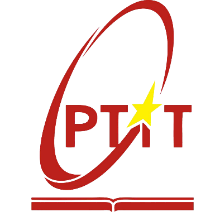
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-------🙞🙜🕮🙞🙜-------



**CÁC KỸ THUẬT GIẤU TIN**

**Đề bài: Tìm hiểu về chuẩn mã hóa video H264-AVC**

|  |  |
| --- | --- |
| **HỌ TÊN:** | **PHẠM HẢI DƯƠNG** |
| **MÃ SINH VIÊN:** | **B21DCAT072** |

Giảng viên hướng dẫn: PGS.TS. Đỗ Xuân Chợ

**HÀ NỘI, 03/2025**

**MỤC LỤC**

[1. Giới thiệu chung về bài thực hành 4](#_Toc196125231)

[2. Nội dung và hướng dẫn bài thực hành 4](#_Toc196125232)

[1.1. Mục đích 4](#_Toc196125233)

[1.2. Yêu cầu đối với sinh viên 4](#_Toc196125234)

[1.3. Nội dung thực hành 5](#_Toc196125235)

[3. Phân tích yêu cầu của bài thực hành 8](#_Toc196125236)

[4. Cài đặt và cấu hình máy ảo 8](#_Toc196125237)

[5. Tích hợp và triển khai 11](#_Toc196125238)

[5.1. Docker Hub 11](#_Toc196125239)

[5.2. GitHub 11](#_Toc196125240)

[6. Thử nghiệm và đánh giá 12](#_Toc196125241)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Giao diện Labedit 8](#_Toc196125041)

[Hình 2: Cấu hình result 9](#_Toc196125042)

[Hình 3: Cấu hình Result 9](#_Toc196125043)

[Hình 4: Cấu hình Goal 10](#_Toc196125044)

[Hình 5: Docker container bob 10](#_Toc196125045)

[Hình 6: Docker container bob 11](#_Toc196125046)

[Hình 7: Đẩy lab lên trên docker 11](#_Toc196125047)

[Hình 8: Đẩy lab lên Github 12](#_Toc196125048)

# **Giới thiệu chung về bài thực hành**

Bài thực hành này được thiết kế nhằm giúp sinh viên hiểu rõ cách thức hoạt động của chuẩn mã hóa video H264-AVC, từ quá trình nén video, mã hóa các khung hình đến giải mã và tái tạo lại video. Sinh viên sẽ được rèn luyện kỹ năng sử dụng các công cụ và thư viện lập trình để thực hiện các thao tác mã hóa và giải mã video, qua đó hiểu rõ về cách thức xử lý và tối ưu hóa dữ liệu video trong các ứng dụng truyền tải và lưu trữ.

H264-AVC (Advanced Video Coding) là một chuẩn mã hóa video phổ biến, được sử dụng rộng rãi trong các hệ thống phát sóng truyền hình, video streaming, và các dịch vụ lưu trữ video trực tuyến. H264-AVC cung cấp một mức độ nén video cao mà không làm giảm chất lượng hình ảnh quá nhiều, nhờ vào các kỹ thuật nén tiên tiến như nén không gian và nén thời gian, giúp giảm băng thông và dung lượng lưu trữ.

Sau khi hoàn thành bài thực hành, sinh viên sẽ có cái nhìn tổng quan về quy trình mã hóa và giải mã video theo chuẩn H264-AVC, nắm vững các bước trong việc xử lý video và hiểu rõ tầm quan trọng của việc áp dụng các kỹ thuật nén video trong các ứng dụng thực tế, cũng như các yếu tố ảnh hưởng đến chất lượng video và hiệu suất hệ thống.

# **Nội dung và hướng dẫn bài thực hành**

## **Mục đích**

* Hiểu về tổng quan về quy trình mã hóa và giải mã video theo chuẩn H264-AVC
* Nắm vững các bước thực hành sử dụng công cụ và các thuật toán mã hoa svideo trong chuẩn H264-AVC

## **Yêu cầu đối với sinh viên**

* Có kiến thức cơ bản về kỹ thuật giấu tin trong video
* Hiểu khái niệm và ứng dụng của video H264
* Biết cách sử dụng các công cụ media như ffmepg, Hnadbreak, mediainfo, …

## **Nội dung thực hành**

Sinh viên khởi động bài lab

Chạy lệnh:

***labtainer -r stego\_h264\_video\_encoding***

*(Chú ý: sinh viên sử dụng <TÊN\_TÀI\_KHOẢN> của mình để nhập thông tin người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm.)*

