

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

19120573 - Huỳnh Nguyễn Thị Lựu

19120592 - Đỗ Duy Nam

19120600 - Bùi Nguyên Nghĩa

**BÁO CÁO ĐỒ ÁN**

***Chủ đề về:***

**Quản lý bộ nhớ &&**

**Quản lý tiến trình**

Thầy Thái Hùng Văn

**Giảng viên:**

**Sinh viên:**

HỆ ĐIỀU HÀNH CQ2019/4

**MỤC LỤC**

[PHẦN I. Chương quản lý bộ nhớ 2](#_Toc93008463)

[1.Trình bày kỹ thuật phân trang và bộ nhớ ảo. 2](#_Toc93008464)

[1.1 Bộ nhớ ảo. 2](#_Toc93008465)

[1.2 Phân trang. 3](#_Toc93008466)

[2. Khảo sát và cho biết giá trị của các thông số quản lý bộ nhớ ảo trên một hệ thống máy tính cụ thể. 6](#_Toc93008467)

[PHẦN II. Chương quản lý tiến trình 7](#_Toc93008468)

[1.Phần thiết kế hệ thống. 7](#_Toc93008469)

[1.1 Đối với chương trình C. 7](#_Toc93008470)

[1.2 Đối với chương trình P. 8](#_Toc93008471)

[2.Phần cài đặt hệ thống. 9](#_Toc93008472)

[2.1 Các màn hình giao diện. 9](#_Toc93008473)

[2.2 Mã nguồn các hàm ứng với từng chức năng tương ứng. 9](#_Toc93008474)

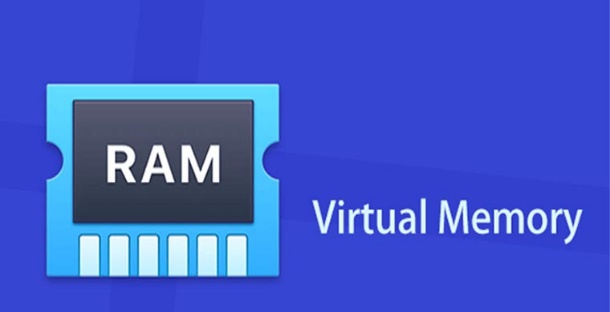
[PHẦN III: Phụ lục 10](#_Toc93008475)

[1.Trình bày cách chuyển python script sang exe bằng cách sử dụng Pyinstaller. 10](#_Toc93008476)

[2.Tạo project trên google console. 11](#_Toc93008477)

[3.Trình bày về semaphore. 18](#_Toc93008478)

[PHẦN IV: Các trích dẫn tham khảo được sử dụng trong bài báo cáo 19](#_Toc93008479)



# PHẦN I. Chương quản lý bộ nhớ

## 1.Trình bày kỹ thuật phân trang và bộ nhớ ảo.

**Đặt vấn đề:** Nếu ta muốn thực thi chương trình mà yêu cầu bộ nhớ lớn hơn bộ nhớ ta đang có sẵn thì phải làm thế nào?

**Giải pháp:** Giả lập chúng ta có bộ nhớ lớn hơn -> **Bộ nhớ ảo**.

### 1.1 Bộ nhớ ảo.

Bộ nhớ ảo = “True lie”

-Đối với User: không giới hạn, bộ nhớ riêng.

-Đối với OS: giới hạn, bộ nhớ dùng chung.

Bộ nhớ ảo:

-Là một phương pháp có ý tưởng cơ bản là khi chúng ta xét toàn bộ chương trình, gồm có mã nguồn, dữ liệu, stack thì tổng kích thước có thể lớn hơn kích thước vật lý của bộ nhớ chính. Vì vậy, hệ thống chỉ có thể lưu phần chương trình đang thực thi trên bộ nhớ chính, phần còn lại để trên ổ đĩa.

**Ví dụ:** Một chương trình 12MB có thể thực thi trên máy có 4MB bộ nhớ chính, hệ thống sẽ chia chương trình thành 3 phần khác nhau, mỗi phần được nạp lên bộ nhớ tại mỗi thời điểm mà nó thực thi.

-Bộ nhớ ảo cung cấp một không gian địa chỉ logic của bộ nhớ cho lập trình viên, nó tách biệt lập trình viên với địa chỉ vật lý của bộ nhớ.

-Cơ chế này cho người sử dụng thấy một không gian địa chỉ bộ nhớ ảo lớn hơn nhiều so với địa chỉ bộ nhớ vật lý thật sự.

-Không gian địa chỉ ảo của một tiến trình là không gian địa chỉ logic của tiến trình đó trên bộ nhớ ảo.

-Thông thường, địa chỉ logic của một tiến trình sẽ bắt đầu tại offset 0 và liên tục trên bộ nhớ ảo.

