

Họ và tên: Nguyễn Việt Hoàng – Võ Nguyễn Thái Học – Ngô Hồng Phúc

Mã số sinh viên: 22520471 – 22520489 – 22521124

Lớp: IT007.O23.1

HỆ ĐIỀU HÀNH

BÁO CÁO LAB 6

Phân công:

- 1. Nguyễn Việt Hoàng:** bài 1, 3
- 2. Võ Nguyễn Thái Học:** bài 5
- 3. Ngô Hồng Phúc:** bài 2, 4

CHECKLIST

6.4. BÀI TẬP THỰC HÀNH

	Câu 1	Câu 2	Câu 3	Câu 4	Câu 5
Trình bày giải thuật	X	X	X	X	X
Chụp hình minh chứng (chạy ít nhất 3 lệnh)	X	X	X	X	X
Giải thích code, kết quả	X	X	X	X	X

Tự chấm điểm:

**Lưu ý: Xuất báo cáo theo định dạng PDF, đặt tên theo cú pháp:*

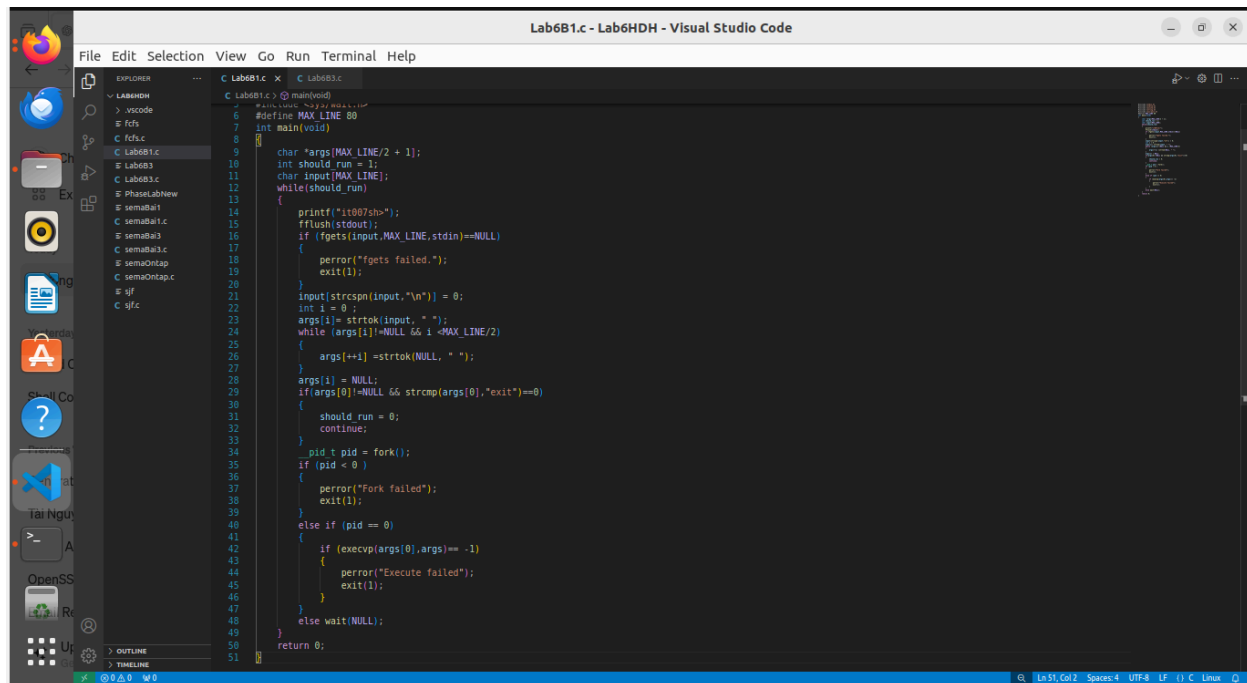
<Tên nhóm>_LAB6.pdf

6.4. BÀI TẬP THỰC HÀNH

1. Câu 1: Thực thi command trong tiến trình con

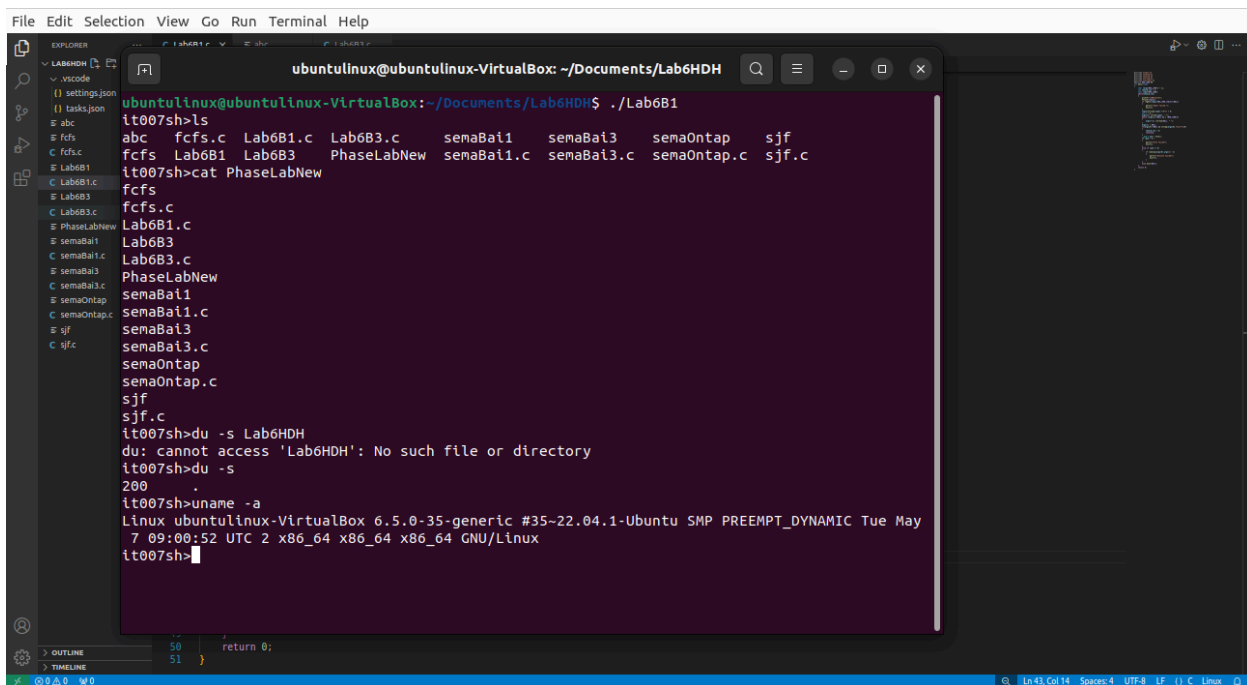
Trả lời:

- a. Trình bày giải thuật
 - Đầu tiên ta khai báo các biến:
 - `char *args[MAX_LINE/2 + 1]`: Mảng `*args` với số lượng là `MAX_LINE/2+1`, dùng để lưu các đối số của lệnh nhập vào.
 - `int should_run = 1`: tạo biến để kiểm soát vòng lặp của chương trình chính, giá trị = 1 là true.
 - `char input[MAX_LINE]`: lưu dòng lệnh nhập vào từ người dùng.
 - Tiếp đó, trong vòng lặp chính `while(should_run)`, ta thực hiện:
 - In ra dấu nhắc “it007sh>” .
 - Dùng `if` kiểm tra xem lệnh nhập vào từ người dùng có bằng `NULL` hay không bằng lệnh `fgets`, nếu có thì xin ra thông báo lỗi và thoát.
 - Sau đó, dùng `strcspn` để loại bỏ kí tự “ “ ở cuối.
 - Ta chia chuỗi nhập vào thành các từ, tách bởi khoảng trắng và lưu vào mảng `args` với lệnh `strtok(input, “ “)`. Dùng `while (args[i] != NULL && i < MAX_LINE/2)`: Tiếp tục tách và lưu các từ cho đến khi không còn từ nào hoặc đạt đến giới hạn số lượng từ. Sau đó, ta kiểm tra điều kiện kết thúc dấu nhắc bằng việc so sánh chuỗi nhập vào với “exit”. Rồi đặt lại `should_run = 0`;
 - Cuối cùng, ta tạo một tiến trình con. Sau đó kiểm tra xem lệnh `fork()` thành công hay chưa, nếu `pid < 0` thì thất bại; nếu thành công ở tiến trình con thì dùng `if (execvp(args[0], args) == -1)` với mục đích thay thế tiến trình con bằng chương trình mới được chỉ định bởi `args`. Sau cùng, nếu `pid > 0` thì dùng `wait` để đợi tiến trình con hoàn thành.
- b. Chụp hình minh chứng
Hình ảnh đoạn code:



```
Lab6B1.c - Lab6HDH - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  Lab6HDH
    .vscode
    fdfs
    fdfs.c
    Lab6B1.c
    Lab6B3
    Lab6B3.c
    PhaseLabNew
    semaBai1
    semaBai1.c
    semaBai3
    semaBai3.c
    semaOntap
    semaOntap.c
    sjf
    sjf.c
  OUTLINE
  TIMELINE
  C Lab6B1.c
  C Lab6B3.c
  main(void)
  #define MAX_LINE 80
  int main(void)
  {
    char *args[MAX_LINE/2 + 1];
    int should_run = 1;
    char input[MAX_LINE];
    while(should_run)
    {
      printf("it007sh>");
      fflush(stdout);
      if (fgets(input, MAX_LINE, stdin) == NULL)
      {
        perror("fgets failed.");
        exit(1);
      }
      input[strlen(input) - 1] = 0;
      int i = 0;
      args[i] = strtok(input, " ");
      while (args[i] != NULL && i < MAX_LINE/2)
      {
        args[i] = strtok(NULL, " ");
      }
      args[i] = NULL;
      if (args[0] != NULL && strcmp(args[0], "exit") == 0)
      {
        should_run = 0;
        continue;
      }
      _pid_t pid = fork();
      if (pid < 0)
      {
        perror("Fork failed");
        exit(1);
      }
      else if (pid == 0)
      {
        if (execvp(args[0], args) == -1)
        {
          perror("Execute failed");
          exit(1);
        }
      }
      else wait(NULL);
    }
    return 0;
  }
```

Hình ảnh chạy code:



```
ubuntu@ubuntu-VirtualBox: ~/Documents/Lab6HDH
ubuntu@ubuntu-VirtualBox:~/Documents/Lab6HDH$ ./Lab6B1
it007sh>ls
abc  fdfs.c  Lab6B1.c  Lab6B3.c  semaBai1  semaBai3  semaOntap  sjf
fdfs  Lab6B1  Lab6B3  PhaseLabNew  semaBai1.c  semaBai3.c  semaOntap.c  sjf.c
it007sh>cat PhaseLabNew
fdfs.c
Lab6B1.c
Lab6B3
Lab6B3.c
PhaseLabNew
semaBai1
semaBai1.c
semaBai3
semaBai3.c
semaOntap
semaOntap.c
sjf
sjf.c
it007sh>du -s Lab6HDH
du: cannot access 'Lab6HDH': No such file or directory
it007sh>du -s
200
it007sh>uname -a
Linux ubuntu-VirtualBox 6.5.0-35-generic #35-22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue May
7 09:00:52 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
it007sh>
```

- c. Giải thích code, kết quả: sử dụng dấu nhắc it007sh>
- Dùng lệnh ls để liệt kê các tệp có trong Folder đang được mở trong Terminal
 - Sau đó dùng lệnh cat PhaseLabNew để xem nội dung trong tệp PhaseLabNew

- Với lệnh `du -s <tên folder>`, ta dùng để xuất ra dung lượng của folder đó
- Lệnh `uname -a`: dùng để lấy thông tin của máy ảo.