Sau khi khởi động bài lab, một container hiện lên sinh viên thực hiện làm theo yêu cầu

***TASK 1: Mã hóa video chuẩn h264 theo thuật toán CAVLC bằng FFMPEG***

* Sau khi khởi động xong một terminal ảo sẽ xuất hiện, terminal sử dụng để thực hành bài lab
* Trong terminal bob đã có sẵn 1 file video.mpeg.
* Sinh viên sử dụng công cụ ffmpeg để nén video mẫu thành 1 video theo chuẩn mã hóa H264 bằng thuật toán CAVLC:

***ffmpeg -i video.mpeg -c:v libx264 -profile:v baseline -x264opts cabac=0 -crf 28 video\_cavlc\_by\_ffmpeg.mp4***

* Câu lệnh này sử dụng ffmpeg để chuyển đổi video từ định dạng mpeg sang video H.264 với các tùy chọn mã hóa:
  + -c:v libx264 : chỉ định codec video mà bạn muốn sử dụng. Trong trường hợp này, libx264 là codec để mã hóa video với chuẩn H.264 (x264 là một thư viện mã hóa H.264). Codec này cho phép nén video với chất lượng cao và kích thước tệp nhỏ.
  + -profile:v baseline: chỉ định profile H.264 mà bạn muốn sử dụng. Baseline là profile cơ bản, phù hợp cho các thiết bị có phần cứng yếu hơn và cho các ứng dụng như streaming video hoặc video hội nghị. Profile này sẽ không sử dụng nhiều tính năng nén nâng cao như CABAC (Context-adaptive binary arithmetic coding), giúp quá trình mã hóa đơn giản hơn.
  + -x264opts cabac=0: -x264opts là tham số dành riêng cho x264 (codec H.264). cabac=0 vô hiệu hóa CABAC (Context-adaptive binary arithmetic coding), một kỹ thuật nén nâng cao giúp giảm kích thước tệp nhưng cũng yêu cầu tài nguyên hệ thống cao hơn.
  + -crf 28: -crf (Constant Rate Factor) là một tham số điều chỉnh chất lượng video.
* Video mới được tạo có tên là video\_cavlc\_byu\_ffmpeg.mp4. Sinh viên có thể sử dụng câu lệnh:

***ffprobe -v error -select\_streams v:0 -show\_entries stream=profile -of default=noprint\_wrappers=1 video\_cavlc\_by\_ffmpeg.mp4***

để có thể xem thông tin về profile của video mới này

***TASK 2: Mã hóa video chuẩn h264 theo thuật toán CAVLC bằng Handbreak***

* Vẫn sử dụng video mẫu như task1, nhưng lần này ta sử dụng công cụ HandBreakeCLI:

***HandBrakeCLI -i video.mpeg -o video\_cavlc\_by\_handbreak.mp4 -e x264 --encoder-profile baseline -q 28***

* -q 28: Tham số -q (Quality) điều chỉnh chất lượng video khi sử dụng mã hóa với codec x264.
* --encoder-profile baseline: Tham số này sử dụng profile baseline cho x264, điều này làm giảm các tính năng nén phức tạp của H.264
* Sinh viên kiểm tra profile của video mới:

***mediainfo video\_cavlc\_by\_handbreak.mp4 |grep "Format profile"***

***TASK 3: Mã hóa video chuẩn h264 theo thuật toán CABAC bằng FFMPEG***

* Lần này chúng ta sử dụng mpeg nhưng sử dụng thuật toán CABAC:

***ffmpeg -i video.mpeg -c:v libx264 -profile:v high -x264opts cabac=1 -crf 28 video\_cabac\_by\_ffmpeg.mp4***

* Câu lệnh lần này đã có các tham số khác đi câu lệnh ở task 1:
  + -profile:v high: -profile:v chỉ định profile H.264. high là một trong các profile của H.264 và cho phép sử dụng các tính năng nén nâng cao như CABAC, giúp video có chất lượng cao hơn và hiệu quả nén tốt hơn.
  + cabac=1 bật CABAC (Context-adaptive binary arithmetic coding), một kỹ thuật nén nâng cao trong chuẩn H.264. CABAC giúp cải thiện tỷ lệ nén và chất lượng video nhưng đòi hỏi phần cứng mạnh hơn và mất thêm thời gian mã hóa.
* Sau đó sinh viên sử dụng câu lệnh như task1 nhưng với video vừa tạo để xem profile của video mới.Nó có gì khác so với task1?

