

**LAB 3**

**SỬ DỤNG SHELL SCRIPTING, QUẢN LÝ TIẾN TRÌNH, TẬP TIN NHẬT KÝ HỆ THỐNG**

|  |
| --- |
| Họ tên và MSSV: Bùi Trần Ngọc Ly – B1908338  Nhóm học phần: 07 |

*- Các sinh viên bị phát hiện sao chép bài của nhau sẽ nhận 0đ cho tất cả bài thực hành của môn này.*

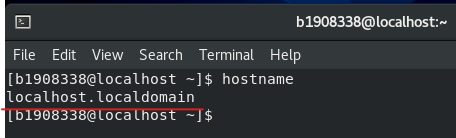
*- Bài nộp phải ở dạng PDF, hình minh họa phải rõ ràng chi tiết.*

1. **Cài đặt CentOS**

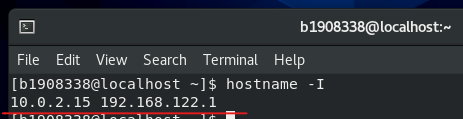
Thực hiện cài đặt CentOS 8 vào máy tính cá nhân (hoặc máy ảo) của bạn (Nếu chưa cài, không cần chụp hình minh hoạ)

1. **Quản trị với shell scripting**
   1. Thực hiện các lệnh bên dưới và cho biết ý nghĩa của chúng (chụp hình minh hoạ):

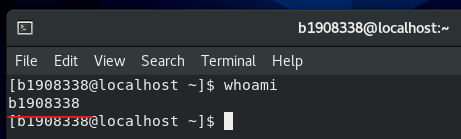
* Hostname: là chương trình được sử dụng để thiết lập hoặc hiển thị tên máy chủ, tên miền hoặc tên nút hiện tại của hệ thống **Linux**. Những tên này được sử dụng bởi các phân mềm network để xác định trên hệ thống.



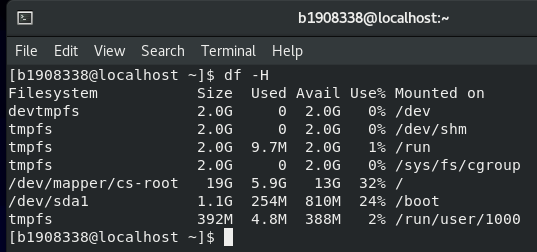
* hostname –I: dùng để hiển thị tất cả đị chỉ IP cho tài khoản host



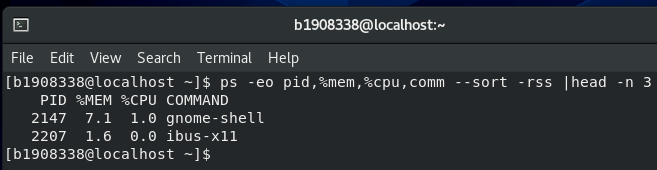
* whoami: bắt nguồn từ cụm từ tiếng Anh “Who am I”(tôi là ai), có chức năng xuất ra tên tài khoản người dùng đang đăng nhập. Lệnh có chức năng tương tự với lệnh Unix í –un,… Ví dụ nếu người dùng đăng nhập tài khoản John và su và root, whoami sẽ hiển thị root còn echo $USER xuất ra John



* df –H: kích thước in ở định dạng con người có thể đọc được và sử dụng công suất 1000 không phải 1024

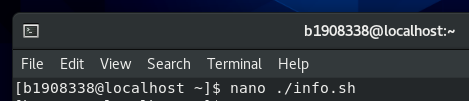


* ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3: dùng để hiển thị tiếng trình đầu tiên của việc sử dụng tài nguyên của hệ thống (mem, cpu) và sắp xép theo chiều thử tự giảm dần