2. Câu 2: History Feature

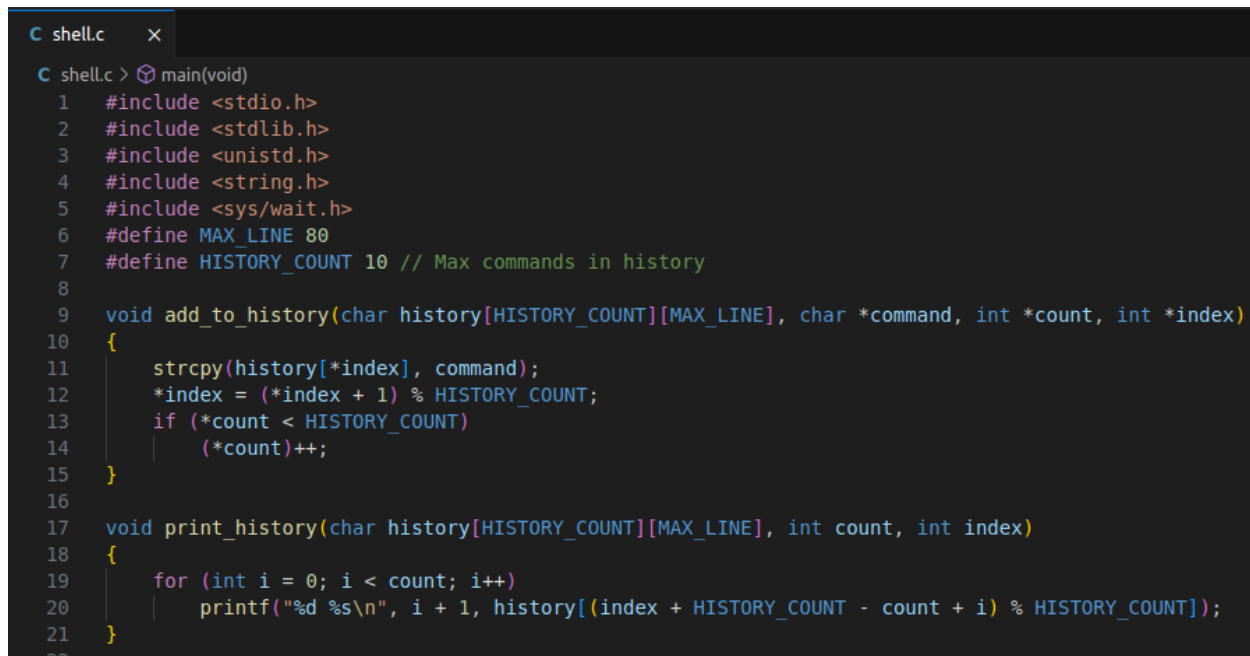
- Hàm `add_to_history`: thêm các lệnh vào lịch sử. Sử dụng một mảng vòng (**circular buffer**) để quản lý lịch sử lệnh, giúp lưu trữ một số lượng lệnh giới hạn nhất định

+ `strcpy(history[*index], command)`: sao chép chuỗi **command** vào vị trí hiện tại trong mảng **history**, vị trí này được xác định bởi ***index**.

+ `*index = (*index + 1) % HISTORY_COUNT`: Tăng index lên 1 và chia lấy dư (%) với **HISTORY_COUNT** để đảm bảo index luôn nằm trong khoảng **[0; HISTORY_COUNT-1]**. Điều này tạo ra một mảng vòng, khi index đạt đến **HISTORY_COUNT**, nó sẽ quay lại 0.

+ `if (*count < HISTORY_COUNT) (*count)++`: Nếu số lệnh hiện có nhỏ hơn **HISTORY_COUNT**, tăng biến đếm (***count**) lên 1. Điều này đảm bảo rằng số lượng lệnh không vượt quá giới hạn **HISTORY_COUNT**.

- Hàm `print_history`: in các lệnh trong lịch sử theo thứ tự thời gian gần đây nhất.



```
C shell.c x
C shell.c > main(void)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <string.h>
5  #include <sys/wait.h>
6  #define MAX_LINE 80
7  #define HISTORY_COUNT 10 // Max commands in history
8
9  void add_to_history(char history[HISTORY_COUNT][MAX_LINE], char *command, int *count, int *index)
10 {
11     strcpy(history[*index], command);
12     *index = (*index + 1) % HISTORY_COUNT;
13     if (*count < HISTORY_COUNT)
14         (*count)++;
15 }
16
17 void print_history(char history[HISTORY_COUNT][MAX_LINE], int count, int index)
18 {
19     for (int i = 0; i < count; i++)
20         printf("%d %s\n", i + 1, history[(index + HISTORY_COUNT - count + i) % HISTORY_COUNT]);
21 }
22
```

- Nếu nhập lệnh HF, sẽ in ra lịch sử lệnh.
- Chương trình sẽ thêm những lệnh không phải HF vào lịch sử.

```
C shell.c x
C shell.c > main(void)
26
27 int main(void)
28 {
29     char *args[MAX_LINE / 2 + 1];
30     int should_run = 1;
31     char input[MAX_LINE];
32     char history[HISTORY_COUNT][MAX_LINE]; // History buffer
33     int history_count = 0;                  // Number of commands in history
34     int history_index = 0;                  // Current index in history buffer
35     while (should_run)
36     {
37         printf("it007sh>");
38         fflush(stdout);
39         if (fgets(input, MAX_LINE, stdin) == NULL)
40         {
41             perror("fgets failed.");
42             exit(1);
43         }
44         input[strcspn(input, "\n")] = 0;
45
46         // HF
47         if (strcmp(input, "HF") == 0)
48         {
49             print_history(history, history_count, history_index);
50             continue;
51         }
52
53         // Add the command to history if it's not 'HF'
54         add_to_history(history, input, &history_count, &history_index);
55
56         // Tokenize the input command
57         int i = 0;
58         args[i] = strtok(input, " ");
59         while (args[i] != NULL && i < MAX_LINE / 2)
60         {
61             args[++i] = strtok(NULL, " ");
62         }
63         args[i] = NULL;
64         if (args[0] != NULL && strcmp(args[0], "exit") == 0)
65         {
66             should_run = 0;
```

```
fud@fud-Nitro-AN515-58: ~/He_dieu_hanh
fud@fud-Nitro-AN515-58:~/He_dieu_hanh$ ./shell
it007sh>ls
ans_mutex    consumer    mutex       producer    rr.cpp      shell.c
ans_mutex.c  consumer.c  mutex.c     producer.c  shell
it007sh>pwd
/home/fud/He_dieu_hanh
it007sh>which make
/usr/bin/make
it007sh>whoami
fud
it007sh>HF
1 ls
2 pwd
3 which make
4 whoami
it007sh>
```

3. Câu 3: Chuyển hướng vào ra

Trả lời:

- a. Trình bày giải thuật
 - Đoạn đầu giống với câu 1.
 - Trong đoạn tạo tiến trình con:
 - o Ở phần kiểm tra xem tiến trình hiện tại có phải tiến trình con hay không, nếu đúng thì :
 - Khởi tạo biến: `int in_redir = -1, out_redir = -1`;: Các biến này được sử dụng để lưu trữ chỉ mục của các ký hiệu chuyển hướng (< cho đầu vào, > cho đầu ra) trong mảng `args` (có khả năng chứa các đối số lệnh). Ban đầu, chúng được đặt thành -1, cho biết không có chuyển hướng nào.
 - Dùng vòng lặp `for (int j = 0; args[j] != NULL; j++)`: Vòng lặp này lặp qua mảng `args` cho đến khi gặp dấu chấm kết thúc (NULL), biểu thị cuối danh sách đối số.
 - `if (strcmp(args[j], "<") == 0)`: Điều kiện này kiểm tra xem đối số hiện tại (`args[j]`) có bằng chuỗi "<" hay không, cho biết chuyển hướng đầu vào. Nếu đúng, `in_redir` được đặt thành chỉ mục hiện tại `j`, đánh dấu vị trí của ký hiệu chuyển hướng. `if (strcmp(args[j], ">") == 0)`: Tương tự, điều kiện này kiểm tra ký hiệu chuyển hướng đầu ra (>). Nếu đúng, `out_redir` được đặt thành `j`.
 - Ta xử lý nếu là đầu vào: `int fd = open(args[in_redir + 1], O_RDONLY)`;: Dòng này mở tệp được chỉ định sau ký hiệu

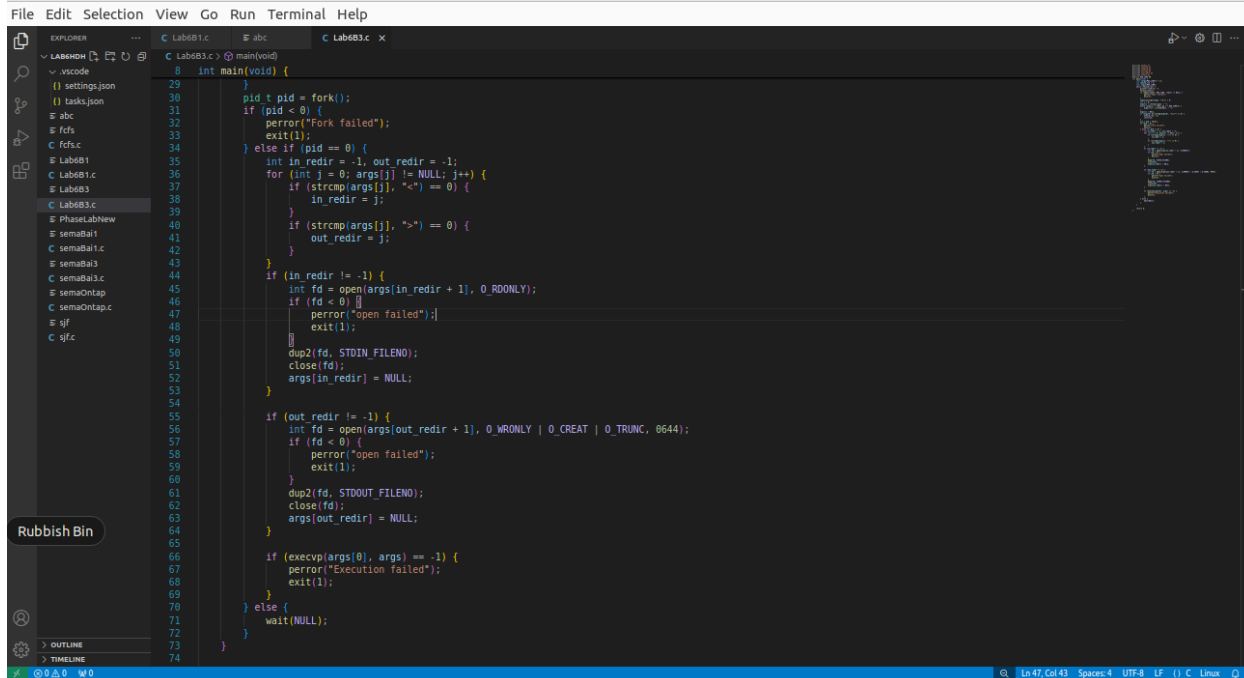
chuyển hướng (args[in_redir + 1]). Flag O_RDONLY chỉ định quyền truy cập chỉ đọc. if (fd < 0):

- Ta xử lí đầu ra: `int fd = open(args[out_redir + 1], O_WRONLY | O_CREAT | O_TRUNC, 0644);`; Dòng này mở tệp được chỉ định sau ký hiệu chuyển hướng. `WRONLY`: Cho phép ghi vào tệp. `O_CREAT`: Tạo tệp nếu nó không tồn tại. `O_TRUNC`: Cắt ngắn bất kỳ nội dung hiện có trong tệp thành độ dài bằng không. `0644`: Đặt quyền truy cập tệp cho phép chủ sở hữu đọc/ghi, nhóm và những người khác chỉ đọc.
- `Dup2(fd, STDIN_FILENO);`; Dòng này chuyển hướng đầu vào chuẩn (`STDIN`) của tiến trình con sang trình mô tả tệp đã mở (`fd`). Điều này có nghĩa là bất cứ điều gì được đọc từ tệp sẽ được coi là đầu vào chuẩn của tiến trình con.

