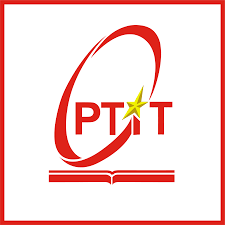
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**



**MÔN HỌC:** PHÂN TÍCH MÃ ĐỘC

**BÁO CÁO THỰC HÀNH BÀI 5**

**Giảng viên:** PGS.TS. Đỗ Xuân Chợ

**Sinh viên:** Hoàng Trung Kiên – B20DCAT098

Hà Nội – 5/2023

Mục lục

[I. Lý thuyết. 3](#_Toc148990501)

[1. Mục đích. 3](#_Toc148990502)

[2. Thực hành. 3](#_Toc148990503)

[***Nhiệm vụ 1: Thao tác với các biến môi trường*** 3](#_Toc148990504)

[*Nhiệm vụ 2: Kế thừa biến môi trường từ tiến trình cha* 4](#_Toc148990505)

[*Nhiệm vụ 3: Các biến môi trường và hàm execve()* 6](#_Toc148990506)

[*Nhiệm vụ 4: các biến môi  trường và hàm system()* 7](#_Toc148990507)

[*Nhiệm vụ 5: các biến môi  trường và các chương trình Set-UID* 8](#_Toc148990508)

[*Nhiệm vụ 6: hàm system() và các chương trình Set-UID* 9](#_Toc148990509)

[*Nhiệm vụ 7: biến môi  trường LD\_PRELOAD và các chương trình Set-UID* 11](#_Toc148990510)

[*Nhiệm vụ 8: Khả năng rò rỉ* 12](#_Toc148990511)

[II. Checkwork 13](#_Toc148990512)

# **I. Lý thuyết.**

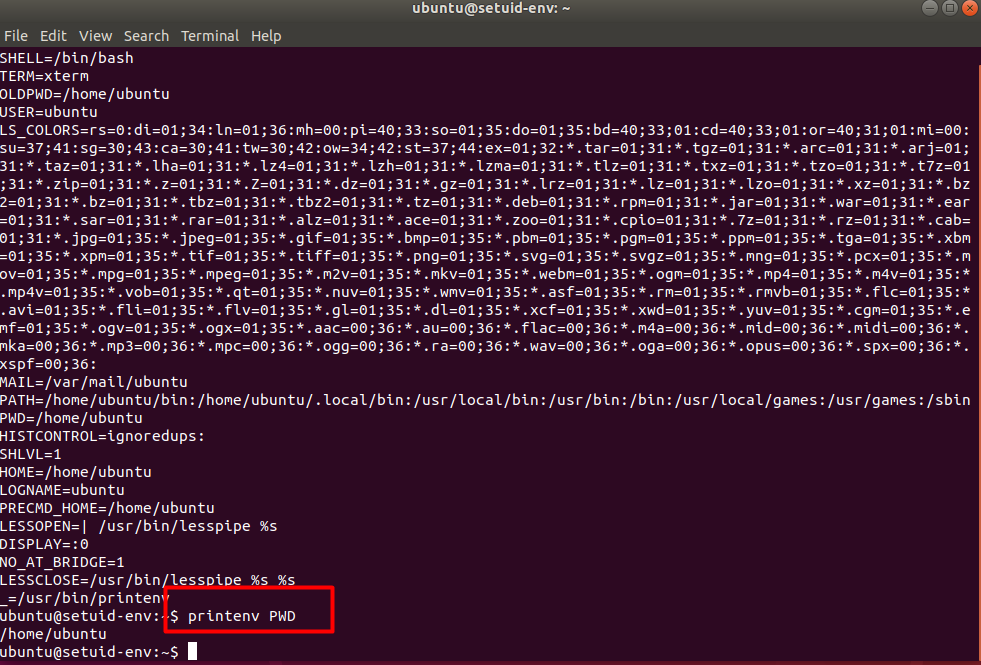
## 1. Mục đích.