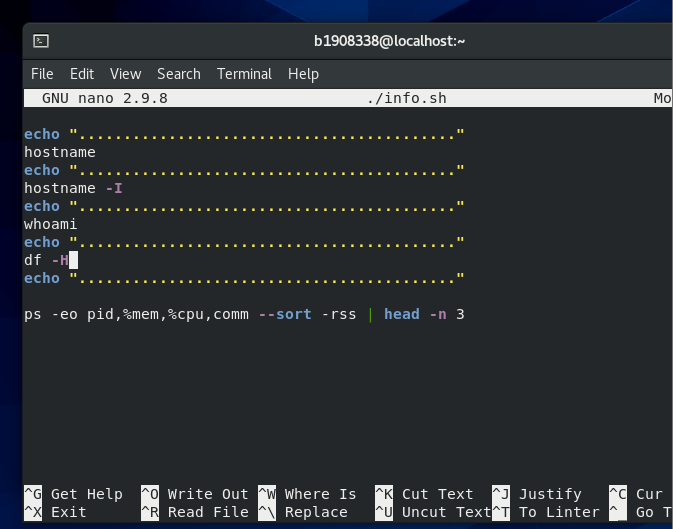


(KHÔNG CÓ KHOẢNG TRẮNG SAU DẤU PHẨY)

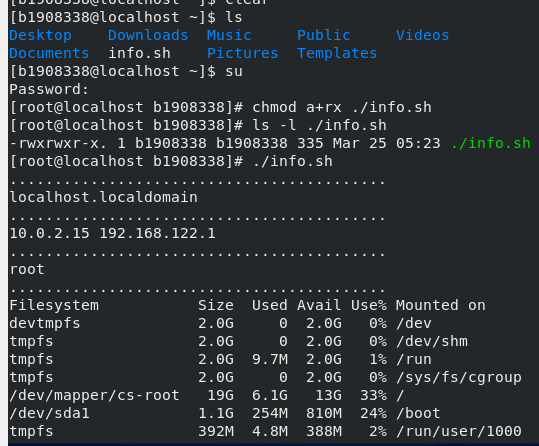
* 1. Viết shell script có tên *info.sh* thực hiện tất cả các lệnh ở 2.1 (chụp hình minh hoạ).
* **Viét lệnh shell scrpit**

****

* **Thực thi lệnh và ghi nội dung**

****

* **Cấp quyền và chạy kết quả:**

****

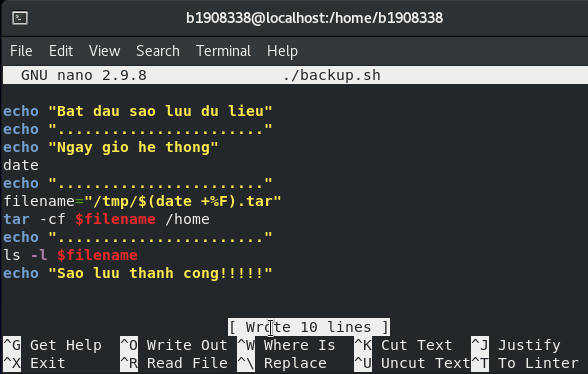
* 1. Viết shell script có tên *backup.sh* thực hiện:
* In ra ngày giờ hiện tại
* Nén toàn bộ thư mục **/home** thành tập tin **/tmp/<YYYY-MM-DD>.tar** (YYYY-MM-DD là ngày hiện tại, ví dụ: 2020-04-22.tar)
* In thông tin đầy đủ của tập tin /tmpn/<YYYY-MM-DD>.tar
* In thông ra thông báo “Sao lưu thành công!!!!”

Thực thi backup.sh để kiểm tra (chụp hình minh hoạ).

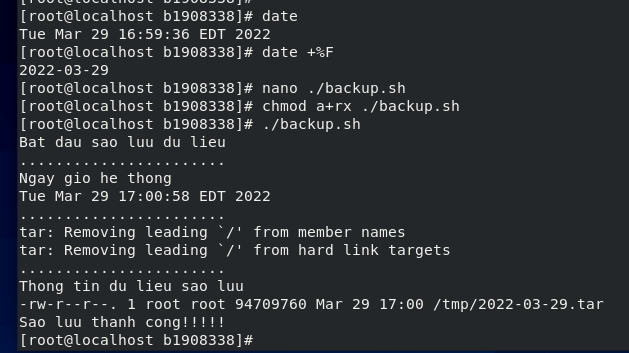
* **Viết lệnh:**



* **Ghi nội dung:**



* **Chạy và cho kết quả**



* 1. Bổ sung dòng các lệnh cần thiết vào shell script *safe\_rm.sh* bên dưới để thực hiện công việc sau:

- Nhận 01 tham số từ dòng lệnh. Thông báo lỗi và thoát khỏi chương trình nếu không có tham số hoặc nhiều hơn 01 tham số.

