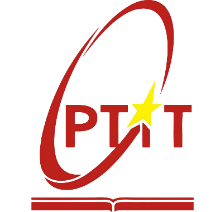
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-------🙞🙜🕮🙞🙜-------



**Báo cáo bài thực hành**

**Quản lý tiến trình**

Sinh viên thực hiện:

B20DCAT59 Nguyễn Cảnh HIếu

Giảng viên hướng dẫn: TS.Phạm Hoàng Duy

MỤC LỤC

[MỤC LỤC 1](#_Toc186690162)

[DANH MỤC CÁC HÌNH VẼ 2](#_Toc186690163)

[DANH MỤC CÁC BẢNG BIỂU 2](#_Toc186690164)

[1.1 Giới thiệu chung về bài thực hành 3](#_Toc186690165)

[1.2 Nội dung và hướng dẫn bài thực hành 3](#_Toc186690166)

[1.2.1 Mục đích 3](#_Toc186690167)

[1.2.2 Yêu cầu đối với sinh viên 3](#_Toc186690168)

[1.2.3 Nội dung thực hành 3](#_Toc186690169)

[1.3 Phân tích yêu cầu bài thực hành 4](#_Toc186690170)

[1.4 Thiết kế bài thực hành 4](#_Toc186690171)

[1.5 Cài đặt và cấu hình các máy ảo 5](#_Toc186690172)

[1.6 Tích hợp và triển khai 7](#_Toc186690173)

[***1.6.1 Docker Hub*** 7](#_Toc186690174)

[***1.6.2 Github*** 7](#_Toc186690175)

[1.7 Thử nghiệm và đánh giá 7](#_Toc186690176)

[TÀI LIỆU THAM KHẢO 11](#_Toc186690177)

DANH MỤC CÁC HÌNH VẼ

[Hình 1 Giao diện Labedit của bài lab 6](#_Toc186690178)

[Hình 2 Cài đặt phần Result 6](#_Toc186690179)

[Hình 3 Dockerfiles của máy resources 6](#_Toc186690180)

[Hình 4 Treataslocal của resources 7](#_Toc186690181)

[Hình 5 Đẩy các vùng chứa lên dockerhub 7](#_Toc186690182)

[Hình 6 Đã đẩy lên docker 7](#_Toc186690183)

[Hình 7 IP của máy resources 8](#_Toc186690184)

[Hình 8 Quan sát tổng quan các thông tin về tiến trình bằng top 8](#_Toc186690185)

[Hình 9 Nội dung file progress.sh 9](#_Toc186690186)

[Hình 10 Cấp quyền và chạy progress.sh 9](#_Toc186690187)

[Hình 11 Checkwork nhiệm vụ 1 10](#_Toc186690188)

[Hình 12 Thông tin về các tiến trình mới 10](#_Toc186690189)

[Hình 13 Checkwork nhiệm vụ 2 10](#_Toc186690190)

[Hình 14 Kết thúc tiến trình 11](#_Toc186690191)

[Hình 15 Checkwork nhiệm vụ 3 11](#_Toc186690192)

DANH MỤC CÁC BẢNG BIỂU

[Bảng 1. Bảng Result 5](#_Toc186690193)

* 1. Giới thiệu chung về bài thực hành

Bài thực hành "Quản lý tiến trình" được thiết kế nhằm giúp sinh viên hiểu rõ hơn về khái niệm cơ bản của tiến trình và quản lý các tiến trình trong máy tính giúp phát hiện các tiến trình lạ không cần thiết và có thể kết thúc để tăng hiệu quả cpu . Đây là một bước khởi đầu quan trọng để sinh viên làm quen với các kỹ thuật liên quan đến an ninh mạng. Trong bài thực hành này, sinh viên sẽ tìm hiểu cách sử dụng các lênh như top, ps, free -h để xem tổng quan về hệ thống và các thông tin về tiến trình. Đây là một kỹ năng cơ bản nhưng vô cùng quan trọng bài thực hành này không chỉ giúp sinh viên nắm vững lý thuyết mà còn cung cấp cơ hội để áp dụng kiến thức vào thực tế, từ đó nâng cao kỹ năng và sự tự tin khi làm việc với hệ điều hành Linux.

* 1. Nội dung và hướng dẫn bài thực hành
     1. Mục đích

Bài thực hành này nhằm giúp sinh viên nắm vững các lệnh quản lý tiến trình trong Linux, như ps, top, kill. Sinh viên sẽ thực hành tạo và quản lý tiến trình ở chế độ nền và mặt trước. Qua đó, sinh viên sẽ phát triển kỹ năng giải quyết vấn đề, làm quen với môi trường ảo hóa và áp dụng kiến thức vào thực tế, nâng cao sự tự tin khi làm việc với hệ điều hành Linux.

* + 1. Yêu cầu đối với sinh viên

Có kiến thức cơ bản về hệ điều hành Linux, khái niện về tiến trình trong hệ điều hành

* + 1. Nội dung thực hành
* Khởi động bài lab:

Vào terminal, gõ:

*Labtainer occ\_ac\_progress\_n3\_hieu\_3 -r*

*(chú ý: sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)*

Sau khi khởi động xong terminal **resources** ảo sẽ xuất hiện

Trên terminal **resources** thực hiện quan sát các tiến trình theo thời gian thực. thực hiện kết thúc các tiến trình cụ thể

* Kết thúc bài lab:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*stoplab occ\_ac\_progress\_n3\_hieu\_3 -r*

Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

* Khởi động lại bài lab:

Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

*startlab occ\_ac\_progress\_n3\_hieu\_3 -r*

* 1. Phân tích yêu cầu bài thực hành

Bài thực hành gồm một máy ảo **resources**. Để hoàn thành bài thực hành sinh viên cần thực hiện quan sát, phân tích các tiến trình theo thời gian thực bằng “top” viết 1 script để tạo ra các tiến trình mới sau đó quan sát hiệu suất cpu cuối cùng cần thực hiện kết thúc các tiến trình vừa tạo trên.

* 1. Thiết kế bài thực hành

Trên môi trường máy ảo Ubuntu được cung cấp, sử dụng docker để tạo ra một container: container mang tên “**resources**” là một máy tính ảo chạy hệ điều hành Linux đóng vai trò chính trong bài thực hành.

Cấu hình docker gồm có:

* config: lưu cấu hình hoạt động của hệ thống
* dockerfiles: mô tả cấu hình của container:
* Các nhiệm vụ cần phải thực hiện để thực hành thành công:
* Trên máy resources thực hiện lệnh top để xem tổng quan về các thông tin của tiến trình
* Thực hiện viết mã script để tạo ra tiến trình mới và thực hiện chạy nền các tiến trình
* Thực hiện lệnh top để xem cá tiến trình mới vừa được tạo theo dõi hiệu suất cpu, sau đó dùng lệnh kill để kết thúc các tiến vừa tạo

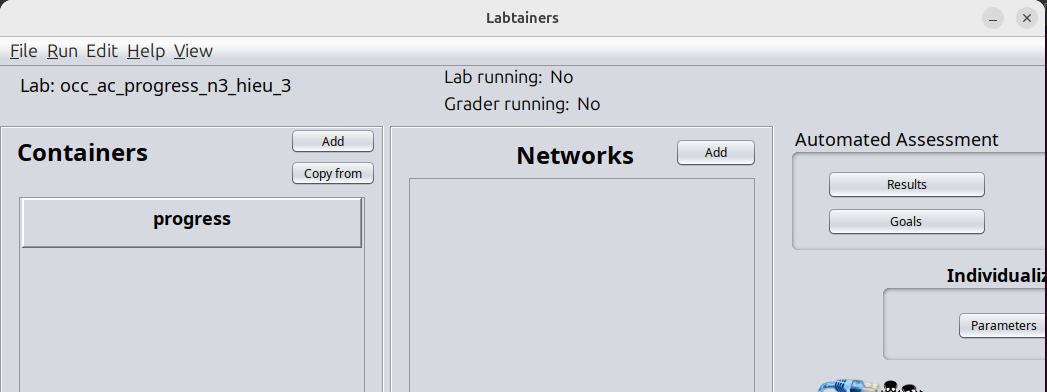
Kết thúc bài lab và đóng gói kết quả.