-Trên bộ nhớ vật lý, có thể chỉ có một phần của tiến trình trên bộ nhớ chính và phần còn lại trên bộ nhớ phụ và chúng cũng không nhất thiết liên tục nhau.

-Bộ nhớ ảo có thể được triển khai bằng cách phân trang hoặc phân đoạn, hiện tại phân trang thông dụng hơn.

**Hữu ích của bộ nhớ ảo:**

-Lập trình viên không cần lo lắng với việc các máy tính khác nhau có kích thước vật lý khác nhau.

-Phân mảnh trong môi trường đa chương.

### 1.2 Phân trang.

-Là lược đồ quản lý bộ nhớ cho phép không gian địa chỉ vật lý của tiến trình không nhất thiết phải liên tục nhau.

-Không gian địa chỉ ảo sẽ được chia thành các trang có kích thước cố định.

-Tương ứng cho mỗi trang trên địa chỉ ảo sẽ có một trang thật có cùng kích thước trên bộ nhớ vật lý.

-Chuyển đổi địa chỉ thông qua bảng trang, được đánh chỉ mục bằng page number.

-Mỗi mục tin trong bảng trang lưu một con số đại diện page frame mà trang đó ánh xạ tới và trạng thái của trang trong bộ nhớ.

-Các trạng thái: valid/invalid, access permission, reference bit, modified bit, caching.

-Việc phân trang là “trong suốt” với người lập trình.

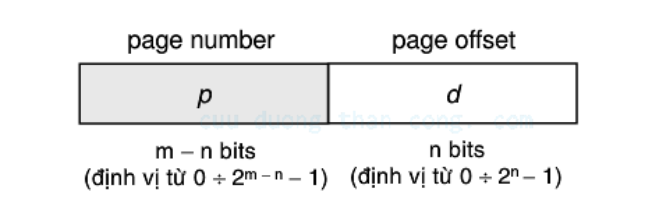
**Cơ chế phân trang:**

Địa chỉ luận lý gồm có:

-Số hiệu trang(Page number): p

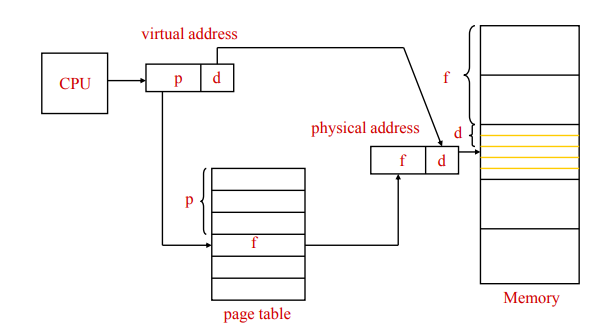
-Địa chỉ tương đối của trang (Page offset): d

-Nếu kích thước của không gian địa chỉ vật lý là 2m và kích thước của trang là 2n thì

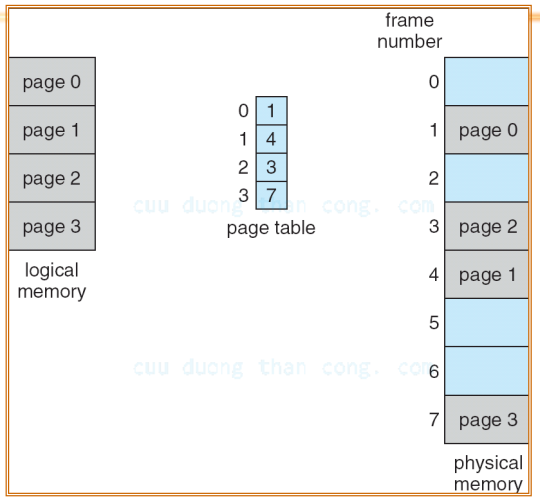


Bảng trang sẽ có tổng cộng 2m/2n=2m-n (entry)

**Chuyển đổi địa chỉ trong phân trang:**



**Mô hình phân trang:**



**Bảng trang:**

-Mục đích của bảng trang là ánh xạ địa chỉ ảo sang địa chỉ vật lý.

-Bảng trang là một hàm số mà tham số đầu vào là địa chỉ ảo và đầu ra là địa chỉ vật lý.

**Bảng trang giải quyết 2 bài toán lớn:**

-Kích thước bảng trang có thể rất lớn.

-Ánh xạ phải nhanh.

*Vấn đề thứ nhất*, thực tế các máy tính bây giờ sử dụng ít nhất 32 bit địa chỉ ảo. Nếu mỗi trang có kích thước là 4KB thì phải có một triệu trang. Như vậy bảng trang phải có một triệu mẫu tin.

*Vấn đề thứ hai*, việc truy cập địa chỉ bộ nhớ là thường xuyên, vì vậy cần một hàm ánh xạ thật nhanh để tăng tốc xử lý của hệ thống.