Mục tiêu của bài thực hành này là giúp sinh viên hiểu được cách biến môi trường ảnh hưởng đến hành vi của các chương trình và hệ thống. Biến môi trường là một tập hợp các giá trị được gán tên và có tính động, có thể ảnh hưởng đến cách mà các tiến trình đang chạy sẽ hoạt động trên máy tính. Chúng được sử dụng bởi hầu hết các hệ điều hành, kể từ khi chúng được giới thiệu vào Unix vào năm 1979. Mặc dù biến môi trường ảnh hưởng đến hành vi của chương trình, cách chúng hoạt động thường không được hiểu rõ bởi nhiều lập trình viên. Kết quả là, nếu một chương trình sử dụng biến môi trường, nhưng nhà lập trình không biết rằng chúng được sử dụng, chương trình có thể có lỗ hổng bảo mật. Trong bài thực hành này, sinh viên sẽ hiểu cách biến môi trường hoạt động, cách chúng được truyền từ tiến trình cha đến tiến trình con và cách chúng ảnh hưởng đến hành vi của hệ thống/ chương trình. Chúng tôi đặc biệt quan tâm đến cách biến môi trường ảnh hưởng đến hành vi của các chương trình Set-UID (thường được sử dụng bởi các chương trình đặc quyền).

2. Thực hành.

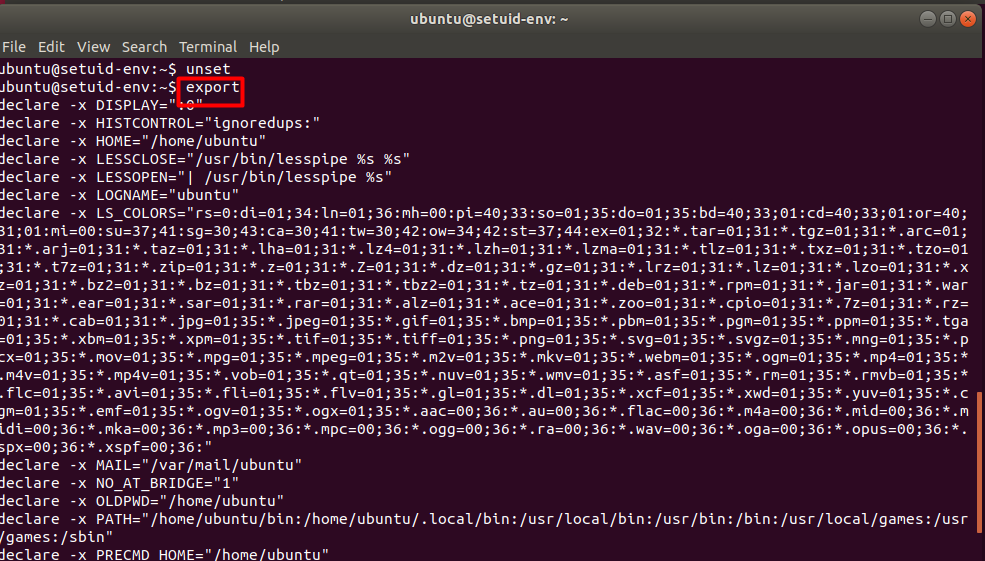
### ***Nhiệm vụ 1: Thao tác với các biến môi trường***

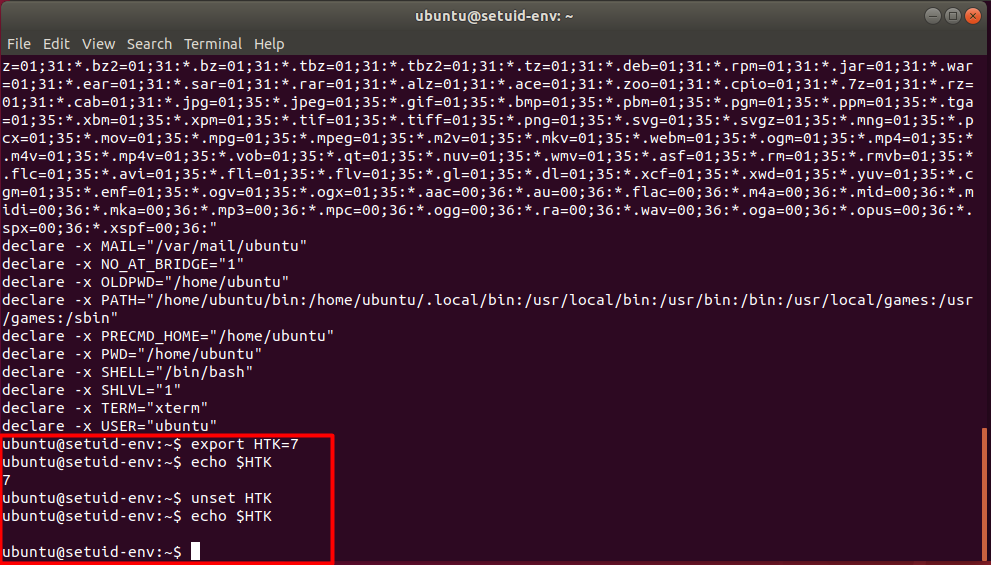
Sử dụng lệnh printenv hoặc env để in ra danh sách các biến môi trường.