Để đánh giá được sinh viên đã hoàn thành bài thực hành hay chưa, cần chia bài thực hành thành các nhiệm vụ nhỏ, mỗi nhiệm vụ cần phải chỉ rõ kết quả để có thể dựa vào đó đánh giá, chấm điểm. Do vậy, trong bài thực hành này hệ thống cần ghi nhận các thao tác, sự kiện được mô tả và cấu hình như bảng 1:

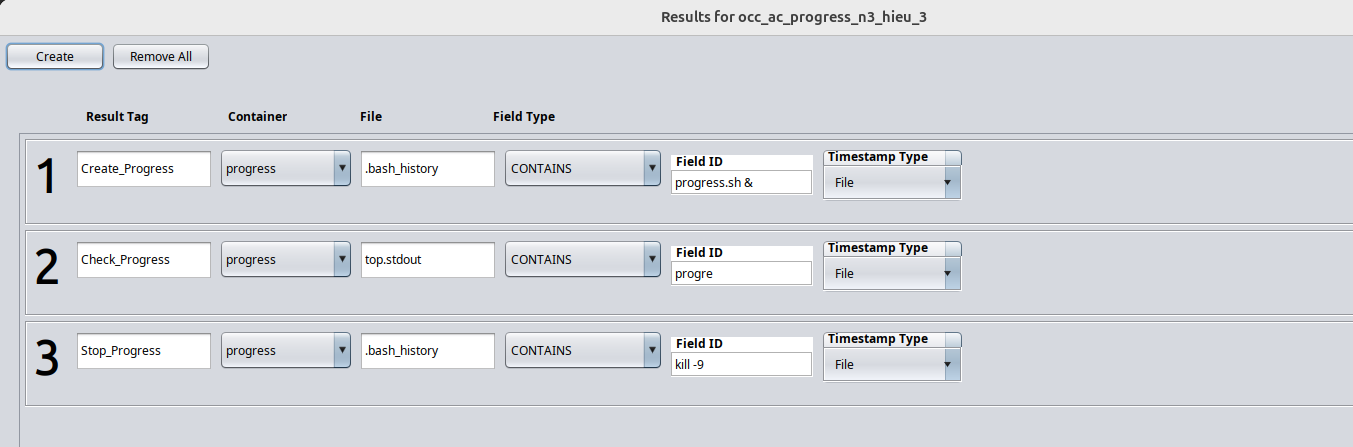
1. Bảng Result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Result Tag | Container | File | Field Type | Field ID | Timestamp Type |
| Create\_Progress | resources | .bash\_history | CONTAINS | progress & | File |
| Check\_Progress | resources | top.stdout | CONTAINS | progre | File |
| Stop\_Progress | resources | .bash\_history | CONTAINS | kill -9 | File |

