

Kiểm Tra Thực Hành

Total points 88/100 ?

The respondent's email (**52100813@student.tdtu.edu.vn**) was recorded on submission of this form.

19 of 21 points

Lệnh clear dùng để làm gì?

2/2

- ☒ Xóa trắng màn hình làm việc
- ☐ Xóa tập tin được chỉ định
- ☐ Xóa trắng thư mục làm việc
- ☐ Phục hồi trạng thái màn hình trước đó

Để cập nhật các ứng dụng đã có trong hệ thống Linux, cần dùng lệnh gì?

3/3

- ☐ apt-update
- ☒ sudo apt-get update
- ☐ apt-get
- ☐ sudo apt-update



Để trở lại dấu nhắc lệnh khi một tiến trình đang chạy, cần thao tác gì?

3/3

- ☒ Nhấn tổ hợp Ctrl + Z
- ☐ Nhấn tổ hợp Ctrl + Alt + Del
- ☐ Gõ lệnh terminal
- ☐ Gõ lệnh pause

Lệnh cấp quyền quản trị cho một số lệnh đặc quyền là gì?

2/2

- ☒ sudo
- ☐ admin
- ☐ apt-get
- ☐ super

Lệnh rm dùng để làm gì?

.../2

- ☐ Biên tập tập tin
- ☒ Xóa tập tin
- ☐ Di chuyển thư mục
- ☐ Tạo mới thư mục



Lệnh dùng để "tìm hiểu về một lệnh nào đó" là gì?

2/2

- ☐ guide
- ☐ help
- ☐ ask
- ☒ man

Lệnh nào liệt kê nội dung thư mục đang làm việc?

2/2

- ☒ ls
- ☐ list
- ☐ ps
- ☐ dir

Để biết thông tin về trình biên dịch cần dùng lệnh gì?

2/2

- ☒ gcc --version
- ☐ gcc version
- ☐ man gcc
- ☐ gcc -version



Để đi lên thư mục cha, cần dùng lệnh nào?

2/2

- ☒ `cd ..`
- ☐ `cd ~`
- ☐ `cd /`
- ☐ `cd parent`

Tập tin khả thực thi thể hiện như thế nào trong Linux?

1/1

- ☐ đối tượng thể hiện TÊN IN HOA
- ☐ đối tượng không có đặc trưng
- ☒ đối tượng có màu xanh lục
- ☐ đối tượng kèm `.out` sau tên

LAB TEST - 02

27 of 27 points

Để có được tập tin khả thực thi `Run.out`, cần sử dụng lệnh biên dịch nào? Giả sử thư mục làm việc đang tồn tại tập tin `vidu.c`

- ☐ `gcc -c Run.out vidu.c`
- ☒ `gcc -o Run.out vidu.c`
- ☐ `gcc vidu.c -out Run.out`
- ☐ `gcc vidu.c`



Để thực thi một tập tin Run.out đã tồn tại cần dùng lệnh gì?

2/2

- ☐ run Run.out
- ☐ Run.out
- ☒ ./Run.out
- ☐ exec Run.out

Đối với biên dịch liên kết động, chọn tất cả phát biểu đúng

3/3

- ☐ Máy tính khác có thể thực thi tập tin đầu ra mà không cần thư viện.
- ☒ Luôn kèm theo thư viện khi đóng gói phát hành.
- ☐ Các thư viện không còn cần thiết sau khi biên dịch thành công.
- ☒ Tiết kiệm dung lượng cho tập tin đầu ra.
- ☒ Tiết kiệm bộ nhớ máy tính khi thực thi.

Để thực hiện truyền đối số qua lời gọi, hàm main cần khai báo như thế nào?

3/3

- ☐ int main(int n, string argv[n])
- ☒ int main(int n, char ** list)
- ☐ int main(int argc, char * argv)
- ☐ int main(int argc, char argv)



Đối với biên dịch liên kết tĩnh, chọn tất cả phát biểu đúng

3/3

- ☒ Các thư viện không còn cần thiết sau khi biên dịch thành công.
- ☒ Máy tính khác có thể thực thi tập tin đầu ra mà không cần thư viện.
- ☐ Luôn kèm theo thư viện khi đóng gói phát hành.
- ☒ Tập tin khả thực thi có kích thước rất lớn.
- ☐ Tiết kiệm dung lượng cho tập tin đầu ra.

Câu lệnh "int x = argv[2];" sẽ thực thi như thế nào? Giả sử truyền đủ đối số cần 3/3 khi thực thi.

- ☐ x nhận giá trị của đối số thứ ba
- ☐ x nhận giá trị của đối số thứ hai
- ☐ câu lệnh gây nên lỗi không đồng nhất kiểu dữ liệu
- ☒ câu lệnh tạo cảnh báo và x không nhận được đối số

Để liên kết các tập tin đối tượng a.o và b.o thành tập tin khả thực thi Run.out 3/3 cần dùng lệnh gì?