Sử dụng lệnh export và unset để thiết lập hoặc hủy bỏ các biến môi trường



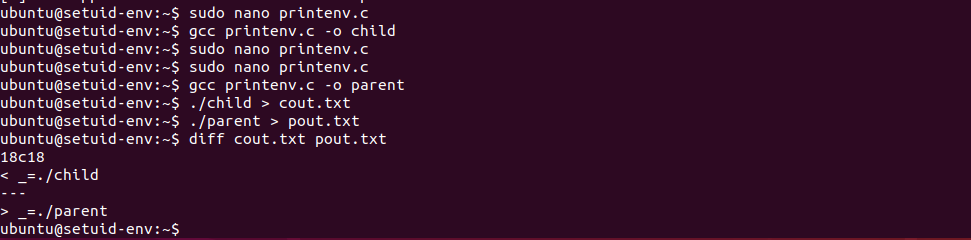


### Nhiệm vụ 2: Kế thừa biến môi trường từ tiến trình cha

**Bước 1.** Thực hiện biên dịch và chạy chương trình printenv.c

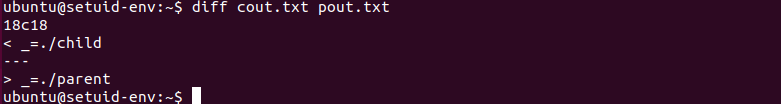


**Bước 2.** Tiếp theo, hãy chú thích câu lệnh printenv() trong trường hợp tiến trình con và bỏ chú thích câu lệnh printenv() trong trường hợp tiến trình cha. Biên dịch và chạy mã nguồn, và mô tả lại những gì quan sát được. Lưu đầu ra vào một tệp khác.





Bước 3. So sánh sự khác biệt giữa hai tệp bằng lệnh diff.