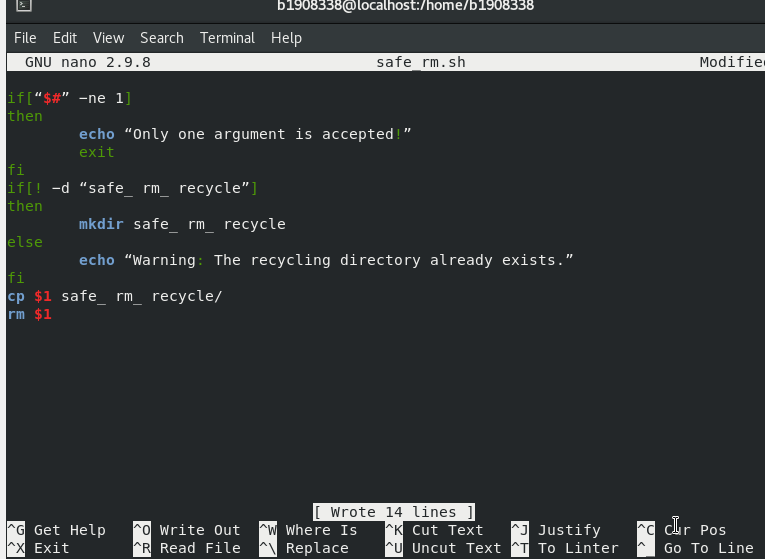
- Tạo thư mục với tên *“safe\_rm\_recycle”* nếu chưa có. Sao chép tập tin với tên là tham số 01 vào thư mục *“safe\_rm\_recycle”* vừa tạo. Sau đó xóa tập tin.

- Tạo tập tin có tên *mydata.txt,* thực thi *safe\_rm.sh* để xóa tập tin *mydata.txt*  (chụp hình minh hoạ).

*safe\_rm.sh*

|  |
| --- |
| if [“$#” −ne 1]  then  echo “Only one argument is accepted!”  // Thêm 1 dòng lệnh ở đây  fi  if [! −d “safe\_ rm\_ recycle”]  then  // Thêm 1 dòng lệnh ở đây  else  echo “Warning: The recycling directory already exists.”  fi  // Thêm 1 dòng lệnh ở đây  // Thêm 1 dòng lệnh ở đây |

* **File safe\_rm.sh**

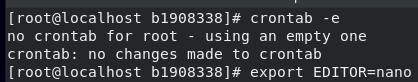
** ­­­­­­**

1. **Lên lịch công việc định kỳ với cron**

Cron là một tiện ích trong Linux cho phép máy tính thiết lập thời gian biểu để thực hiện công việc một cách định kỳ. Một crontab file chứa danh sách các lệnh sẽ được thực thi, kèm theo thông tin về thời điểm lặp lại việc thực thi. Để hiệu chỉnh file crontab với trình soạn thảo nano, ta dùng các lệnh sau:

export EDITOR=nano

crontab –e

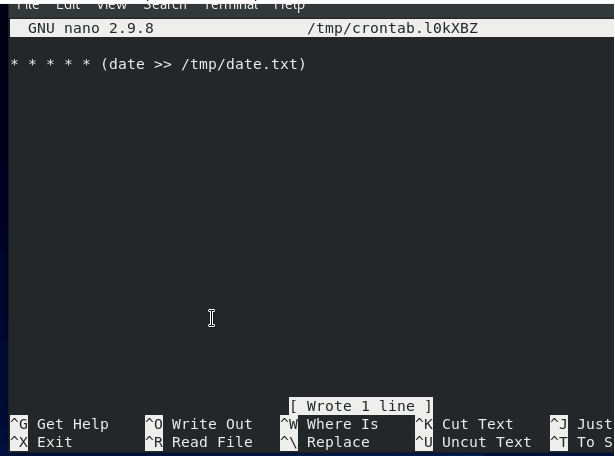


Cho biết cú pháp để thực hiện các yêu cầu sau từ crontab file:

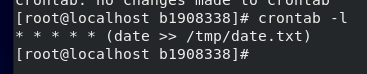
* 1. Chạy lệnh **date** mỗi phút một lần, sau đó ghi kết quả vào cuối tập tin **/tmp/date.txt** (chụp hình minh hoạ)
* **Đặt biến môi trường:**