- ☐ gcc -c Run.out a.o b.o
- ☐ gcc -o a.o -o b.o -c Run.out
- ☒ gcc -o Run.out a.o b.o
- ☐ gcc -Wall Run.out a.o b.o



Chọn các lệnh biên dịch để có được tập tin đối tượng, giả sử thư mục làm việc 2/2 đang tồn tại tập tin vidu.c

- ☐ gcc vidu.c
- ☐ gcc -o vidu.c
- ☒ gcc -c vidu.c
- ☒ gcc -Wall -c vidu.c

Với lời gọi "---a.out 123 abc 456" thì số lượng đối số là bao nhiêu? (--- là lệnh gọi thực thi) 2/2

- ☒ 4
- ☐ 5
- ☐ 3
- ☐ 2

Quá trình biên dịch và liên kết tách biệt được sử dụng khi nào? Chọn tất cả tình huống. 3/3

- ☒ Khi có thư viện liên kết động
- ☒ Khi có nhiều tập tin nguồn .c
- ☒ Khi cần che giấu một số mã nguồn thông qua tập tin đối tượng
- ☒ Khi có thư viện liên kết tĩnh
- ☐ Khi có một tập tin nguồn .c



Lệnh nào in các tiến trình dưới dạng cây để xem cho thuận tiện?

0/3

- ☐ ps --forest
- ☒ ps --tree
- ☐ ps -a
- ☐ ps --list

Lệnh "getppid();" trả về thông tin gì khi thực thi nó trong một terminal?

3/3

- ☒ id của bash đã gọi chạy
- ☐ id của tiến trình chứa câu lệnh
- ☐ id của tiến trình quản lý các tiến trình
- ☐ id của lệnh ppid()



Lệnh wait() ở dòng 14 diễn tiến ra sao?

0/2

```
01 #include <sys/types.h>
02 #include <stdio.h>
03 #include <unistd.h>
04 int value = 5;
05 int main()
06 {
07     pid_t pid;
08     pid = fork();
09     if (pid == 0) { /* child process */
10         value += 15;
11         return 0;
12     }
13     else if (pid > 0) { /* parent process */
14         wait(NULL);
15         printf("PARENT: value = %d",value); /* LINE A */
16         return 0;
17     }
18 }
```

- ☐ Lệnh trì hoãn cho đến khi hàm main() kết thúc
- ☐ Lệnh trì hoãn cho đến khi lệnh cuối cùng của tiến trình con kết thúc
- ☐ Lệnh chờ với tham số thời gian được truyền vào
- ☐ Lệnh chờ cho đến khi tiến trình con được sinh ra



Để bắt buộc tiến trình có pid=1234 dừng ngay lập tức, lệnh nào cần được sử dụng? 2/2

- ☒ kill -9 1234
- ☐ kill -2 1234
- ☐ kill force 1234
- ☐ kill -a 1234

Nếu tiến trình cha kết thúc trước tiến trình con, đó là tình huống gì và diễn tiến tiếp theo là gì? 3/3

- ☒ Tiến trình con trở thành mồ côi và bị buộc kết thúc ngay sau đó.
- ☐ Tiến trình cha không bao giờ kết thúc trước khi con nó hoàn tất.
- ☐ Tiến trình con trở thành xác sống và gây lỗi cho hệ thống
- ☐ Tiến trình con trở thành mồ côi và tiếp tục chạy.

Tiến trình con tiếp nhận những gì từ tiến trình cha? Chọn mọi đáp án đúng. 3/3

- ☒ Sao chép toàn bộ mã nguồn kể từ lời gọi fork()
- ☒ Tạo mới vùng nhớ và sao chép toàn bộ biến số
- ☒ Có bộ đếm chương trình riêng và chạy độc lập
- ☐ Chia sẻ và dùng chung các biến số toàn cục
- ☐ Chiếm hữu các biến số khai báo sau lời gọi fork()



Dòng số 15 sẽ in ra màn hình nội dung gì, và tại sao?

2/2

```
01 #include <sys/types.h>
02 #include <stdio.h>
03 #include <unistd.h>
04 int value = 5;
05 int main()
06 {
07     pid_t pid;
08     pid = fork();
09     if (pid == 0) { /* child process */
10         value += 15;
11         return 0;
12     }
13     else if (pid > 0) { /* parent process */
14         wait(NULL);
15         printf("PARENT: value = %d",value); /* LINE A */
16         return 0;
17     }
18 }
```

- ☐ PARENT: value = 20 bởi vì tiến trình con đã thêm 15 vào giá trị 5 ban đầu của biến value
- ☒ PARENT: value = 5 bởi vì biến số value của tiến trình cha không bị thay đổi
- ☐ PARENT: value = 5 bởi vì có thể lệnh cộng ở dòng 10 xảy ra sau lệnh in ở dòng 15
- ☐ PARENT: value = 0 vì sau khi sinh tiến trình con, biến số value được khởi tạo.



Lời gọi fork() được sử dụng trong trường hợp nào?

3/3

- ☒ Tạo ra thêm một tiến trình
- ☐ Tạo ra thêm hai tiểu trình
- ☐ Thực thi một tiến trình mới
- ☐ Kết thúc tiến trình hiện tại

Nếu tiến trình con kết thúc trước tiến trình cha mà cha không biết về việc kết thúc này, đó là tình huống gì và diễn tiến tiếp theo là gì?

- ☒ Tiến trình con trở thành xác sống và được HĐH quản lý.
- ☐ Tiến trình cha không bao giờ kết thúc trước khi con nó hoàn tất.
- ☐ Tiến trình con trở thành mồ côi và tiếp tục chạy.
- ☐ Tiến trình con trở thành xác sống và gây lỗi cho hệ thống.

LAB TEST - 04

12 of 14 points

Việc chia sẻ biến toàn cục giữa các tiểu trình như thế nào?

3/3

- ☒ Mọi tiểu trình đương nhiên đủ quyền truy cập biến toàn cục
- ☐ Các tiểu trình chia sẻ biến toàn cục thông qua cơ chế đường ống.
- ☐ Mỗi tiểu trình được sao chép một biến toàn cục riêng biệt
- ☐ Các tiểu trình chia sẻ biến toàn cục thông qua vùng nhớ chia sẻ.



Khi biên dịch chương trình có lời gọi `fork()`, các lựa chọn cần bổ sung là gì? 3/3

- ☐ gcc -lm
- ☐ gcc -lpthread
- ☐ gcc -fork
- ☒ gcc (không cần bổ sung)

Đối số truyền cho một tiểu trình chấp nhận những dạng nào sau đây? 0/2

- ☐ Chuỗi kí tự
- ☐ Kiểu con trỏ
- ☒ Cấu trúc tự định nghĩa
- ☐ Số thực/số nguyên

Ý nghĩa của tham số thứ ba trong lời gọi `pthread_create(& tid, & attr, runner, 3/3
argv[1]);` là gì?

- ☐ hàm mô tả công việc do hệ thống gán cho tiểu trình
- ☐ thuộc tính của tiểu trình được tạo ra và chạy
- ☒ hàm mô tả công việc giao cho tiểu trình
- ☐ con trỏ quản lý tiểu trình có được từ khai báo `void runner(param);`



Ý nghĩa của tham số đầu tiên trong lời gọi `pthread_create(& tid, & attr, runner, 3/3 argv[1]);` là gì?

- ☐ hàm mô tả công việc giao cho tiểu trình
- ☐ thuộc tính của tiểu trình được tạo ra và chạy
- ☐ con trỏ quản lý tiểu trình do hệ thống tự cung cấp
- ☒ con trỏ quản lý tiểu trình có được từ khai báo `pid_t tid;`

LAB TEST - 05

11 of 14 points

Đường ống phù hợp với bài toán nào?

3/3

- ☒ Gửi nhận thông điệp
- ☐ Đồng bộ cha - con
- ☐ Tiến trình độc lập
- ☐ Chia sẻ tính toán

Đường ống không tên được khởi tạo bằng những tham số nào?

3/3

- ☐ `makepipe -fd`
- ☐ `mmap(0, SIZE, PROT_WRITE, MAP_SHARED, shm_id, 0);`
- ☐ `((mknod(FIFO1, S_IFIFO | 0666, 0)`
- ☒ `int fd[2]; và pipe(fd);`



Sự khác nhau giữa đường ống có tên và đường ống không tên là gì?

0/3

- ☐ Đường ống không tên không cần chỉ định quyền đọc / ghi.
- ☐ Đường ống có tên không cần chỉ định quyền đọc / ghi.
- ☒ Đường ống không tên chỉ sử dụng cho tiến trình cha - con
- ☒ Đường ống có tên sử dụng cho bất kỳ tiến trình nào trong hệ thống.

Vùng nhớ chia sẻ được sử dụng như thế nào?

2/2

- ☐ Truy cập như một chuỗi kí tự.
- ☒ Truy cập như một mảng số nguyên
- ☐ Ghi và đọc bằng các lời gọi read / write.
- ☐ Truy cập như một tập tin.



Bắt cặp thứ tự khi GỬI thông điệp vào đường ống fd.

	Bước 1	Bước 2	Bước 3	Score
<code>open(fd[READ_END]);</code>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	.../0
<code>close(fd[READ_END]);</code>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	1/1
<code>write(fd[WRITE_END], write_msg, strlen(write_msg) + 1);</code>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	1/1
<code>open(fd[WRITE_END]);</code>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	.../0
<code>close(fd[WRITE_END]);</code>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	1/1
<code>read(fd[READ_END], read_msg, BUFFER_SIZE);</code>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	.../0

This form was created inside of Ton Duc Thang University.

Google Forms