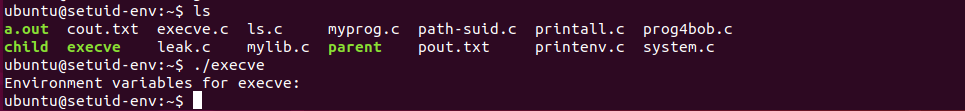


### Nhiệm vụ 3: Các biến môi trường và hàm execve()

**Bước 1.** Thực hiện biên dịch và chạy chương trình execve.c



**Bước 2.** Thay đổi đoạn mã execve("/usr/bin/env", argv, environ);



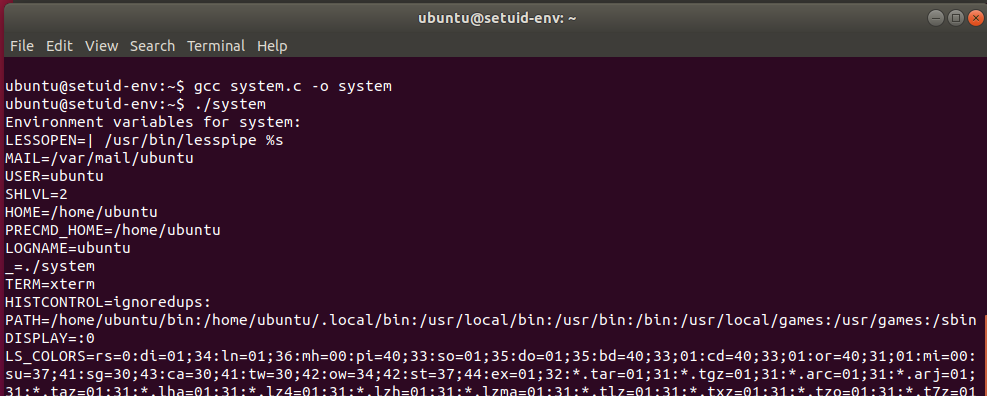


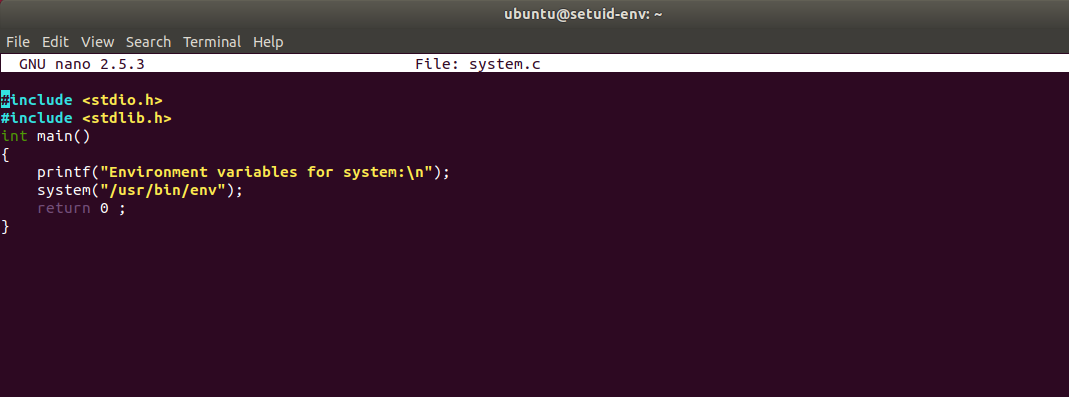
**Bước 3.** Cuối cùng rút ra kết luận về cách mà chương trình mới lấy các biến môi trường của nó.

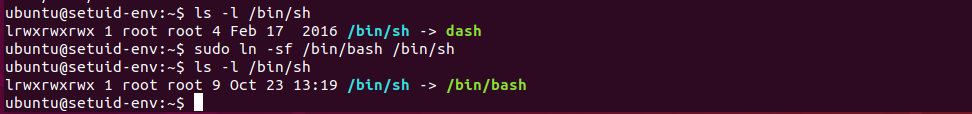


### Nhiệm vụ 4: các biến môi  trường và hàm system()

Thực hiện biên dịch và chạy chương trình system.c



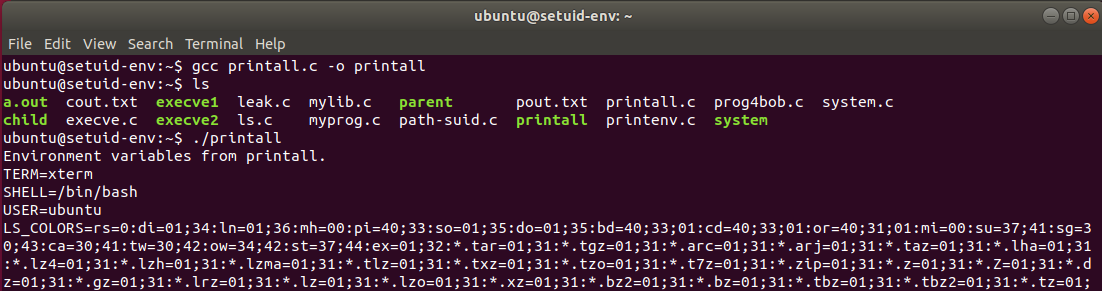


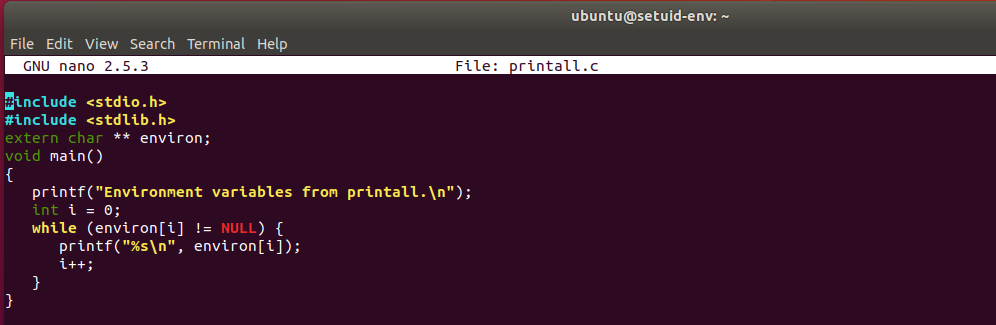


Khi hàm hệ thống thực thi, nó không trực tiếp thực thi mà thay vào đó nó gọi shell và shell thực thi lệnh. Shell gọi nội bộ hệ điều hành và các biến môi trường của quá trình gọi được thông qua tới shell và shell sẽ chuyển nó tới