***TASK 4: Mã hóa video chuẩn h264 theo thuật toán CABAC bằng HandBreakCLI***

* Tương tự ffmpeg, HandBreakCLI cũng cho phép mã hóa 1 video theo chuẩn H264 bằng cách thay đổi 1 số tham số đầu vào:

***HandBrakeCLI -i video.mpeg -o video\_cabac\_by\_handbreak.mp4 -e x264 -q 28 -E copy --encoder-profile high***

* Sau khi tạo được video mới, kiểm tra thông tin profile của file đó bằng lệnh:

***mediainfo video\_cabac\_by\_handbreak.mp4 |grep "Format profile"***

***TASK 5: So sánh 2 thuật toán CAVLC và CABAC trong chuẩn mã hóa video H264***

* Sinh viên hãy ghi lại thông tin của 2 video đã được mã hóa theo 2 thuật toán CAVLC và CABAC:

***mediainfo video\_cavlc\_by\_handbreak.mp4 > cavlc\_info.txt***

***mediainfo video\_cabac\_by\_handbreak.mp4 > cabac\_info.txt***

* Sử dụng lệnh diff để tìm ra các thông số khác nhau giữa 2 video trên:

***diff cavlc\_info.txt cabac\_info.txt > compare.txt***

* Đọc file compare.txt để thấy được sự khác nhau giữa 2 video này. Chú ý các thông số về video (fize size, Format profile, Format settings, Bit rate, Stream size, Encoding settings, ….) và đưa ra nhận xét

Kết thúc lab:

* Trên terminal khởi động lab, sinh viên sử dụng lệnh:

***Stoplab***

* Khi bài lab kết thúc, một tệp lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab. Sinh viên cần nộp file .lab để chấm điểm.
* Để kiểm tra kết quả khi trong khi làm bài thực hành sử dụng lệnh:

***checkwork <tên bài thực hành>***

* Sinh viên cần nộp file .lab để chấm điểm.
* Kiểm tra kết quả trong quá trình làm bài:

***checkwork <tên bài lab>***

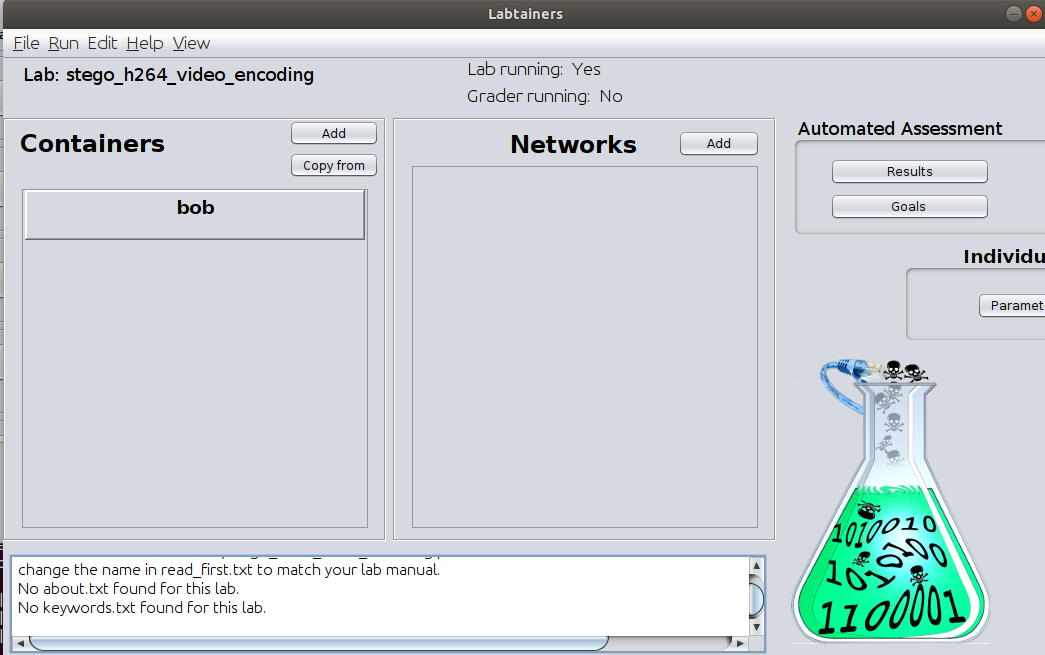
* Khởi động lại bài lab: Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