**Câu hỏi:** Bảng trang được lưu ở đâu? Tốn thêm bộ nhớ vật lý lưu bảng trang để giải quyết vấn đề thiếu hụt bộ nhớ? **Nghịch lý!!**

->*Trên thực tế*, kích thước bộ nhớ sử dụng để chạy phần mềm trên bộ nhớ ảo nhiều hơn, còn phần bộ nhớ vật lý thì giảm.

**Các thuật toán thay trang:**

Khi có lỗi xuất hiện thì hệ thống phải chọn một trang “nạn nhân” để loại ra khỏi bộ nhớ, nhường không gian cho trang đang được yêu cầu. Nếu trang được chọn đã có sửa đổi (D=”1”) thì hệ thống phải cập nhật nội dung trang này trên bộ nhớ phụ. Trang mới lấy từ bộ nhớ phụ chỉ việc chép chồng lên trang được chọn làm thay thế. Trang “nạn nhân” thông thường là trang phải là trang được sử dụng đến ít nhất. Vì nếu hệ thống chọn một trang được sử dụng rất nhiều làm trang nạn nhân thì tổng chi phí cho việc thay trang của hệ thống tăng lên đáng kể.

Có rất nhiều nghiên cứu về các giải thuật thay trang, sau đây là một số thuật toán thay trang cơ bản sẽ được tìm hiểu: Thuật toán thay trang tối ưu, Thay trang ít sử dụng gần đây nhất(LRU), Chiến lược FIFO.

**Thuật toán thay trang tối ưu:**

-Thuật toán này giả thiết mỗi trang có một thông tin tham chiếu khi nào trang đó được gọi lại.

-Khi có lỗi trang xảy ra, thuật toán sẽ chọn trang có số tham chiếu lớn nhất làm nạn nhân.

-Thuật toán dễ hiểu nhưng không thực tế vì không thể biết chắc chắn các sự kiện xảy ra trong tương lai -> giả thiết của thuật toán không đúng hoàn toàn.

**Thay trang ít được sử dụng gần đây nhất:**

-Quan sát tần suất các trang được gọi sử dụng trong quá khứ.

-Nếu trang được gọi sử dụng nhiều thì xác suất nó sẽ được gọi lại trong tương lai gần nhiều hơn so với trang ít được gọi hơn.

-Thuật toán LRU sẽ chọn trang ít được sử dụng gần đây nhất làm trang nạn nhân.

-Thuật toán được chấp thuận về mặt lý thuyết, việc cài đặt lại tốn nhiều chi phí. Thuật toán phải dựa vào sự hỗ trợ của phần cứng để xác định thời điểm cuối trang được truy xuất hoặc sử dụng stack hay danh sách liên kết -> Có thêm tag xác định thời điểm truy cập cuối -> Tốn chi phí.

**Chiến lược FIFO:**

-Trang được vào hệ thống sớm nhất sẽ là trang được chọn để thay thế.

-Hệ điều hành sử dụng một danh sách với con trỏ đầu và cuối trang để trỏ vào trang tồn tại trong bộ nhớ lâu nhất và trang mới được nạp vào bộ nhớ.

-Thuật toán đơn giản nhất để triển khai cài đặt.

## 2. Khảo sát và cho biết giá trị của các thông số quản lý bộ nhớ ảo trên một hệ thống máy tính cụ thể.

**Hệ thống máy tính có:**

RAM: 4GB

Địa chỉ logic hệ thống: 32bit

Kích thước trang: 4KB

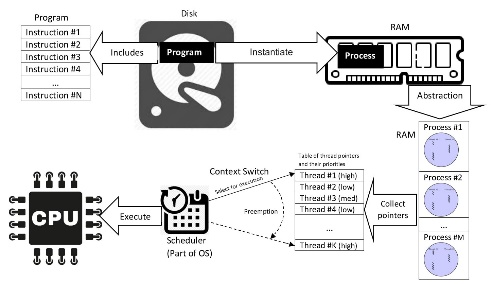
Page Size actually refers to its real world counterpart – Page Frame, whose size is typically between 1KB and 8KB and is generally 4 KB for 32 bit systems.

For a 32 bit system, with 4KB Page Size (2^12 bytes),

Number of Pages = Virtual Memory Space / Page Size = 2^32 / 2^12 = 2^20.

Or, Virtual Memory Space = Number of Pages \* Page Size

2^32 = 2^20 \* 2^12



# PHẦN II. Chương quản lý tiến trình

**Yêu cầu:** Xây dựng hệ thống chương trình cho phép liên lạc giữa các tiến trình trong một máy và/ hoặc trên các máy (với các hệ điều hành có thể khác nhau); sao cho chúng có khả năng phòng ***Deadlock*** hoặc tránh cùng rơi vào ***Critical section***.