### Nhiệm vụ 5: các biến môi  trường và các chương trình Set-UID

**Bước 1.** Sinh viên cần sử dụng chương trình printall.c

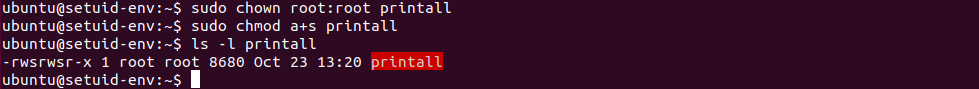




Bước 2. Thực hiện biên dịch chương trình printall.c và thay đổi chủ sở hữu của chương trình sang root và đặt Set-UID cho nó.

sudo chown root:root a.out

sudo chmod a+s a.out

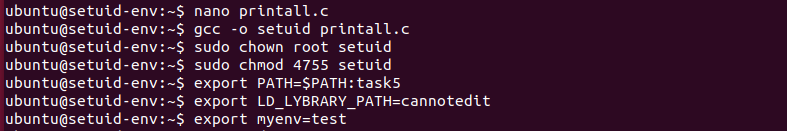


Trong Bash shell, sinh viên có thể sử dụng lệnh export để đặt các biến môi trường sau (chúng có thể đã tồn tại):

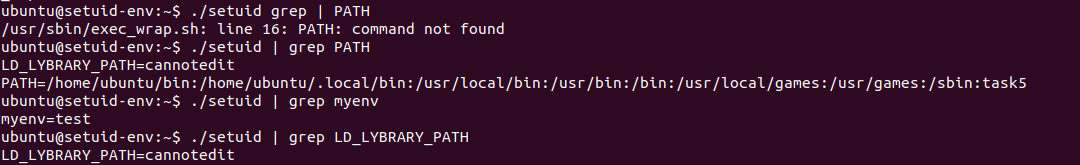
PATH

LD LIBRARY PATH

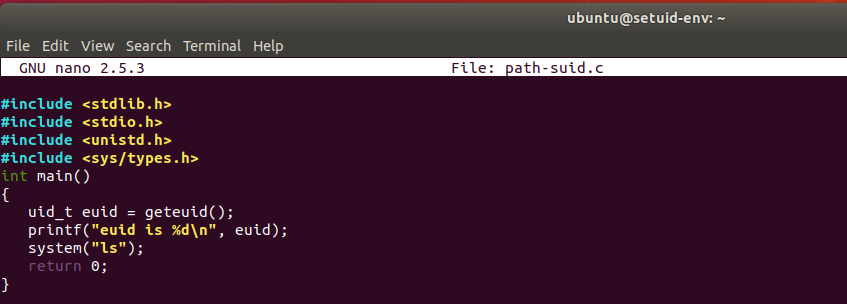
ANY NAME

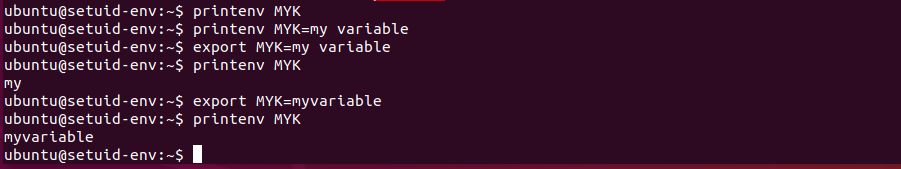


Hãy kiểm tra xem tất cả các biến môi trường sinh viên đặt trong tiến trình cha có được truyền đến tiến trình con của Set-UID hay không.