***labtainer -r stego\_h264\_video\_encoding***

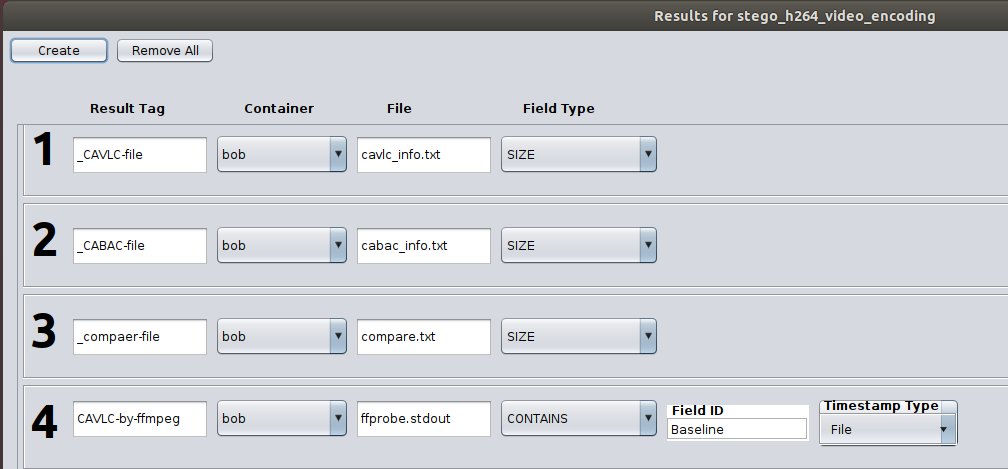
# **Phân tích yêu cầu của bài thực hành**

 Bài thực hành gồm 1 máy tính bob có chưa 1 video mẫu dạng mpeg .. Để hoàn thành bài thực hành sinh viên cần tạo được các video theo chuẩn mã hóa video h264 sử dụng 2 thuật toán là CAVLC và CABAC bằng 2 công cụ là ffmpeg và handbreakcli, sau đó so sánh chúng với nhau.