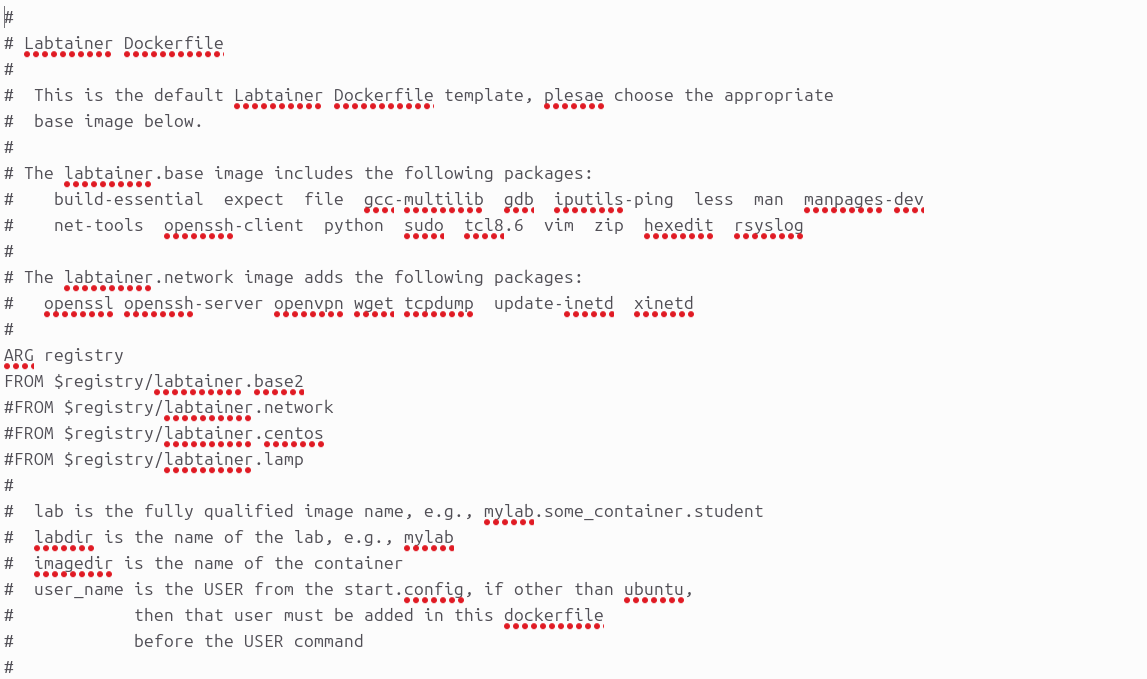
* Create\_Progress: Viết mã script để tạo và chạy tiến trình mới
* Check\_Progress: kiểm tra, xem xét tổng quan các thông tin về tiến trình
* Stop\_Progress: kết thúc các tiến trình vừa được tạo ở trên
  1. Cài đặt và cấu hình các máy ảo



1. Giao diện Labedit của bài lab



1. Cài đặt phần Result



1. Dockerfiles của máy resources



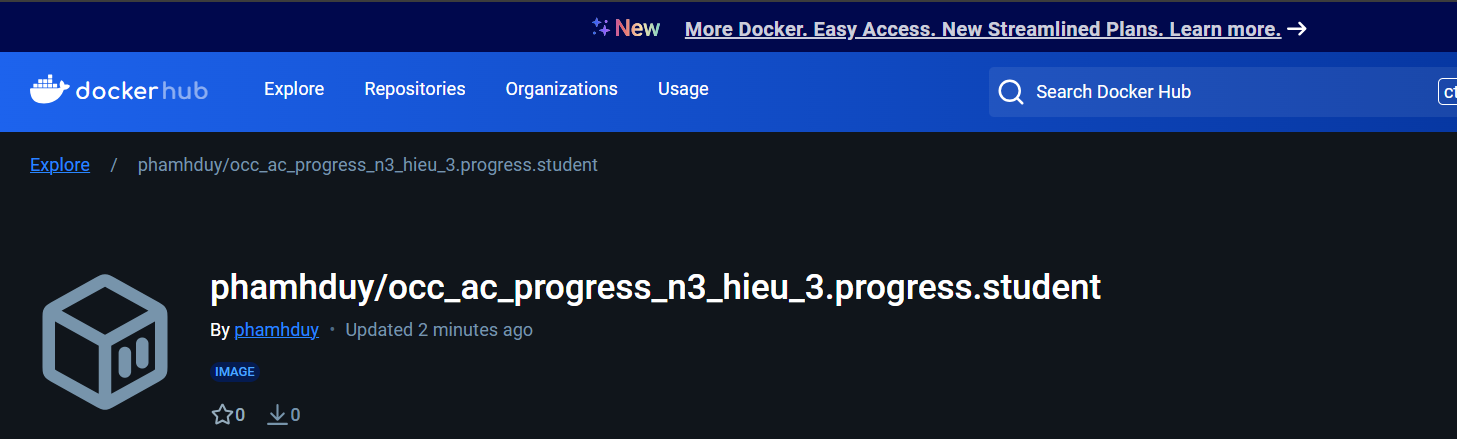
1. Treataslocal của resources
   1. Tích hợp và triển khai