****

* **Chạy lệnh date mỗi phút 1 lần**

****

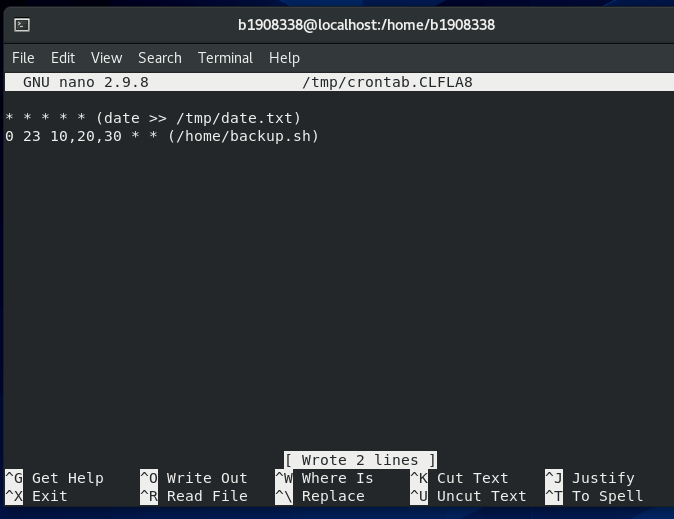
* **Thực hiện lệnh crontab –l để thấy các công việc đã lên lịch**

****

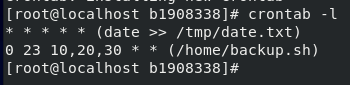
* **In tập tin ngày giờ hệ thống, cứ mỗi phút tập tin được thêm 1 dòng**

****

* 1. Thực thi *backup.sh* ở 2.3 vào 23:00 giờ ngày 10, 20 và 30 hàng tháng (chụp hình minh hoạ).
* **Mở crontab –e:** thực thi backup.sh 23:00 giờ ngày 10, 20 và 30 hàng tháng



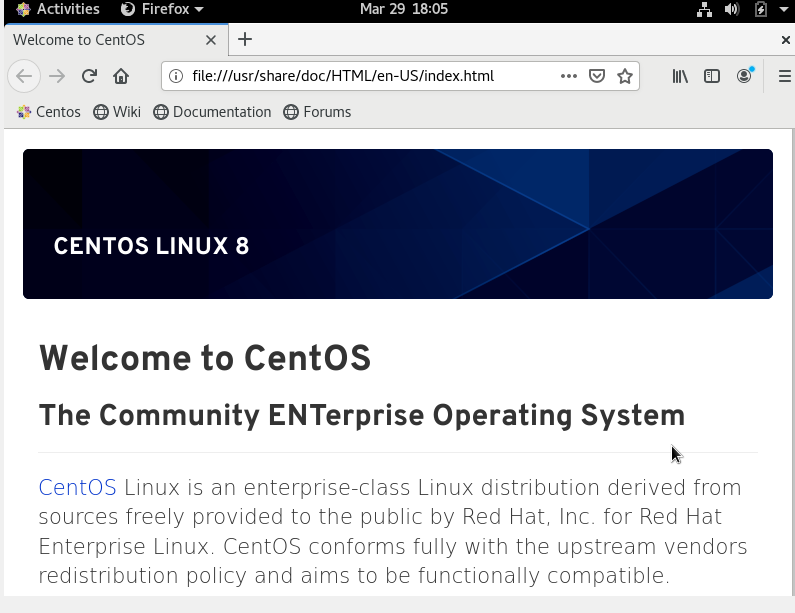
* Kết quả:



1. **Thao tác với tiến trình**

Tìm hiểu và thực hiện các yêu cầu sau:

* 1. Mở trình duyệt Firefox. Sau đó dùng lệnh **ps -aux** hoặc **pgrep** tìm PID của firefox. Tiếp theo, dùng lệnh **kill** để tạm dừng tiến trình firefox (chụp hình minh hoạ). Điều gì xảy ra khi bạn dịch chuyển cửa sổ firefox hoặc nhấn chọn menu của nó ngay lúc này (chụp hình minh hoạ)?
* **Mở Firefox:**

****

* **Tìm mã số tiến trình:** mã số của tiến trình là 7161

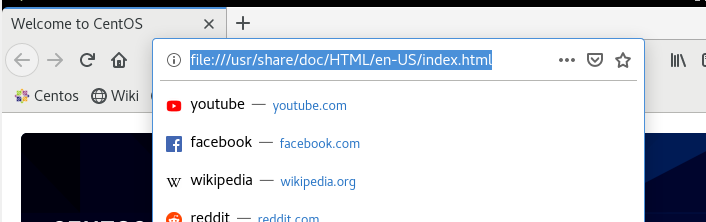
****

* **Tạm dừng tiến trình với lệnh kill tham số -s lệnh cần gửi và mã số tiến trình**

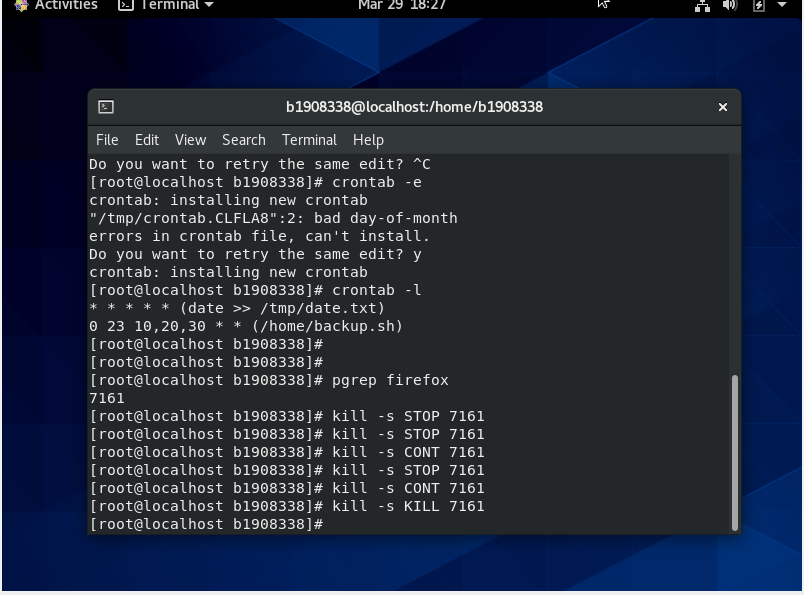
****

* 1. Dùng lệnh **kill** để phục hồi trạng thái trước đó của firefox và quan sát kết quả (chụp hình minh hoạ).
* **Tiếp tục tiến trình gửi tín hiệu COUNT**

****

****

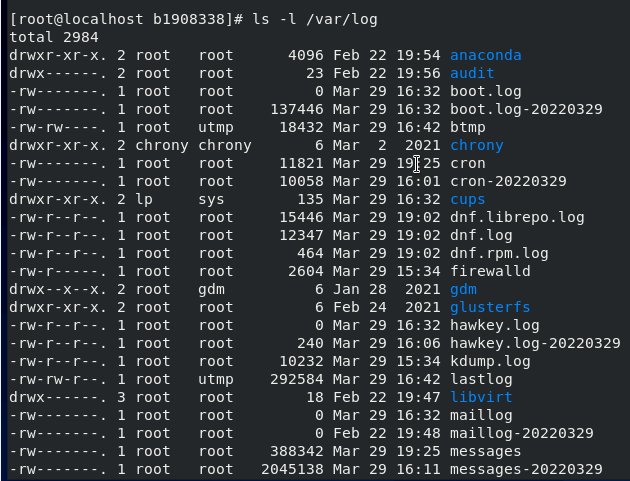
* 1. Dùng lệnh **kill** để hủy tiến trình firefox (chụp hình minh hoạ).
* **Tiến trình đã bị hủy**

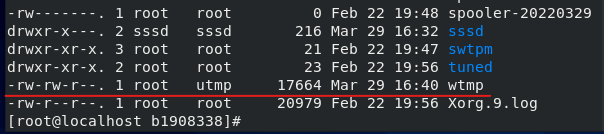
****

1. **Tập tin log**

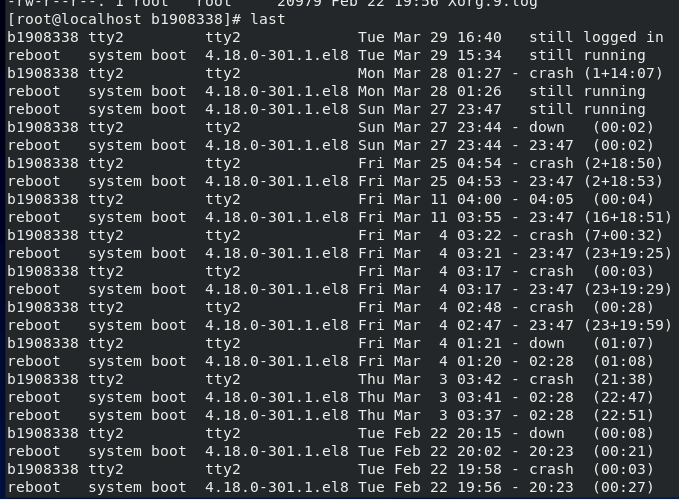
Tìm hiểu và thực hiện các yêu cầu sau:

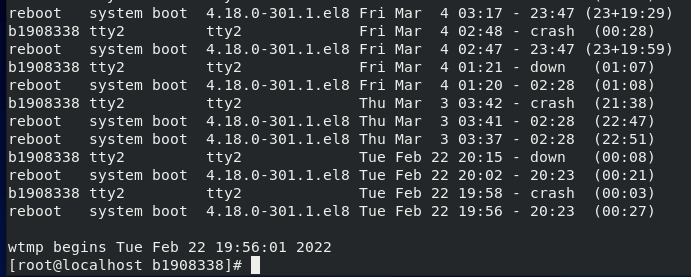
* 1. Tìm thông tin về người dùng, thời gian của lần đăng nhập sau cùng vào hệ thống (chụp hình minh hoạ).
* **Xem danh sách các file log trong folder ls -l / var / log**

****

****

* **Đọc tập tin sử dụng lệnh last hiện tất cả các lần đăng nhập của người dùng, do tập tin dưới dạng nhị phân**

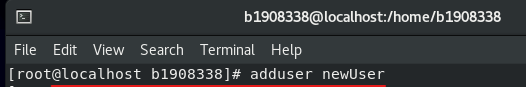
****

****

* **Hiển thị lần đăng nhập lần cuối cùng sử dụng lệnh head với tham số -n =1: in người dùng cuối cùng đăng nhập vào hệ thống**

****

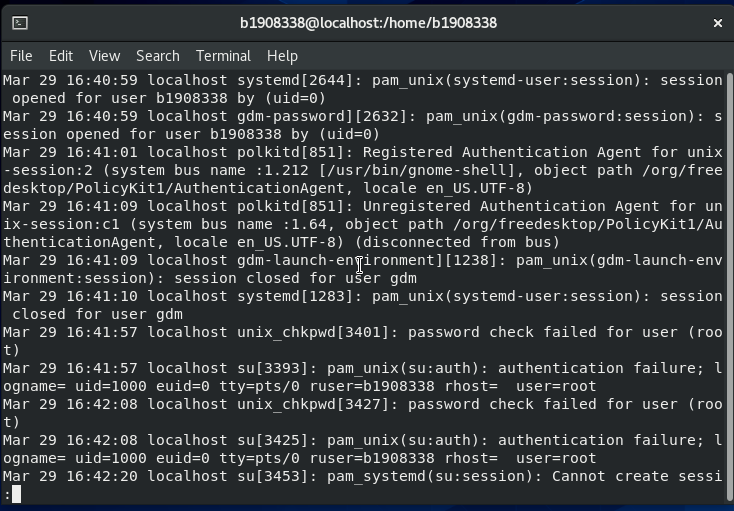
* 1. Tạo một người dùng mới.
* **Tạo người dùng mới bằng lệnh adduser**

****

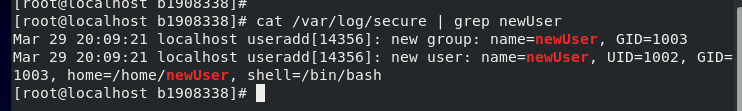
Người dùng mới có tên là **newUser**

* Trong tập tin log có tập tin chứa các sự kiện liên quan an ninh hệ thống **(secure)**

****

****

* 1. Tìm thời gian người dùng ở 5.2 được tạo ra (chụp hình minh hoạ).

****

--- Hết ---