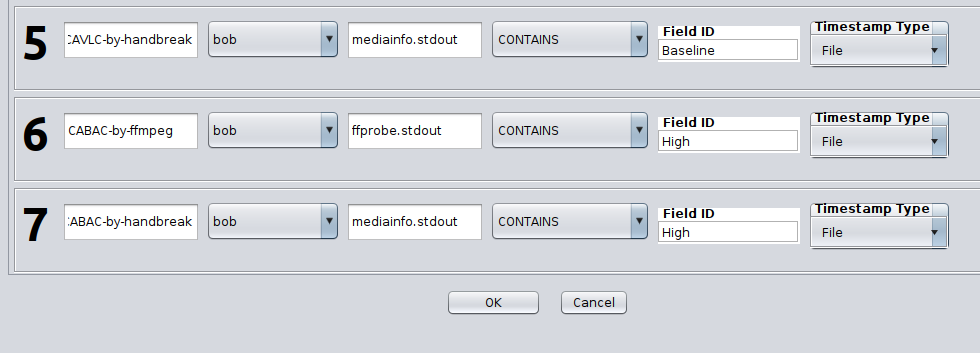
# **Cài đặt và cấu hình máy ảo**



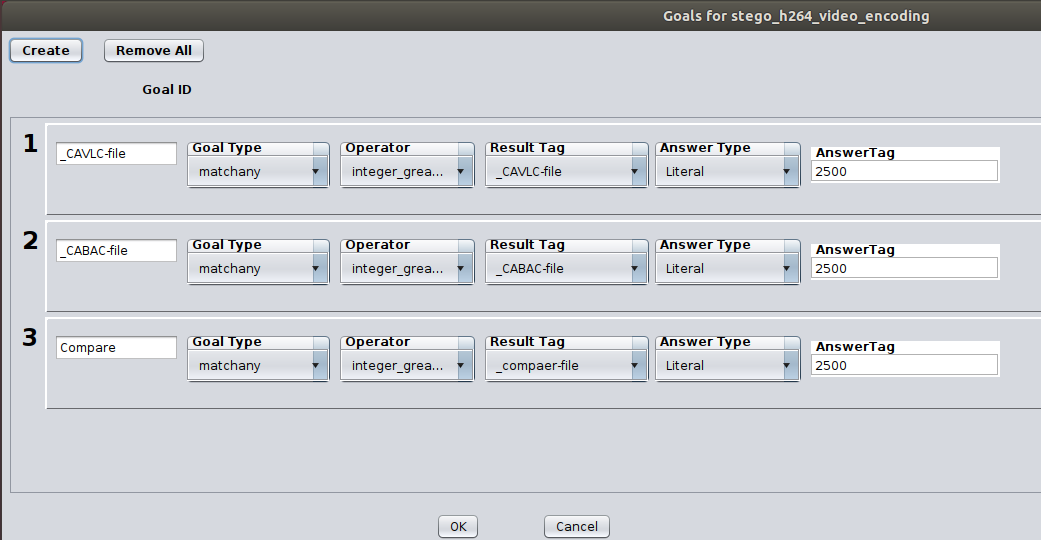
Hình 1: Giao diện Labedit



Hình 2: Cấu hình result



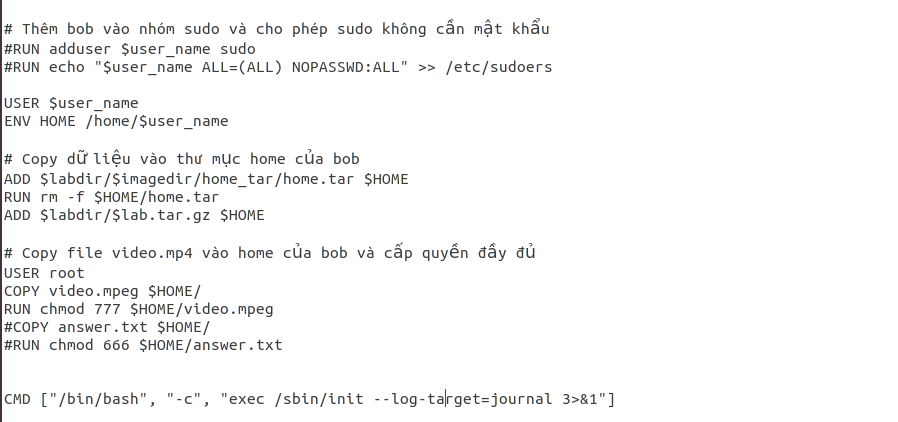
Hình 3: Cấu hình Result



Hình 4: Cấu hình Goal



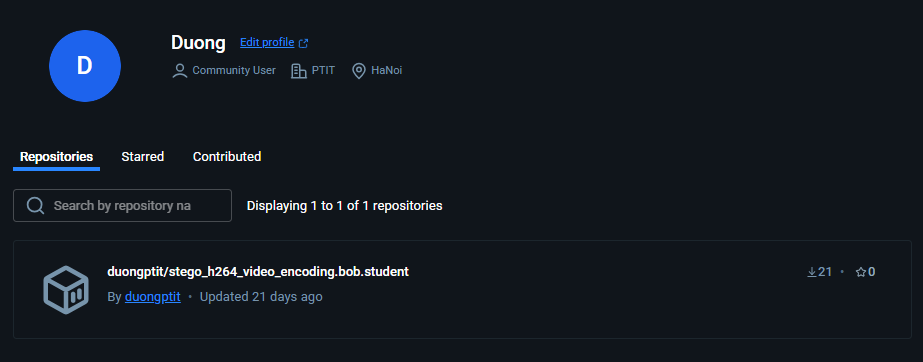
Hình 5: Docker container bob