Bài thực hành đã được triển khai như sau:

***Docker Hub***

https://hub.docker.com/r/phamhduy/occ\_ac\_progress\_n3\_hieu\_3.progress.student

1. Đẩy các vùng chứa lên dockerhub

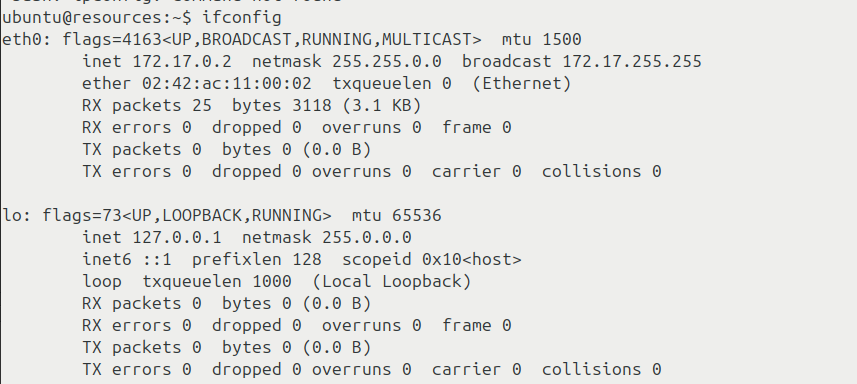


1. Đã đẩy lên docker

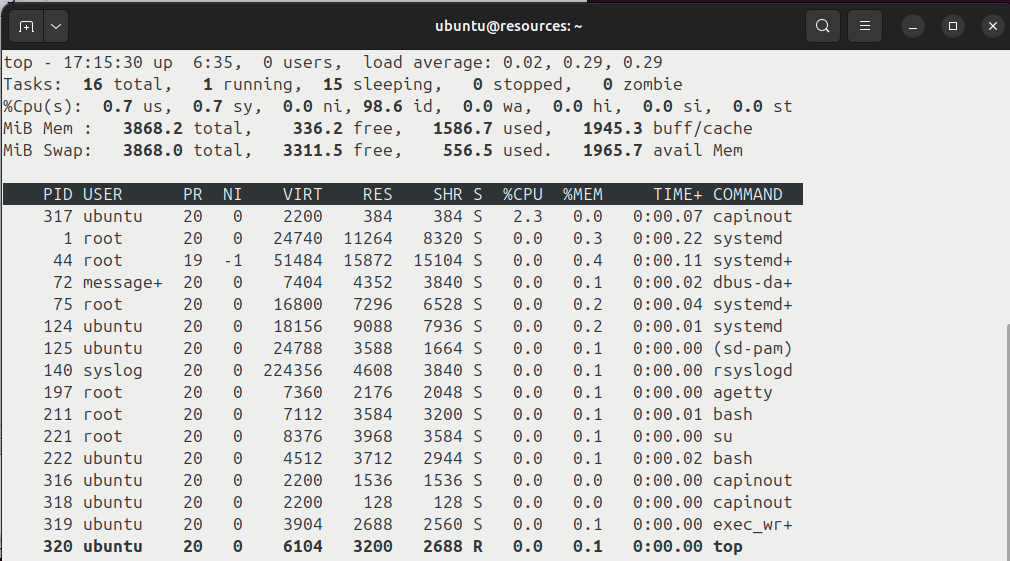
***Github***

* 1. Thử nghiệm và đánh giá

Bài thực hành đã được xây dựng thành công, dưới đây là hình ảnh minh họa về bài thực hành:



1. IP của máy resources



1. Quan sát tổng quan các thông tin về tiến trình bằng top

Lệnh top trong Linux là một công cụ mạnh mẽ giúp giám sát hiệu suất hệ thống theo thời gian thực. Khi chạy lệnh này, nó sẽ hiển thị một bảng thông tin chi tiết về các tiến trình đang chạy và tài nguyên hệ thống. Một số thông tin quan trọng:

* Thời gian hệ thống: Hiển thị thời gian hiện tại và thời gian hệ thống đã hoạt động.
* Số phiên người dùng: Số lượng người dùng đang đăng nhập vào hệ thống.
* Tình trạng CPU: Hiển thị mức sử dụng CPU, bao gồm phần trăm CPU đang được sử dụng bởi các tiến trình người dùng, hệ thống, và các tiến trình nhàn rỗi.
* Sử dụng bộ nhớ: Thông tin về tổng dung lượng RAM và swap, dung lượng đã sử dụng, còn trống, và dung lượng đang được sử dụng làm bộ đệm/cache.
* Danh sách tiến trình: Hiển thị danh sách các tiến trình đang chạy, bao gồm PID (Process ID), tên tiến trình, người dùng, mức sử dụng CPU và bộ nhớ của từng tiến trình.
* ***Nhiệm vụ 1:*** Tạo tiến trình mới

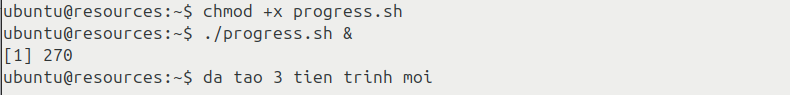
Viết mã script để tạo ra tiến trình mới và chia luồng CPU