b. Chụp hình minh chứng

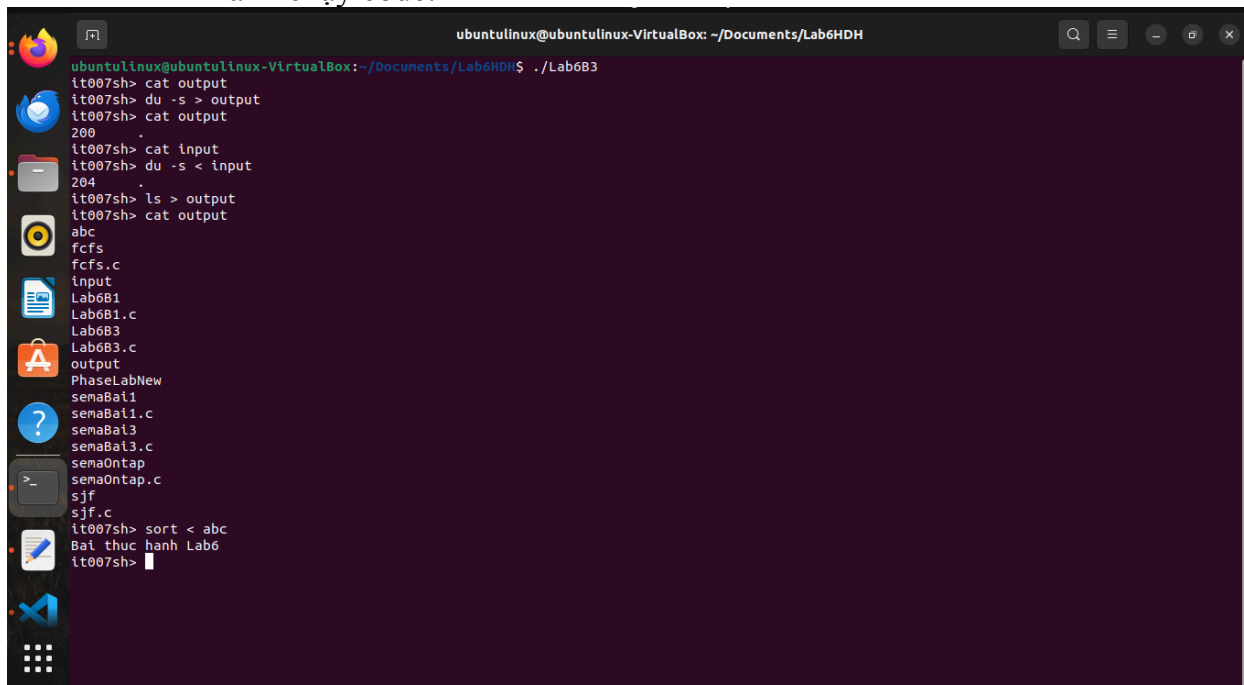
Hình ảnh code:

```
File Edit Selection View Go Run Terminal Help
EXPLORER ... F abc C Lab6B3.c x
LAB6B3H
└─ vscode
└─ {} settings.json
└─ {} tasks.json
└─ F abc
└─ F fdfs
└─ C fdfs.c
└─ E Lab6B1
└─ C Lab6B1.c
└─ E Lab6B3
└─ C Lab6B3.c
└─ E PhaseLabNew
└─ E semaBai1
└─ C semaBai1.c
└─ E semaBai3
└─ C semaBai3.c
└─ E semaOntap
└─ C semaOntap.c
└─ E sf
└─ C sf.c
C Lab6B3.c
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <sys/wait.h>
6 #include <fcntl.h>
7 #define MAX_LINE 80
8 int main(void) {
9     char *args[MAX_LINE/2 + 1];
10    int should_run = 1;
11    char input[MAX_LINE];
12    while (should_run) {
13        printf("fdfs> ");
14        fflush(stdout);
15        if (fgets(input, MAX_LINE, stdin) == NULL) {
16            perror("fgets failed");
17            exit(1);
18        }
19        input[strlen(input, "\n")] = 0;
20        int i = 0;
21        args[i] = strtok(input, " ");
22        while (args[i] != NULL && i < MAX_LINE/2) {
23            args[++i] = strtok(NULL, " ");
24        }
25        args[i] = NULL;
26        if (args[0] && strcmp(args[0], "exit") == 0) {
27            should_run = 0;
28            continue;
29        }
30        pid_t pid = fork();
31        if (pid < 0) {
32            perror("fork failed");
33            exit(1);
34        } else if (pid == 0) {
35            int in_redir = -1, out_redir = -1;
36            for (int j = 0; args[j] != NULL; j++) {
37                if (strcmp(args[j], "<"*) == 0) {
38                    in_redir = j;
39                }
40                if (strcmp(args[j], ">") == 0) {
41                    out_redir = j;
42                }
43            }
44            if (in_redir != -1) {
45                int fd = open(args[in_redir + 1], O_RDONLY);
46                if (fd < 0) {
47                    perror("open failed");
```



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  Lab6HDH
    settings.json
    tasks.json
    abc
    fcfs
    fcfs.c
    Lab6B1
    Lab6B1.c
    Lab6B3
    Lab6B3.c
    PhaseLabNew
    semaBai1
    semaBai1.c
    semaBai3
    semaBai3.c
    semaOntap
    semaOntap.c
    sjf
    sjf.c
  Rubbish Bin
  OUTLINE
  TIMELINE
  C Lab6B3.c
  E abc
  C Lab6B3.c
  main(void)
  8 int main(void) {
  29 }
  30 pid_t pid = fork();
  31 if (pid < 0) {
  32     perror("Fork failed");
  33     exit(1);
  34 } else if (pid == 0) {
  35     int in_redir = -1, out_redir = -1;
  36     for (int j = 0; args[j] != NULL; j++) {
  37         if (strcmp(args[j], "<") == 0) {
  38             in_redir = j;
  39         }
  40         if (strcmp(args[j], ">") == 0) {
  41             out_redir = j;
  42         }
  43     }
  44     if (in_redir != -1) {
  45         int fd = open(args[in_redir + 1], O_RDONLY);
  46         if (fd < 0) {
  47             perror("open failed");
  48             exit(1);
  49         }
  50         dup2(fd, STDIN_FILENO);
  51         close(fd);
  52         args[in_redir] = NULL;
  53     }
  54     if (out_redir != -1) {
  55         int fd = open(args[out_redir + 1], O_WRONLY | O_CREAT | O_TRUNC, 0644);
  56         if (fd < 0) {
  57             perror("open failed");
  58             exit(1);
  59         }
  60         dup2(fd, STDOUT_FILENO);
  61         close(fd);
  62         args[out_redir] = NULL;
  63     }
  64     if (execvp(args[0], args) == -1) {
  65         perror("Execution failed");
  66         exit(1);
  67     } else {
  68         wait(NULL);
  69     }
  70 }
  71 }
  72 }
  73 }
  74 }
```

Hình ảnh chạy code:



```
ubuntulinux@ubuntulinux-VirtualBox: ~/Documents/Lab6HDH
it007sh> cat output
it007sh> du -s > output
it007sh> cat output
200
it007sh> cat input
it007sh> du -s < input
204
it007sh> ls > output
it007sh> cat output
abc
fcfs
fcfs.c
input
Lab6B1
Lab6B1.c
Lab6B3
Lab6B3.c
output
PhaseLabNew
semaBai1
semaBai1.c
semaBai3
semaBai3.c
semaOntap
semaOntap.c
sjf
sjf.c
it007sh> sort < abc
Bai thuc hanh Lab6
it007sh>
```

- c. Giải thích code, kết quả
- Đầu tiên, ta dùng lệnh cat output để xem nội dung của tệp output.txt, do không có nội dung nên trả về dấu nhắc it007sh>

- Sau đó ta dùng lệnh `du -s > output`: nhằm chuyển hướng đầu ra của lệnh `du -s` sang tệp `output.txt`, cụ thể là chuyển nội dung hiện thị trên terminal sang `output.txt`. Rồi ta dùng lệnh `cat output` để kiểm tra thành công hay chưa.
- Tương tự ta có lệnh `du -s < input`, ở đây ta dùng dấu chuyển hướng đầu vào "<", tức là dùng lệnh `du -s` đọc dung lượng từ tệp `input.txt`.
- Và ta thử lại với lệnh `it007>ls>output` và `it007>sort<abc`.

4. Câu 4: Pipe

- Hàm `parse_input`: phân tích chuỗi lệnh đầu vào để tìm "|". Nếu có "|", chia chuỗi lệnh thành hai phần: lệnh trước "|" lưu trong `args` và lệnh sau "|" lưu trong `args2`. Biến `has_pipe` được đặt để cho biết sự tồn tại của "|".

```
C shell.c x
C shell.c > main(void)
22
23 void parse_input(char *input, char **args, char **args2, int *has_pipe)
24 {
25     int i = 0;
26     *has_pipe = 0;
27     args[i] = strtok(input, " ");
28     while (args[i] != NULL)
29     {
30         if (strcmp(args[i], "|") == 0)
31         {
32             *has_pipe = 1;
33             args[i] = NULL;
34             int j = 0;
35             args2[j] = strtok(NULL, " ");
36             while (args2[j] != NULL)
37             {
38                 args2[++j] = strtok(NULL, " ");
39             }
40             break;
41         }
42         args[++i] = strtok(NULL, " ");
43     }
44     args[i] = NULL;
45 }
```

```
C shell.c x
C shell.c > main(void)
46
47 int main(void)
48 {
49     char *args[MAX_LINE / 2 + 1];
50     char *args2[MAX_LINE / 2 + 1];
51     int should_run = 1;
52     char input[MAX_LINE];
53     char history[HISTORY_COUNT][MAX_LINE]; // History buffer
54     int history_count = 0; // Number of commands in history
55     int history_index = 0; // Current index in history buffer
56     while (should_run)
57     {
58         printf("it007sh>");
59         fflush(stdout);
60         if (fgets(input, MAX_LINE, stdin) == NULL)
61         {
62             perror("fgets failed.");
63             exit(1);
64         }
65         input[strcspn(input, "\n")] = 0;
66
67         // HF
68         if (strcmp(input, "HF") == 0)
69         {
70             print_history(history, history_count, history_index);
71             continue;
72         }
73
74         // Add the command to history if it's not 'HF'
75         add_to_history(history, input, &history_count, &history_index);
76
77         int has_pipe;
78         parse_input(input, args, args2, &has_pipe);
79
80         // Check for 'exit' command to terminate the shell
81         if (args[0] != NULL && strcmp(args[0], "exit") == 0)
82         {
83             should_run = 0;
84             continue;
85         }
86
87         // Fork a child process to execute the command
88         __pid_t pid = fork();
89         if (pid < 0)
90         {
```

```
C shell.c x
C shell.c > main(void)
47  int main(void)
89      if (pid < 0)
90      {
91          perror("Fork failed");
92          exit(1);
93      }
94      else if (pid == 0)
95      {
96          if (has_pipe)
97          {
98              int fd[2];
99              if (pipe(fd) == -1)
100             {
101                 perror("Pipe failed");
102                 exit(1);
103             }
104             __pid_t pid2 = fork();
105             if (pid2 < 0)
106             {
107                 perror("Fork failed");
108                 exit(1);
109             }
110             else if (pid2 == 0)
111             {
112                 close(fd[0]);
113                 dup2(fd[1], STDOUT_FILENO);
114                 close(fd[1]);
115                 if (execvp(args[0], args) == -1)
116                 {
117                     perror("Execute failed");
118                     exit(1);
119                 }
120             }
121             else
122             {
123                 close(fd[1]);
124                 dup2(fd[0], STDIN_FILENO);
125                 close(fd[0]);
126                 if (execvp(args2[0], args2) == -1)
127                 {
128                     perror("Execute failed");
129                     exit(1);
130                 }
131             }
132         }
133     }
```

* Nếu có "|"

- Tạo đường ống:

- `int fd[2];`: mảng `fd` để lưu 2 đầu đường ống.
- `if (pipe(fd) == -1)`: Tạo đường ống bằng hàm `pipe(fd)`. Nếu tạo đường ống thất bại, `pipe` trả về -1.
- `perror("Pipe failed");`: In ra thông báo lỗi nếu tạo đường ống thất bại.
- `exit(1);`: Thoát chương trình với mã lỗi 1 nếu tạo đường ống thất bại.

- Tạo tiến trình con thứ 2:

- `__pid_t pid2 = fork();`: Tạo một tiến trình con thứ hai.
- `if (pid2 < 0)`: Kiểm tra nếu `fork` trả về giá trị âm, nghĩa là tạo tiến trình con thất bại.

- Xử lý trong tiến trình con thứ 2:

- `else if (pid2 == 0)`: Kiểm tra nếu `pid2` bằng 0, nghĩa là đang trong tiến trình con thứ hai.
- `close(fd[0]);`: Đóng đầu đọc của đường ống vì tiến trình này sẽ ghi vào đường ống.
- `dup2(fd[1], STDOUT_FILENO);`: Sao chép đầu ghi của đường ống vào đầu ra chuẩn (STDOUT).
- `close(fd[1]);`: Đóng đầu ghi của đường ống sau khi đã sao chép.
- `if (execvp(args[0], args) == -1)`: Thực thi lệnh đầu tiên. Nếu thất bại, `execvp` trả về -1.

- Xử lý trong tiến trình cha của tiến trình con thứ 2:

- `else`: Tiến trình cha của tiến trình con thứ hai sẽ thực thi lệnh thứ hai.
- `close(fd[1]);`: Đóng đầu ghi của đường ống vì tiến trình này sẽ đọc từ đường ống.

Báo cáo thực hành môn Hệ điều hành - Giảng viên: Trần Hoàng Lộc.

- `dup2(fd[0], STDIN_FILENO);`: Sao chép đầu đọc của đường ống vào đầu vào chuẩn (STDIN).
- `close(fd[0]);`: Đóng đầu đọc của đường ống sau khi đã sao chép.
- `if (execvp(args2[0], args2) == -1)`: Thực thi lệnh thứ hai. Nếu thất bại, `execvp` trả về -1.

```
fud@fud-Nitro-AN515-58: ~/He_dieu_hanh$ ./shell
it007sh-ps aux | grep firefox
fud      3182  20.3  3.4 12737504 548364 ?        Sl   11:20   5:50 /snap/firefox/4136/usr/lib/firefox/firefox
fud      3655  0.0  0.2 210604 47680 ?        Sl   11:20   0:00 /snap/firefox/4136/usr/lib/firefox/firefox
appDir /snap/firefox/4136/usr/lib/firefox/browser (169c4859-7228-4c2f-8a9a-ad005c2032e) 3182 true socket
fud      3832  0.1  0.7 2464960 125432 ?        Sl   11:20   0:02 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 1 -isForBrowser -prefsLen 31480 -prefMapSize 242198 -jsInI
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser (edfa891d-8866-4848-afbc-ec22dc64548) 3182 true tab
fud      3965  0.0  0.5 2443532 95820 ?        Sl   11:20   0:00 /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {4e84dc0-ca75-4acb-9176-575b75a89513} 3182 true tab
fud      4130  0.0  0.3 346140 60796 ?        Sl   11:20   0:00 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -parentBuildID 20240410171251 -sandboxingKind 0 -prefsLen 37024 -pr
efMapSize 242198 -appDir /snap/firefox/4136/usr/lib/firefox/browser {66ac3c2a-6e26-4436-8008-e26a91dd5be7} 3182 true utility
fud      4134  4.7  2.9 3147840 468872 ?        Sl   11:20   1:21 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 3 -isForBrowser -prefsLen 31295 -prefMapSize 242198 -jsInI
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {c9db28c6-b248-4793-84a1-5716bf2b9232} 3182 true tab
fud      4137  0.0  0.6 2442380 98624 ?        Sl   11:20   0:00 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 4 -isForBrowser -prefsLen 31295 -prefMapSize 242198 -jsInI
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {9764a21b-9af9-40ed-b72a-felcf68c02e1} 3182 true tab
fud      4311  2.7  2.0 2910880 333944 ?        Sl   11:20   0:47 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 6 -isForBrowser -prefsLen 31484 -prefMapSize 242198 -jsInI
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {1ab7fe04-7a06-4fd0-8400-77e91797b190} 3182 true tab
fud      4378  0.0  0.7 673568 119488 ?        Sl   11:20   0:00 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -parentBuildID 20240410171251 -prefsLen 37213 -prefMapSize 242198 -
appDir /snap/firefox/4136/usr/lib/firefox/browser {21985ebe-486a-4577-bcf6-60c0b6544032} 3182 true rdd
fud      5969  3.3  1.9 2769944 313528 ?        Sl   11:22   0:54 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 10 -isForBrowser -prefsLen 31706 -prefMapSize 242198 -jsIn
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {09e45857-0e60-421b-91ee-fd7cf86ad160} 3182 true tab
fud      6000  5.2  1.4 6099740 228608 ?        Sl   11:22   1:23 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 11 -isForBrowser -prefsLen 31706 -prefMapSize 242198 -jsIn
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {8295a6d1-f99e-4d19-a6c3-1a2548b64a48} 3182 true tab
fud      6011  0.0  0.4 2397672 21680 ?        Sl   11:22   0:00 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 12 -isForBrowser -prefsLen 31706 -prefMapSize 242198 -jsIn
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {43956ee4-3be9-4070-b441-6c0163360c2c} 3182 true tab
fud      6125  1.0  0.8 2574252 139180 ?        Sl   11:22   0:16 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 13 -isForBrowser -prefsLen 37435 -prefMapSize 242198 -jsIn
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {b5bc426e-bae8-4aab-a2c8-12364c21baa} 3182 true tab
fud      6381  0.0  0.4 2397676 73760 ?        Sl   11:23   0:00 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 14 -isForBrowser -prefsLen 31706 -prefMapSize 242198 -jsIn
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {0443abdf-1993-4c8b-ae79-17da6b0851bf} 3182 true tab
fud      7630  0.0  0.4 2397672 22640 ?        Sl   11:26   0:00 /snap/firefox/4136/usr/lib/firefox/firefox -contentproc -childID 15 -isForBrowser -prefsLen 31706 -prefMapSize 242198 -jsIn
tlen 234952 -parentBuildID 20240410171251 -greomni /snap/firefox/4136/usr/lib/firefox/omni.ja -appomni /snap/firefox/4136/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/4136/usr/lib/f
firefox/browser {a4d2bc01-facb-4ada-8fe7-d7fc00777206} 3182 true tab
fud      8366  0.0  0.0 12088 2720 pts/1  S+   11:48   0:00 grep firefox
it007sh-
```

5. Câu 5: Kết thúc dấu nhắc

```
C tesst.c C labb6.c
C labb6.c > sigint_handler(int)
5  #include <sys/wait.h>
6  #include <signal.h>
7  #define MAX_LINE 80
8  int should_run = 1;
9  void sigint_handler(int sig) {}
10 {}
11
12 int main(void)
13 {
14     char *args[MAX_LINE/2 + 1];
15
16     char input[MAX_LINE];
17     while(should_run)
18     {
19         printf("it007sh>");
20         fflush(stdout);
21         if (fgets(input,MAX_LINE,stdin)==NULL)
22         {
23             perror("fgets failed.");
24             exit(1);
25         }
26         input[strcspn(input,"\n")] = 0;
27         int i = 0 ;
28         args[i]= strtok(input, " ");
29         while (args[i]!=NULL && i <MAX_LINE/2)
30         {
31             args[++i] =strtok(NULL, " ");
32         }
33         args[i] = NULL;
34         if(args[0]!=NULL && strcmp(args[0],"exit")==0)
35         {
36             should_run = 0;
37             continue;
38         }
39         __pid_t pid = fork();
40         if (pid < 0 )
41         {
42             perror("Fork failed");
43             exit(1);
44         }
45         else if (pid == 0)
46         {
47             if (execvp(args[0],args)== -1)
48             {
49                 perror("Execute failed");
50                 exit(1);
51             }
52         }
53         else
54         {
55             wait(NULL);
56             if(signal(SIGINT,sigint_handler))
57                 continue;
58         }
59     }
60     return 0;
61 }
62 }
```

Báo cáo thực hành môn Hệ điều hành - Giảng viên: Trần Hoàng Lộc.

```
Tasks: 307 total, 1 running, 305 sleeping, 0 stopped, 1 zombie
%Cpu(s): 8,3 us, 1,9 sy, 0,0 ni, 89,7 id, 0,1 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 7730,5 total, 854,8 free, 4120,2 used, 2755,4 buff/cache
MiB Swap: 2048,0 total, 2005,7 free, 42,2 used. 2110,1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR  S  %CPU  %MEM    TIME+  COMMAND
 6562 thna1108  20   0 384492 323504 10880 S 63,5   4,1 15:33.93 cpptools
 6464 thna1108  20   0 1135,6g 301460 76404 S  8,0   3,8  5:17.00 code
 6486 thna1108  20   0   32,4g 80312 68548 S  2,7   1,0  1:30.47 code
 6528 thna1108  20   0 1123,1g 79020 50488 S  1,3   1,0  0:07.41 code
1397 thna1108   9 -11 3104956 25500 20380 S  1,0   0,3  1:18.50 pulseaudio
 538 avahi      20   0 10304 6400 3456 S  0,7   0,1  0:36.98 avahi-daemon
5852 thna1108  20   0 953620 70416 37488 S  0,7   0,9  0:35.81 Xwayland
 6393 thna1108  20   0 1123,3g 141848 95300 S  0,7   1,8  0:29.32 code
 6513 thna1108  20   0 1123,2g 133320 52288 S  0,7   1,7  0:28.89 code
 6529 thna1108  20   0 1123,2g 87280 50304 S  0,7   1,1  0:38.51 code
 378 root       -51   0   0   0   0 S  0,3   0,0  0:01.52 irq/156-iwlwifi:default_queue
1590 thna1108  20   0 6053856 301436 133220 S  0,3   3,8  4:30.92 gnome-shell
2103 thna1108  20   0 3200220 60476 41196 S  0,3   0,8  0:03.43 gjs
 6314 thna1108  20   0 564424 53420 40808 S  0,3   0,7  0:07.20 gnome-terminal-
 8076 thna1108  20   0 2619020 219052 96704 S  0,3   2,8  1:07.52 Isolated Web Co
 9780 root       20   0   0   0   0 I  0,3   0,0  0:00.02 kworker/u16:3-flush-259:0
 9897 thna1108  20   0 24732 15924 10956 S  0,3   0,2  0:00.06 cpptools-srv
 9918 thna1108  20   0 16232 4480 3584 R  0,3   0,1  0:00.02 top
   1 root       20   0 168296 12884 8148 S  0,0   0,2  0:06.10 systemd
   2 root       20   0   0   0   0 S  0,0   0,0  0:00.00 kthreadd
   3 root       0 -20   0   0   0 I  0,0   0,0  0:00.00 rcu_gp
   4 root       0 -20   0   0   0 I  0,0   0,0  0:00.00 rcu_par_gp
   5 root       0 -20   0   0   0 I  0,0   0,0  0:00.00 slub_flushwq
   6 root       0 -20   0   0   0 I  0,0   0,0  0:00.00 netns
   8 root       0 -20   0   0   0 I  0,0   0,0  0:00.00 kworker/0:0H-events_highpri
  11 root       0 -20   0   0   0 I  0,0   0,0  0:00.00 mm_percpu_wq
  12 root       20   0   0   0   0 I  0,0   0,0  0:00.00 rcu_tasks_kthread
  13 root       20   0   0   0   0 I  0,0   0,0  0:00.00 rcu_tasks_rude_kthread
  14 root       20   0   0   0   0 I  0,0   0,0  0:00.00 rcu_tasks_trace_kthread
  15 root       20   0   0   0   0 S  0,0   0,0  0:00.90 ksoftirqd/0
  16 root       20   0   0   0   0 I  0,0   0,0  0:02.92 rcu_preempt
  17 root       rt   0   0   0   0 S  0,0   0,0  0:00.01 migration/0
  18 root      -51   0   0   0   0 S  0,0   0,0  0:00.00 idle_inject/0
  19 root       20   0   0   0   0 S  0,0   0,0  0:00.00 cpuhp/0
  20 root       20   0   0   0   0 S  0,0   0,0  0:00.00 cpuhp/1
  21 root      -51   0   0   0   0 S  0,0   0,0  0:00.00 idle_inject/1
  22 root       rt   0   0   0   0 S  0,0   0,0  0:00.07 migration/1
  23 root       20   0   0   0   0 S  0,0   0,0  0:00.64 ksoftirqd/1
  25 root       0 -20   0   0   0 I  0,0   0,0  0:00.00 kworker/1:0H-events_highpri
  26 root       20   0   0   0   0 S  0,0   0,0  0:00.00 cpuhp/2
  27 root      -51   0   0   0   0 S  0,0   0,0  0:00.00 idle_inject/2
  28 root       rt   0   0   0   0 S  0,0   0,0  0:00.07 migration/2
  29 root       20   0   0   0   0 S  0,0   0,0  0:00.38 ksoftirqd/2
  31 root       0 -20   0   0   0 I  0,0   0,0  0:00.00 kworker/2:0H-events_highpri
  32 root       20   0   0   0   0 S  0,0   0,0  0:00.00 cpuhp/3
  33 root      -51   0   0   0   0 S  0,0   0,0  0:00.00 idle_inject/3
  34 root       rt   0   0   0   0 S  0,0   0,0  0:00.07 migration/3
  35 root       20   0   0   0   0 S  0,0   0,0  0:00.33 ksoftirqd/3
tt007sh>
```

Giải thích code:

- Đoạn code ở câu 5 tương tự như ở câu 1, khác ở đây là khi tiến trình đang hoạt động, ta thêm vô câu kiểm tra xem là có tín hiệu SIGINT được nhập vào không, nếu có chương trình sẽ tiếp tục vòng lặp for và dừng tiến trình lại.