Hình 6: Docker container bob

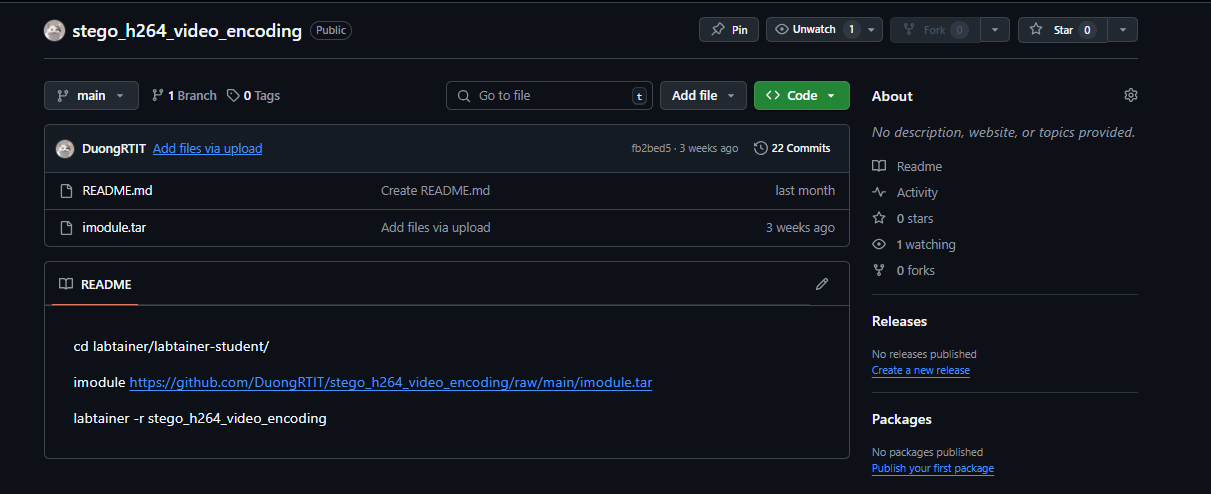
# **Tích hợp và triển khai**

## **Docker Hub**



Hình 7: Đẩy lab lên trên docker

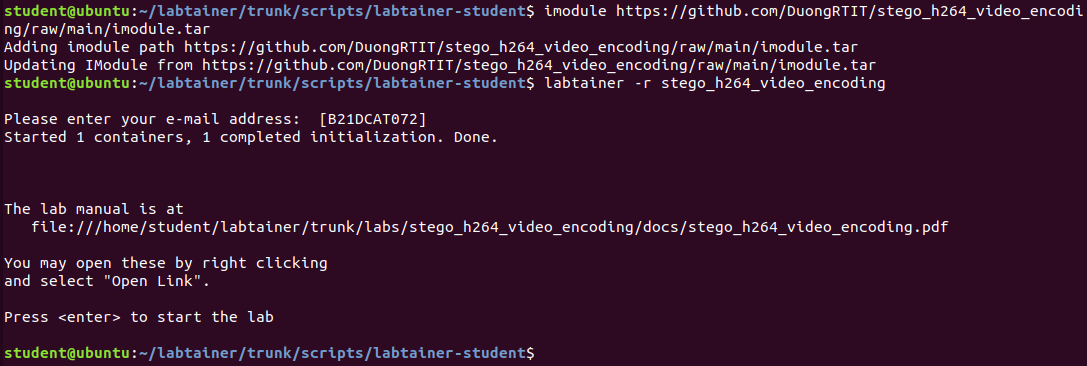
## **GitHub**



Hình 8: Đẩy lab lên Github

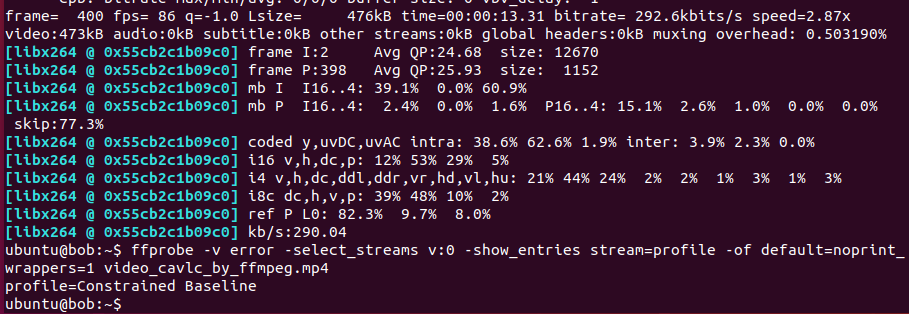
# **Thử nghiệm và đánh giá**

* Tải bài lab về từ Github và chạy bài lab:

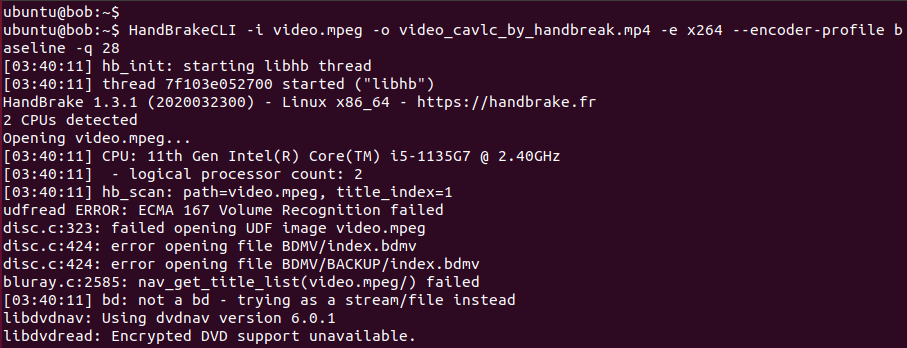


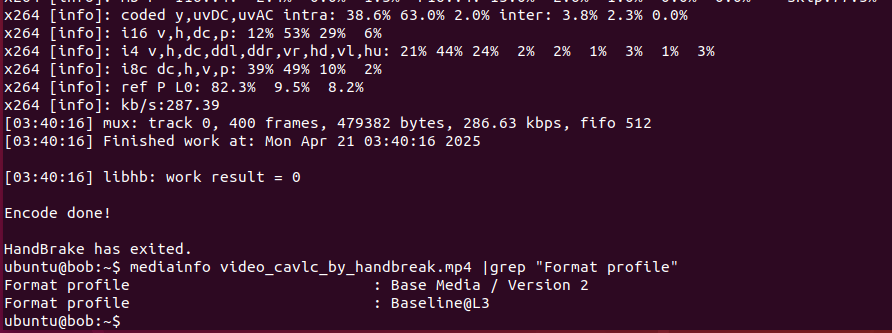
* Thực hiện Task 1:



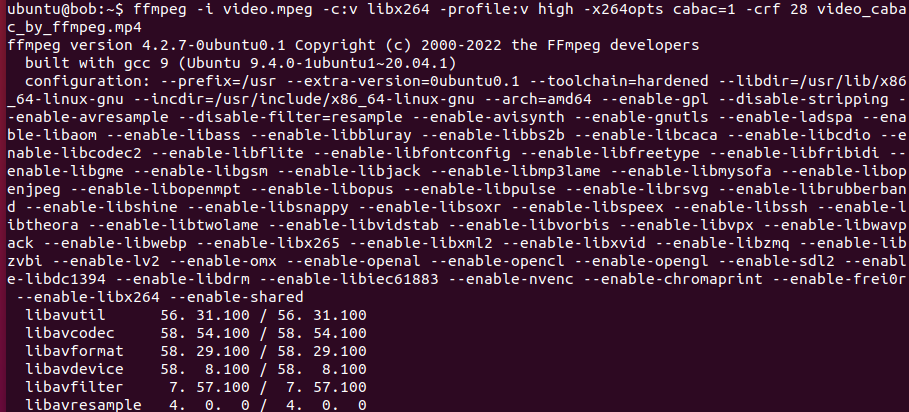


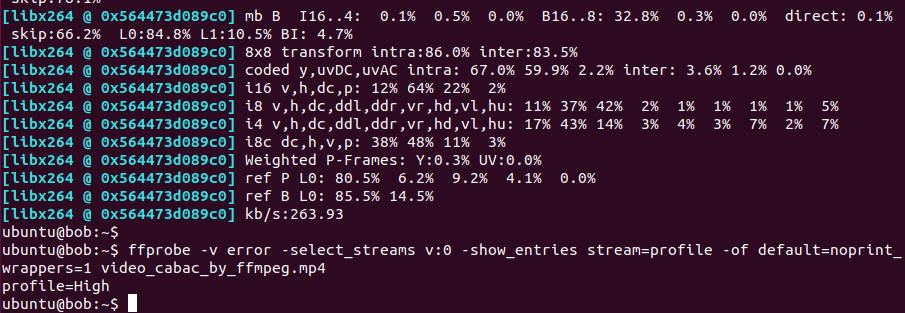
* Thực hiện task 2:



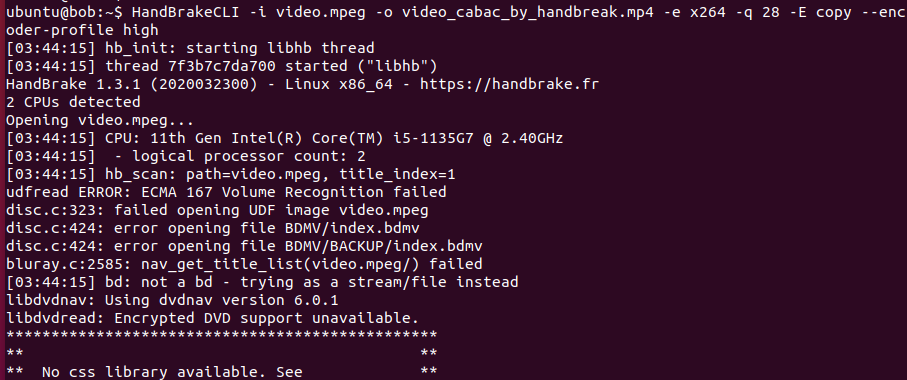


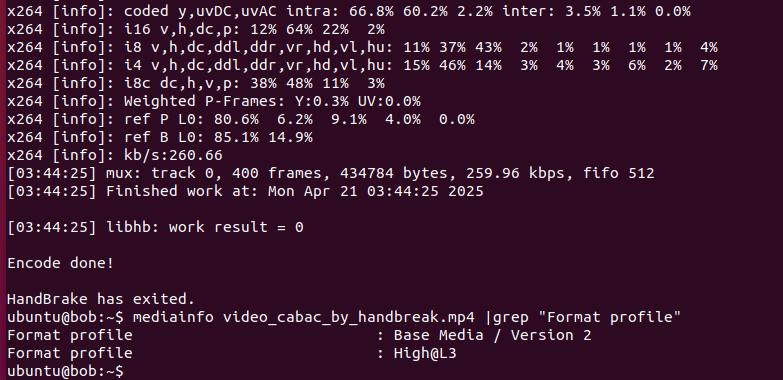
* Thực hiện task 3:



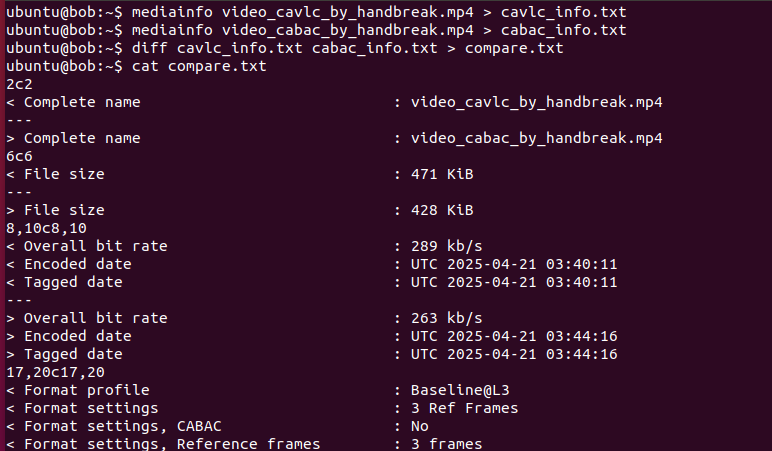


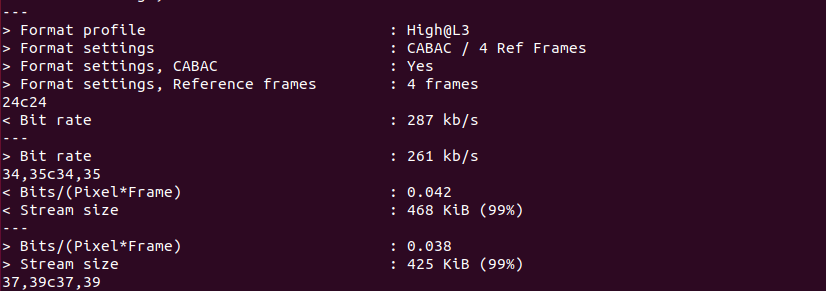
* Thực hiện task 4:

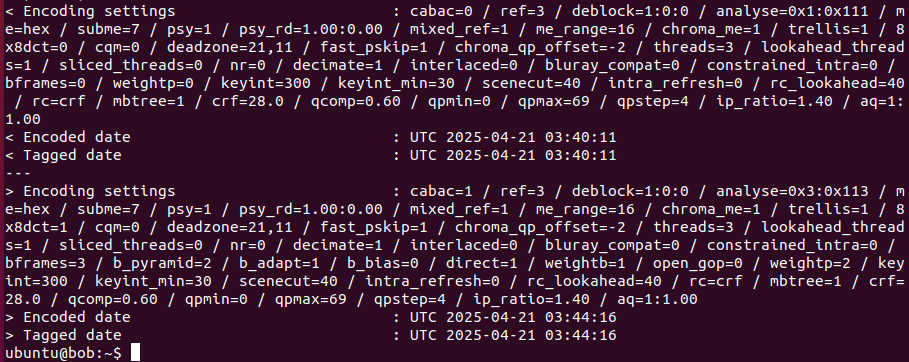




* Thực hiện task 5:







* Checkwork:

