

ĐẠI HỌC BÁCH KHOA HÀ NỘI
Trường Công nghệ Thông tin và Truyền thông



TINY SHELL

Tương tác với hệ điều hành thông qua giao diện dòng lệnh

Nhóm tác giả:

STT	Họ và tên	MSSV
1	Lưu Thịnh Khang	20220031
2	Nguyễn Viết Tuấn Kiệt	20220032
3	Bùi Quang Phong	20224887
4	Nguyễn Thanh Tuyền	20224908

Giảng viên hướng dẫn:

TS. Phạm Đăng Hải

Học phần:

IT3070 - Nguyên lý hệ điều hành

Hà Nội - Ngày 13 tháng 6 năm 2024

Tiny Shell

Tương tác với hệ điều hành thông qua giao diện dòng lệnh

Học phần: IT3070 - Nguyên lý Hệ điều hành
Giảng viên hướng dẫn: TS. Phạm Đăng Hải

N. V. T. Kiệt^{1,2}, B. Q. Phong^{1,2}, L. T. Khang¹, N. T. Tuyền²¹

¹Chương trình tài năng - Khoa học máy tính K67
Trường Công nghệ Thông tin và Truyền thông

²Phòng thí nghiệm Mô hình hóa, Mô phỏng và Tối ưu hóa
Trung tâm nghiên cứu quốc tế về Trí tuệ nhân tạo, BKAI

Ngày 13 tháng 6 năm 2024

Notes

Mục lục

Giới thiệu chung

Tiny Shell

Kết luận & Thảo luận

Notes

Giới thiệu chung

Khái niệm
Windows API
Shell

Tiny Shell

Cài đặt và triển khai
Các tính năng cơ bản
Các tính năng nâng cao
Các tiện ích mở rộng

Kết luận & Thảo luận

Tính ứng dụng
Khó khăn và giải pháp

Notes

Giao diện dòng lệnh

- **Giao diện dòng lệnh** (Command Line Interface - CLI) là một phương thức tương tác giữa người dùng và hệ thống máy tính.
 - Người dùng nhập các lệnh dưới **dạng văn bản** và nhận phản hồi dưới dạng văn bản.
 - Yêu cầu người dùng có kiến thức về các lệnh và cú pháp.
- Chương trình xử lý giao diện dòng lệnh được gọi là Trình thông dịch dòng lệnh (command-line interpreter), Trình xử lý dòng lệnh (command-line processor), hay Shell.

Notes

Lời gọi hệ thống

- **Lời gọi hệ thống** (system call) là một giao diện mà chương trình ứng dụng (user application) sử dụng để yêu cầu các dịch vụ từ hệ điều hành.
- Lời gọi hệ thống thường được gọi thông qua các hàm thư viện trong ngôn ngữ lập trình.

Ví dụ:

- ▶ Trên Unix/Linux: `open()`, `wait()`, `mmap()`, `socket()`,...
- ▶ Trên Windows: `CreateFile()`, `WaitForSingleObject()`, `VirtualAlloc()`,...
- Trên Windows, các lời gọi hệ thống được cung cấp thông qua các API của Windows, chủ yếu từ Windows API (WinAPI). Các lời gọi hệ thống trên Windows thường phức tạp hơn và bao gồm nhiều chức năng bổ sung để tương thích với kiến trúc của Windows.

Notes

Windows API

- **Windows API** là một tập hợp các giao diện lập trình ứng dụng được Microsoft cung cấp, cho phép các phần mềm tương tác với hệ điều hành Windows.
- Chức năng: WinAPI cung cấp các chức năng cần thiết để quản lý tệp, tiến trình, bộ nhớ, giao tiếp mạng và nhiều hơn nữa.

Notes

Tính năng	Windows (WinAPI)	Unix/Linux
Quản lý tệp	CreateFile, ReadFile, WriteFile, CloseHandle	open, read, write, close
Quản lý tiến trình	CreateProcess, WaitForSingleObject, TerminateProcess	fork, exec, wait, kill
Quản lý bộ nhớ	VirtualAlloc, VirtualFree	mmap, munmap, brk
Giao tiếp mạng	Winsock: socket, bind, listen, accept, connect	BSD Sockets: socket, bind, listen, accept, connect
Tính mở rộng	Nhiều tùy chọn và tham số	Thường đơn giản, ít tùy chọn hơn
Tài liệu hỗ trợ	Phong phú, chi tiết	Tài liệu tốt nhưng không phong phú bằng Windows

Bảng 1: So sánh lời gọi hệ thống giữa Windows và Unix/Linux

Notes

- Để biên dịch các chương trình sử dụng Windows API, bạn cần đảm bảo rằng trình biên dịch của bạn được cấu hình đúng để liên kết với các thư viện cần thiết.
- Để sử dụng các hàm Windows API, bạn cần bao gồm tệp tiêu đề `<windows.h>` trong mã nguồn C++ của bạn. Đây là tệp tiêu đề chính chứa các khai báo cho hầu hết các hàm Windows API.

Notes

Shell trong Windows

Shell trong Windows là một giao diện cho phép người dùng tương tác với hệ điều hành và quản lý tài nguyên máy tính. Các loại Shell trong Windows:

- **Windows Shell (Graphical Shell):** giao diện đồ họa người dùng (GUI) chính.
- **Command Prompt (CMD):** giao diện dòng lệnh (CLI) truyền thống.
- **PowerShell:** công cụ dòng lệnh mạnh mẽ và một ngôn ngữ script được thiết kế cho quản trị hệ thống và tự động hóa trên Windows.

Notes

Tính năng	CMD	PowerShell
Ngôn ngữ	Dòng lệnh cơ bản	Ngôn ngữ script mạnh mẽ dựa trên .NET
Cmdlets	Không có	Hàng ngàn cmdlets mạnh mẽ
Pipeline	Hạn chế	Mạnh mẽ và linh hoạt
Scripting	Hỗ trợ các script .bat đơn giản	Hỗ trợ script mạnh mẽ với .ps1
Tích hợp .NET	Không có	Tích hợp sâu với .NET
Khả năng tự động hóa	Hạn chế	Mạnh mẽ và linh hoạt

Bảng 2: So sánh giữa CMD và PowerShell

Notes

Giới thiệu chung

Khái niệm
Windows API
Shell

Tiny Shell

Cài đặt và triển khai
Các tính năng cơ bản
Các tính năng nâng cao
Các tiện ích mở rộng

Kết luận & Thảo luận

Tính ứng dụng
Khó khăn và giải pháp

Notes

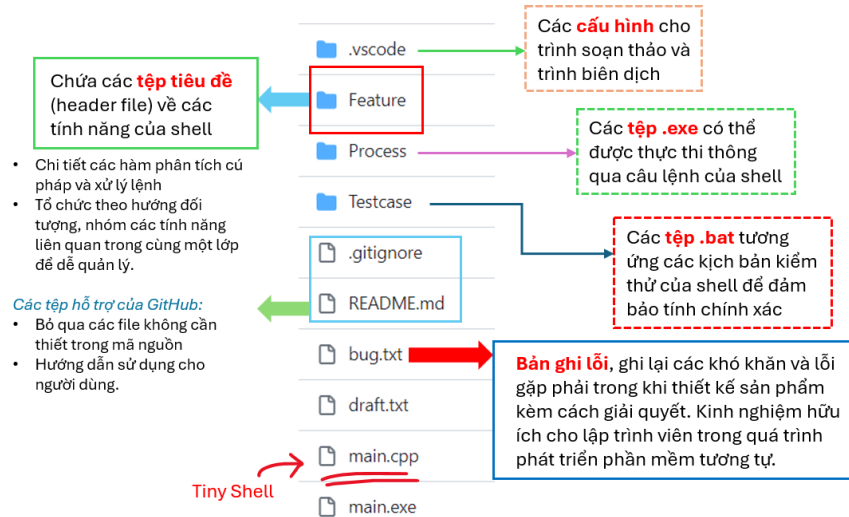
Môi trường phát triển

- **Ngôn ngữ lập trình:** C++ 17 hoặc lớn hơn
- **Trình biên dịch:** g++ (Rev3, Built by MSYS2 project) 13.2.0
- **Phần mềm:** IDE Visual Studio Code V1.90.0 hoặc phiên bản mới nhất
- **Hệ điều hành:** Windows 10 trở lên, tốt nhất nếu Windows 11

Notes

Cấu trúc mã nguồn

Tổ chức mã nguồn¹: Dễ đọc, dễ quản lý, dễ sử dụng kèm hướng dẫn chi tiết.



¹ Mã nguồn dự án: Operating System Project. Hướng dẫn sử dụng: README

Notes

1. Sau khi biên dịch tệp `main.cpp` và chạy chương trình `main.exe` thành công, bạn có thể chính thức sử dụng Tiny Shell.

```
=====
                Tiny Shell
=====
Welcome to Tiny Shell!
This is a simple shell program to interact with the Windows operating system.
PID of Tiny Shell: 25264
Type 'help' to see the list of available commands.
=====
tiny_shell>
```

Hình 1: Màn hình hiện lên sau khi khởi chạy Tiny Shell

Notes

2. Hãy bắt đầu với lệnh `help` để xem **danh sách các lệnh** mà Tiny Shell hỗ trợ.
→ Nếu không rõ cú pháp lệnh, hay gõ (sai) lệnh và Tiny Shell sẽ thông báo.

Ví dụ:

```
tiny_shell> move
Usage: move <source_path> <destination_path>
tiny_shell> function
Error: Usage: function <name>(<params>) = <expression>
tiny_shell> alias h
Usage: alias <name> = <command> or alias
```

Hình 2: Nếu nhập sai cú pháp lệnh, Tiny Shell sẽ nhắc nhở cú pháp đúng

Notes

3. Tiny Shell hỗ trợ **70 lệnh cơ bản** và tổng cộng **90 cú pháp lệnh**. Trong đó:
 - ▶ Quản lý tệp và thư mục: 15 lệnh.
 - ▶ Quản lý tiến trình: 11 lệnh.
 - ▶ Hệ thống và tiện ích: 17 lệnh.
 - ▶ Điều hướng và giao diện: 10 lệnh.
 - ▶ Tính toán nâng cao: 18 lệnh.
4. Lấy cảm hứng từ CMD và PowerShell của hệ điều hành Windows.
 - Tiny Shell còn có những **tính năng riêng đặc biệt**, cung cấp nhiều tiện ích để người dùng dễ dàng giao tiếp với máy tính.

Notes

Quản lý tệp

Tiny Shell có thể thực hiện một số thao tác với tệp:

Tính năng thiết yếu	Tính năng nâng cao
open	check_file
move_file	file_size
rename_file	read_file: hỗ trợ đọc theo dòng
delete_file	write_file: hỗ trợ ghi theo dòng
create_file	extension_file: xem phần mở rộng
copy_file	list_file: liệt kê file với phần mở rộng

Bảng 3: Một số câu lệnh đối với tệp

Notes

```
tiny_shell> create_file code.py
File created successfully: code.py
tiny_shell> create_file code.cpp
File created successfully: code.cpp
tiny_shell> check_file "Process/child.exe"
File exists: Process/child.exe
tiny_shell> list_file Process\
"child.cpp"
"child.exe"
"countdown.cpp"
"countdown.exe"
"duck.cpp"
"duck.exe"
"tictactoe.cpp"
"tictactoe.exe"
tiny_shell> list_file Process\ .exe
"child.exe"
"countdown.exe"
"duck.exe"
"tictactoe.exe"
```

Hình 3: Chạy các câu lệnh với kịch bản kiểm thử về file

Notes

Quản lý thư mục

Tiny Shell có thể thực hiện một số thao tác với thư mục:

- `create`
- `open`
- `delete`
- `copy`
- `rename`
- `move`
- `cd`
- `dir`
- `pwd`
- `list_tree`

Notes

```
tiny_shell> create "New Folder"
Directory created: "New Folder"
tiny_shell> copy "New Folder" "New Folder 2"
Directory copied from "New Folder" to "New Folder 2"
tiny_shell> rename "New Folder 2" "Temp Folder"
File renamed from "New Folder 2" to "Temp Folder"
tiny_shell> delete "Temp Folder"
Directory deleted: "Temp Folder"
tiny_shell> cd "New Folder"
Changed directory to "New Folder"
```

Hình 4: Chạy các câu lệnh với kịch bản kiểm thử về thư mục

Notes

Quản lý tiến trình

Tiny Shell chứa các câu lệnh quản lý tiến trình:

Lệnh	Mô tả lệnh
list_processes	In ra danh sách tiến trình
start_foreground	Tạo tiến trình con ở trạng thái foreground
start_background	Tạo tiến trình con ở trạng thái background
terminate	Chấm dứt một tiến trình
suspend	Tạm dừng một tiến trình
resume	Tiếp tục một tiến trình

Bảng 4: Một số câu lệnh đối với tiến trình

Notes

[illegible]

```
tiny_shell> list_processes
PID      Process Name      Status
-----
0        [System Process]  Access Denied
4        System            Access Denied
140      Secure System     Access Denied
184      Registry          Access Denied
688      smss.exe          Access Denied
1104     csrss.exe         Access Denied
1200     wininit.exe       Access Denied
1208     csrss.exe         Access Denied
1276     services.exe      Access Denied
1288     LsaIso.exe        Access Denied
1308     lsass.exe         Access Denied
```

Hình 5: Câu lệnh liệt kê tiến trình

Notes

Lệnh nâng cao	Mô tả lệnh
list_children	In ra danh sách tiến trình con
manage_threads	Quản lý luồng
child	Tạo một tiến trình con đơn giản (background)
countdown	Tạo một cửa sổ đếm ngược 10 giây (background)
duck	Tạo một chú vịt bơi trên màn hình (foreground)
tictactoe	Khởi động trò chơi Tic-Tac-Toe trên màn hình (foreground)
dancing	Hiển thị một khuôn mặt nhảy múa trên màn hình (foreground)
after	Lên lịch chạy câu lệnh sau một khoảng thời gian

Bảng 5: Một số câu lệnh nâng cao đối với tiến trình

Trong đó:

- Nếu đang có tiến trình con chạy foreground và người dùng ấn tổ hợp phím **Ctrl + C** thì tiến trình con sẽ dừng và trở về Shell chính.
- Nếu Shell đang chạy và người dùng ấn tổ hợp phím **Ctrl + C** thì không khiến Shell dừng, thay vào đó, nó bỏ qua dòng hiện tại tạo ra dòng mới `tiny_shell>`.

Notes

[illegible]


```
tiny_shell> tictactoe
  1 2 3
1 . . .
2 . . .
3 . . .
Player X, enter your move (row and column): 1 1
  1 2 3
1 X . .
2 . . .
3 . . .
Player O, enter your move (row and column): 2 2
  1 2 3
1 X . .
2 . O .
3 . . .
Player X, enter your move (row and column):
```

Hình 6: Chạy Tic-Tac-Toe (foreground), có thể ấn Ctrl + C để thoát và quay lại Tiny Shell

Notes

Quản lý biến môi trường

Tiny Shell chứa các câu lệnh quản lý đường dẫn và biến môi trường:

Lệnh	Mô tả lệnh
add_path	Thêm một đường dẫn vào PATH
is_in_path	Kiểm tra xem một đường dẫn có nằm trong PATH hay không
remove_path	Xóa một đường dẫn khỏi PATH
set_env	Thiết lập một biến môi trường
unset_env	Hủy bỏ một biến môi trường
print_env	In giá trị của một biến môi trường cụ thể
list_env	Liệt kê tất cả các biến môi trường
save_env	Lưu các biến môi trường vào một tệp
load_env	Tải các biến môi trường từ một tệp

Bảng 6: Một số câu lệnh đối với đường dẫn và biến môi trường

Notes

```
tiny_shell> add_path Testcase
Added Testcase to PATH.
tiny_shell> is_in_path Testcase
Testcase is in PATH.
tiny_shell> set_env a = 1
tiny_shell> set_env b = 2
tiny_shell> print_env a
a = 1
tiny_shell> set_env c = a+b
tiny_shell> print_env c
c = 3
tiny_shell> remove_path Testcase
Removed Testcase from PATH.
tiny_shell> unset_env a
Unset variable a
tiny_shell> unset_env b
Unset variable b
tiny_shell> unset_env c
Unset variable c
tiny_shell> print_env b
b is not set.
```

Hình 7: Chạy các câu lệnh với kịch bản kiểm thử về đường dẫn và biến môi trường

Notes

Tiện ích hệ thống

Tiny Shell có thể hiểu một số lệnh về tiện ích hệ thống:

Lệnh thiết yếu	Mô tả lệnh
time	Hiển thị thời gian hiện tại
date	Hiển thị ngày hiện tại
uptime	Hiển thị thời gian hệ thống đã hoạt động
cpuinfo	Hiển thị thông tin về CPU
meminfo	Hiển thị thông tin về bộ nhớ
diskinfo	Hiển thị thông tin về đĩa cứng
calculator	Mở máy tính

Bảng 7: Một số câu lệnh đối với tiện ích hệ thống

Notes

```
tiny_shell> time
Current system time: 2024-6-11 21:42:49
tiny_shell> date
Current system date: 2024-6-11
tiny_shell> uptime
System uptime: 4 days, 1 hours, 49 minutes, 39 seconds
tiny_shell> cpuinfo
CPU Information:
Number of processors: 12
Processor type: 8664
tiny_shell> meminfo
Memory Information:
Total physical memory: 7532 MB
Available physical memory: 1012 MB
Total virtual memory: 19620 MB
Available virtual memory: 1854 MB
tiny_shell> diskinfo D
Disk Information for drive D:\:
Total space: 372 GB
Free space: 102 GB
```

Hình 8: Chạy các câu lệnh với kịch bản kiểm thử với các tiện ích hệ thống

Notes

Lệnh nâng cao	Mô tả lệnh
<code>exit</code>	Thoát khỏi shell
<code>help</code>	Hiển thị trợ giúp về các lệnh có sẵn
<code>history</code>	Hiển thị lịch sử các lệnh đã nhập
<code>clear</code>	Xóa màn hình hiển thị của shell
<code>clear_history</code>	Xóa lịch sử các lệnh đã nhập
<code>change_color</code>	Đổi màu chữ của shell

Bảng 8: Một số câu lệnh nâng cao đối với tiện ích hệ thống

Notes

[illegible]

```
tiny_shell> change_color blue
Color changed to blue
tiny_shell> history
change_color white
clear
help
clear
change_color blue
history
tiny_shell> exit
Exiting Tiny Shell.
```

Hình 9: Chạy một số câu lệnh nâng cao kiểm thử các tiện ích hệ thống

Notes

Thực thi kịch bản

- Thư mục Testcase² chứa các kịch bản kiểm thử. Mỗi kịch bản là dãy lệnh của Tiny Shell được lưu thành một file .bat.
- Tiny Shell có thể thực hiện các câu lệnh được viết trong file .bat.

```

Testcase > Directory.bat
You, yesterday | 1 author (You)

1  pwd
2  create "New Folder"
3  cd "New Folder"
4  create "Child Folder"
5  cd ..
6  pwd
7  delete "New Folder\Child Folder"
8  cd "New Folder"
9  cd "Child Folder"
10 cd ..

11 pwd
12 delete "New Folder"
13 cd "New Folder"
14 pwd
15 create "Level 1"
16 cd "Level 1"
17 create "Level 2"
18 cd "Level 2"
19 create "Level 3"
20 cd "Level 3"

21 pwd
22 cd ..
23 pwd
24 cd ..
25 pwd
26 cd ..
27 pwd
28 delete "Level 1"
29 cd "Level 1"