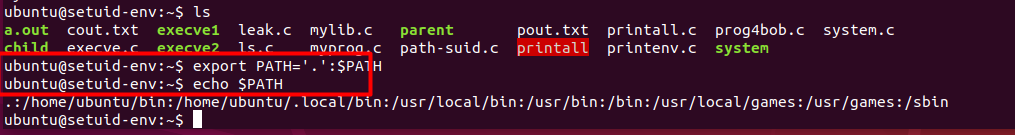


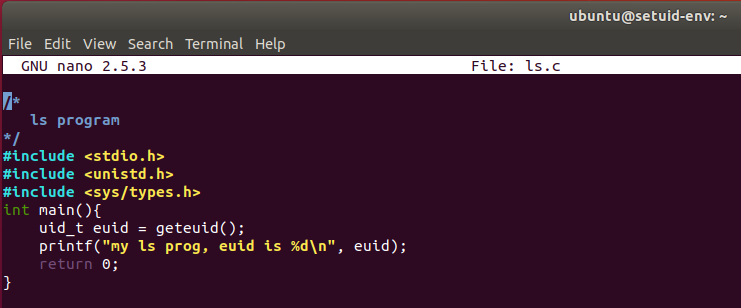
### Nhiệm vụ 6: hàm system() và các chương trình Set-UID

Chương trình setuid path-suid.c được liệt kê dưới đây được thiết kế để thực thi lệnh /bin/ls

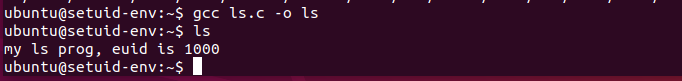


Thay đổi đường dẫn export PATH=’.’:$PATH



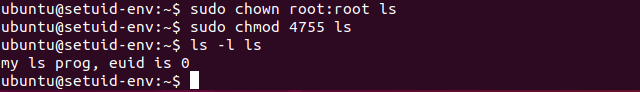


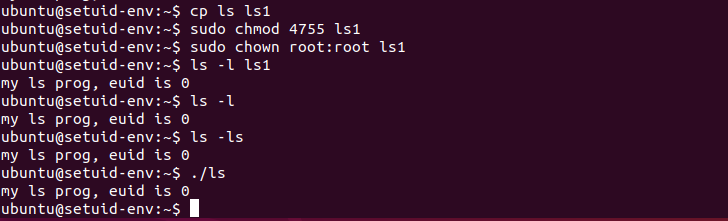
Chạy và biên dịch ls.c ta thấy được euid là 1000



thay đổi chủ sở hữu của chương trình sang root và đặt Set-UID cho nó.

Sau đó chạy lại file ls và thấy euid là 0



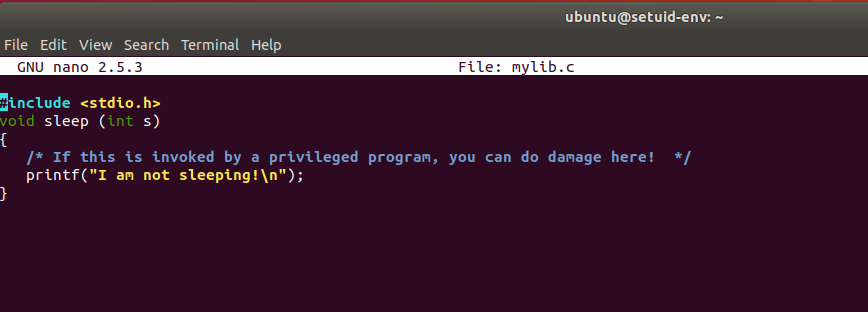


Chúng ta đã thay đổi quyền sở hữu thành root, quan sát biến môi trường đường dẫn tìm kiếm lệnh ls trong thư mục hiện tại trước tiên vì nó được chỉ định. Khi phát hiện thấy ls tồn tại, nó sẽ chạy chương trình đó thay vì lệnh shell ls, điều này chứng minh cho chúng ta thấy chương trình set uid có thể chạy các tệp độc hại có quyền root nếu biến đường dẫn bị thay đổi.

### Nhiệm vụ 7: biến môi  trường LD\_PRELOAD và các chương trình Set-UID

Bước 1. Đầu tiên, chúng ta sẽ xem xét cách các biến môi trường này ảnh hưởng đến hành vi của trình tải/liên kết động khi chạy một chương trình bình thường. Thực hiện theo các bước sau:

Đầu tiên, xây dựng một thư viện liên kết động. Đoạn mã chương trình sau đây được lưu trong tệp mylib.c.