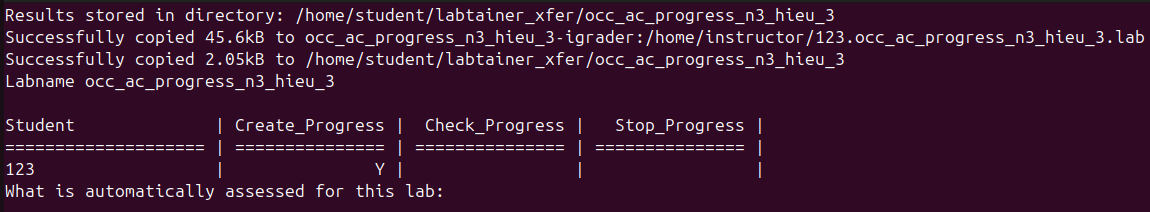
1. Nội dung file progress.sh

Sau đó thực hiện cấp quyền thực thi và chạy progress.sh ở dạng nền

* *chmod +x progress.sh*
* *./progress.sh &*



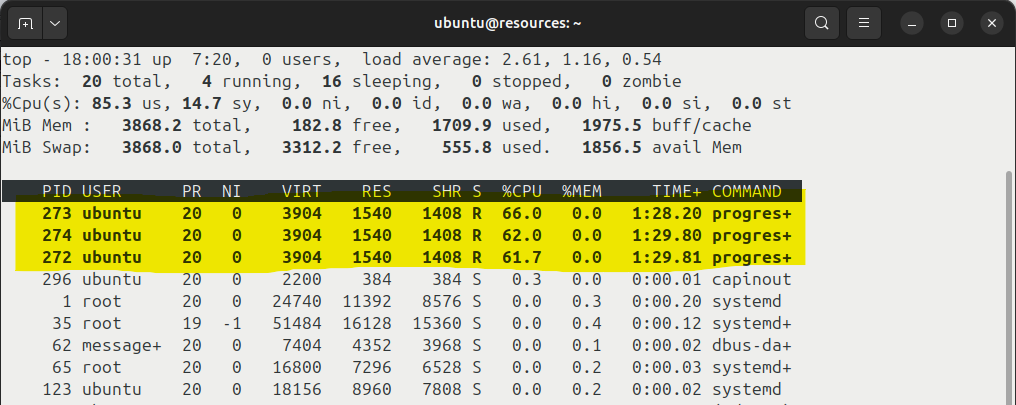
1. Cấp quyền và chạy progress.sh



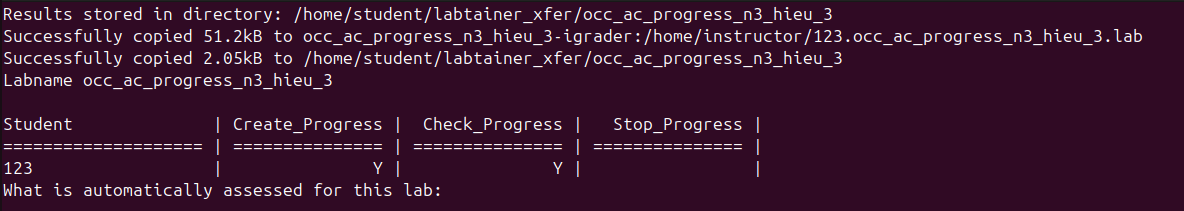
1. Checkwork nhiệm vụ 1

* ***Nhiệm vụ 2:*** Kiểm tra thông tin về các tiến trình mới được tạo.

Kiểm tra các thông tin tiến trình mới và tình trạng CPU, sự dụng bộ nhớ sau khi tạo tiến trình mới

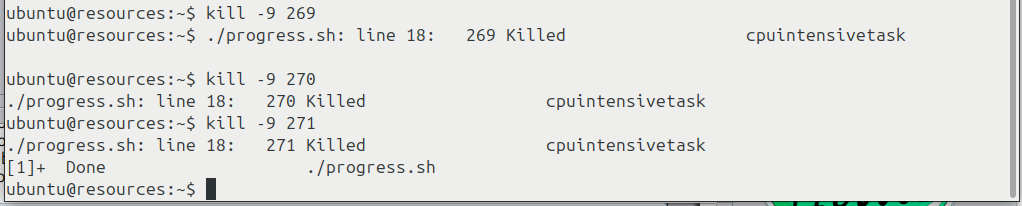


1. Thông tin về các tiến trình mới

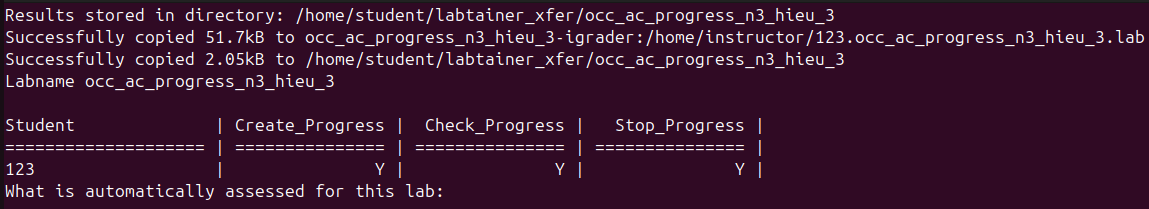


1. Checkwork nhiệm vụ 2

* ***Nhiệm vụ 3:*** Thực hiện kết thúc tiến trình vừa mới được tạo thông qua lệnh kill

******

1. Kết thúc tiến trình



1. Checkwork nhiệm vụ 3

TÀI LIỆU THAM KHẢO