```

Hình 10: Minh họa một kịch bản kiểm tra dãy lệnh với thư mục

² Mã nguồn dự án: Kịch bản kiểm thử

Notes

Tính toán nâng cao

Tiny Shell có thể hiểu một số lệnh về tính toán và điều kiện.

Lệnh	Mô tả lệnh
<code>calculate</code>	Tính giá trị một biểu thức
<code>function</code>	Định nghĩa một hàm
<code>evaluate</code>	Tính giá trị của một hàm tại giá trị cụ thể
<code>convert</code>	Chuyển đổi hệ cơ số

Bảng 9: Một số lệnh đối với tính toán và điều kiện

- Biểu thức nhập vào dưới dạng trung tố (infix), sau đó sử dụng **thuật toán Shunting Yard** chuyển biểu thức trung tố sang biểu thức hậu tố (postfix) để tính toán giá trị.
- Một số câu lệnh vòng lặp và điều kiện không thể hoàn thiện như ngôn ngữ lập trình.

Notes

[illegible]

```
tiny_shell> calculate 1+1
2
tiny_shell> calculate 6+11+2024
2041
tiny_shell> calculate 6/3+5*4*2-42
0
tiny_shell> function f(x) = x*x+x+1
- Function defined!
- Name: f
- Parameters (1): x
- Expression: x * x + x + 1
tiny_shell> evaluate f 3
13
tiny_shell> convert 1150 from 10 to 2
Result: 1000111110
tiny_shell> convert 1150 from 10 to 16
Result: 47E
```

Hình 11: Chạy các câu lệnh kiểm thử chức năng tính toán

Notes

Các tiện ích mở rộng

Shell có thể thực hiện một số tiện ích khác như:

Lệnh	Mô tả lệnh
alias	Định nghĩa viết tắt của câu lệnh
unalias	Hủy định nghĩa viết tắt của câu lệnh
bookmark	Định nghĩa tên gọi tắt cho đường dẫn
loop	Thực hiện một lệnh lặp với số vòng cụ thể
if else	Thực hiện biểu thức điều kiện

Bảng 10: Một số lệnh đối với tiện ích mở rộng

Notes

[illegible]

```
tiny_shell> alias h = history
Alias added: h -> history
tiny_shell> set_env i = 1
tiny_shell> loop 10 set_env i = i+1
tiny_shell> set_env a = 2
tiny_shell> set_env b = 3
tiny_shell> if (i+1 == 10): print_env a else print_env b
b = 3
tiny_shell> print_env a
a = 2
tiny_shell> h
clear
alias h = history
set_env i = 1
loop 10 set_env i = i+1
set_env a = 2
set_env b = 3
if (i+1 == 10): print_env a else print_env b
print_env a
h
tiny_shell> unalias h
Alias removed: h
tiny_shell> h
Unknown command: h
```

Hình 12: Chạy các câu lệnh kiểm thử các tiện ích mở rộng

Notes

Giới thiệu chung

- Khái niệm
- Windows API
- Shell

Tiny Shell

- Cài đặt và triển khai
- Các tính năng cơ bản
- Các tính năng nâng cao
- Các tiện ích mở rộng

Kết luận & Thảo luận

- Tính ứng dụng
- Khó khăn và giải pháp

Notes

Ứng dụng trong học tập

Sản phẩm **Tiny Shell**, mô phỏng giao diện dòng lệnh shell để người dùng tương tác với hệ điều hành, có nhiều tính ứng dụng quan trọng và giá trị trong học tập:

- Hiểu sâu hơn về cách hệ điều hành quản lý và thực thi các lệnh từ người dùng, cách xử lý các tiến trình và cách giao tiếp với phần cứng.
- Xây dựng một shell yêu cầu hiểu biết về lập trình hệ thống, bao gồm quản lý bộ nhớ, quản lý tiến trình và xử lý tín hiệu → **củng cố kiến thức lập trình C/C++ và hệ điều hành**.
- Nâng cao **kỹ năng thiết kế phần mềm** và quản lý dự án: xác định yêu cầu, quá trình triển khai và kiểm thử sản phẩm.

Notes

Ý nghĩa với cá nhân

- Nắm vững **khái niệm cơ bản về hệ điều hành**: tệp, thư mục, tiến trình, chương trình,...
- Hiểu sâu hơn về Windows API, các thư viện Windows và các lời gọi hệ thống.
- Học cách **ghi nhận chi tiết các lỗi** xảy ra trong quá trình phát triển, từ đó phân tích và khắc phục một cách hệ thống.
 - Thông qua việc viết Bug Record (bug.txt), lên kế hoạch kiểm thử và sửa lỗi.
 - Học cách tổ chức mã nguồn theo thư mục hợp lý, giúp mã nguồn dễ đọc, dễ bảo trì.

Notes

```
-- [05] Lỗi khi truyền toán tử: FIXED 🌟
function f(x+y,z)
function f(f(x))
function f(())
--> [KHÔNG CHO PHÉP ĐỊNH NGHĨA HÀM NHƯ VẬY]

-- [06] Lỗi không thể dừng tiến trình: SOLVED ✅
start notepad.exe // notepad lúc này có PID
terminate PID của notepad // -> Không được trên Windows 11 | Được trên Windows 10
--> [SỰ KHÁC BIỆT CỦA WINDOWS 11]
Khi bắt đầu 1 tiến trình, nó tạo ra tiến trình con (hiển thị notepad) rồi lập tức kết thúc
Như vậy muốn dừng tiến trình notepad, phải tìm PID của tiến trình con nó tạo ra và sử dụng PID đó.
```

Hình 13: Bản ghi lỗi³ và cách khắc phục lỗi

- Áp dụng các nguyên lý **lập trình hướng đối tượng** trong việc thiết kế và triển khai các tính năng trong Tiny Shell.
- Làm việc nhóm hiệu quả, chia sẻ kiến thức và phân công công việc một cách hợp lý.

³ Mã nguồn dự án: Bản ghi lỗi

Notes

[illegible]

Khó khăn

- Việc xây dựng bộ phân tích cú pháp để hiểu và xử lý đúng các lệnh từ người dùng gặp nhiều thách thức.
 - Đảm bảo rằng các lệnh sai cú pháp được phát hiện và thông báo lỗi một cách rõ ràng cho người dùng.
 - Xử lý các đường dẫn tệp, bao gồm đường dẫn tương đối và tuyệt đối.
 - Quản lý các tín hiệu hệ thống như SIGINT (Ctrl + C).
- Khó khăn khi xử lý các lệnh có cú pháp phức tạp như lệnh điều kiện, vòng lặp, và các biểu thức toán học.
- Đối mặt với các lỗi phát sinh, thiếu nhất quán trong tổ chức mã nguồn.

Notes

Giải pháp

- Thực hiện tối ưu hóa mã nguồn để Tiny Shell hoạt động hiệu quả, không tiêu tốn quá nhiều tài nguyên hệ thống.
- Liên tục kiểm thử, tối ưu hóa và sửa lỗi, phát triển các kịch bản kiểm thử với độ khó cao.
- Thử thách trước cú pháp và tính năng phức tạp.

Notes

**Cảm ơn thầy và các bạn
đã lắng nghe!**

- **Email:** tuankiet.nv2501@gmail.com
- **GitHub:** [HaiAu2501](#)
- **Phone:** +84978 621 832

Notes
