![%d]" mà lệnh đó là lệnh yêu cầu xử lý chức năng redirecting output + Xét trong trường hợp user nhập vào "!!" hoặc "![%d]" mà lệnh đó là lệnh yêu cầu xử lý chức năng communication via a pipe (5) Fork child to Execute args:  - Kiểm tra pipeFlag đã khởi tạo ở phần Communication via a pipe.  - Xét (pid <0), (pid =0) và (pid >0) \*Check các lệnh để đưa vào History:  - Kiểm tra lần lượt các trường hợp lệnh mà users nhập vào.  - Xử lý việc users truy cập sai cách (Vượt quá giới hạn Index, nhập sai Index)  - Lệnh hợp lệ, lưu vào History \*Thực hiện các hàm giải phóng History, in ra History  \*Thực thi hàm pipe:  - Khởi tạo biến pid1 với các trường hợp ( pid1 <0 , pid1 =0 và pid1 >0) để xét rõ nơi thực thi các dòng lệnh code II. Tài liệu tham khảo Processes. Operating-System-Concepts của Silberschatz |
|  |
|  |