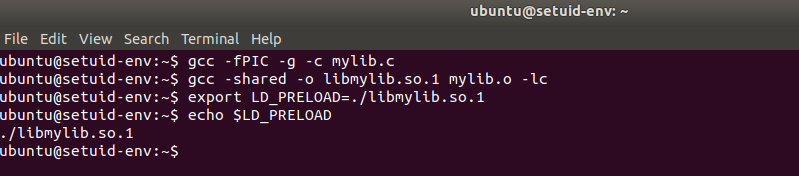
Chúng ta có thể biên dịch chương trình mylib.c bằng các lệnh sau

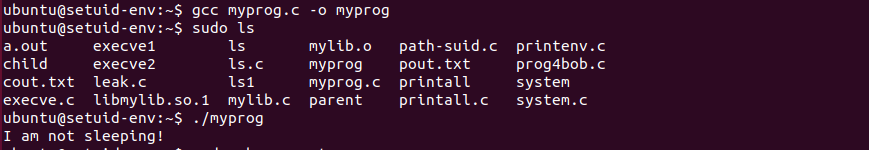
% gcc -fPIC -g -c mylib.c

% gcc -shared -o libmylib.so.1.0.1 mylib.o –lc

Tiếp theo cần thiết lập biến môi trường LD\_PRELOAD

% export LD\_PRELOAD=./libmylib.so.1.0.1





Cuối cùng, biên dịch chương trình myprog.c trong cùng thư mục với thư viện liên kết động libmylib.so.1.0.1 như sau:

**Bước 2.** chạy myprog dưới các điều kiện sau đây, và quan sát điều gì xảy ra.

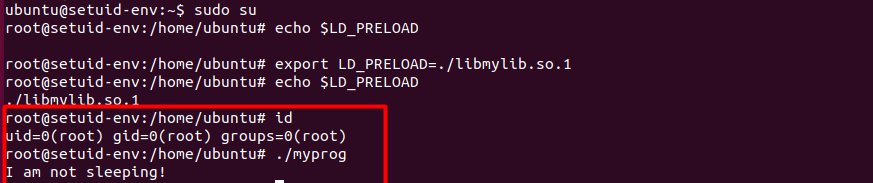
Biến myprog thành một chương trình bình thường và chạy nó dưới dạng một người dùng bình thường.

Biến myprog thành một chương trình Set-UID root và chạy nó dưới dạng một người dùng bình thường.

Trở thành người dùng root với lệnh sudo su, xuất biến môi trường LD\_PRELOAD và chạy lại chương trình myprog.

Biến myprog thành một chương trình Set-UID user1 (tức là chủ sở hữu là user1, là một tài khoản người dùng khác), xuất biến môi trường LD\_PRELOAD lại dưới tên người dùng user1 và chạy nó.



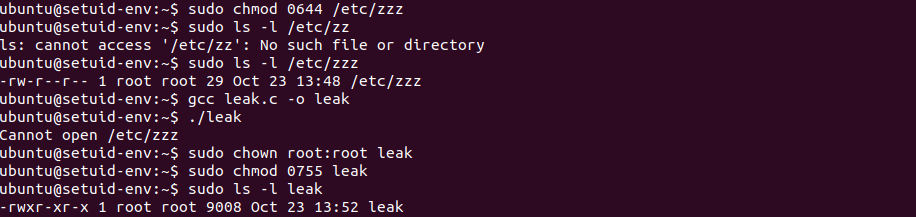


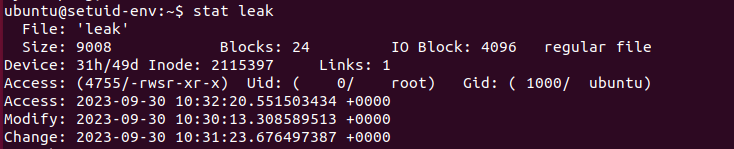
### Nhiệm vụ 8: Khả năng rò rỉ



Chạy và biên dịch file leak.c, sau đó thay đổi chủ sở hữu của nó thành root và biến nó thành một chương trình Set-UID.







Ta mở tệp zzz với quyền root trong /etc, nó sẽ hiện ra dòng chữ “important stuff, not really. 

# II. Checkwork