## 1.Phần thiết kế hệ thống.

**Cụ thể đối với mã nguồn được viết bằng python.**

### 1.1 Đối với chương trình C.

-Auto run: Chuyển python script của chương trình thành file exe rồi sau đó bỏ vào….

Các bước chuyển python script sang exe bằng cách sử dụng Pyinstaller cụ thể được trình bày ở ***phụ lục 1***.

-Lưu trữ và chia sẻ dữ liệu: Dữ liệu sẽ được lưu trên googledrive để liên lạc, giao tiếp, trao đổi dữ liệu giữa các tiến trình khác nhau. Sử dụng google drive api để triển khai, cách tạo project trên google console sẽ được trình bày cụ thể ở ***phụ lục 2***. Ở đây ta cần giải quyết vấn đề đụng độ tài nguyên khi nhiều tiến trình cùng lúc muốn ghi dữ liệu vào cùng 1 tập tin -> Giải quyết bài toán đồng bộ miền găng.

- Chức năng nhập mật khẩu để kiểm tra mật khẩu của trẻ hay phụ huynh:

Mật khẩu mặc định của trẻ là ‘children’, của phụ huynh là ‘parent’

Trẻ điền thông tin mật khẩu vào màn hình giao diện được viết sử dụng ***tkinter*** để chương trình nhận thông tin và kiểm tra mật khẩu.

-Chức năng đợi một khoảng thời gian (cụ thể trong đề bài là 60 phút) khi mật khẩu được nhập vào là của phụ huynh: Sau khi tiến hành lấy mật khẩu và kiểm tra từ chương trình C, nếu mật khẩu được đưa vào khớp với mật khẩu phụ huynh, cửa sổ giao diện nhập mật khẩu sẽ bị huỷ và đồng thời xuất hiện cửa sổ đếm thời gian đợi 60 phút được hiển thị. Sau khi thời gian đợi kết thúc, cửa sổ đếm thời gian đợi sẽ bị huỷ và đồng thời cửa sổ lấy thông tin mật khẩu sẽ được hiển thị trở lại.

-Chức năng đọc dữ liệu từ googledrive để lấy các thông tin về quản lý khung giờ sử dụng máy của trẻ: Thực hiện hàm đọc dữ liệu, parse dữ liệu file Input.txt thành mảng các khung giờ và ràng buộc thời gian sử dụng máy.

-Chức năng cho trường hợp thời điểm hiện tại không nằm trong khung giờ sử dụng máy: Hiển thị cửa sổ thông báo mốc thời gian được sử dụng gần nhất tiếp theo sau đó huỷ cửa sổ thông báo và thực hiện song song 2 công việc sau:

* Tạo 1 Thread phụ thực hiện công việc kiểm tra đã đủ 15 giây chưa kể từ khi thông báo. Nếu đủ thì chương trình tự tắt máy -> Tắt chương trình và shutdown hệ điều hành.
* Trong luồng chính sau khi huỷ cửa sổ thông báo mốc thời gian được sử dụng gần nhất tiếp theo thì cho cửa sổ đăng nhập bằng mật khẩu xuất hiện để tiến hành đăng nhập (nếu có).
* Ở trường hợp này có quá trình giao tiếp giữa 2 tiểu trình về việc nếu phụ huynh đăng nhập thì Thread phụ sẽ không thực hiện tự động tắt máy nữa mà chuyển sang cửa sổ đợi 60 phút -> Sử dụng flag để liên lạc giữa 2 tiểu trình, flag đánh dấu việc đăng nhập ở luồng chính có phải là phụ huynh không.

-Thông báo còn bao nhiêu phút nữa tắt máy và đến mấy giờ có thể bật lại: sử dụng cửa sổ giao diện gồm bộ đếm thời gian còn lại và label hiển thị khung giờ được sử dụng máy trở lại.

-Chạy chương trình ở chế độ sát thực hiện cùng lúc 2 công việc:

* Lưu màn hình, các phím đã gõ sau mỗi phút: (Thread phụ) Luồng lắng nghe sự kiện liên tục thực hiện việc chụp ảnh màn hình và ghi nhớ các phím đã gõ, sau một phút thực hiện viết các dữ liệu theo dõi được lên google drive.
* Kiểm tra dữ liệu file Input.txt trên google drive có bị thay đổi bởi phụ huynh thì cập nhật lại tương ứng: (Thread phụ) -> Sử dụng flag để đánh dấu việc đã đồng bộ dữ liệu giữa chương trình P và chương trình C chưa.
* Kiểm tra thấy còn 1 phút đến thời điểm tắt máy thì hiển thị cửa sổ thông báo thời gian còn lại và giờ tiếp theo được sử dụng máy: (Thread phụ)

### 1.2 Đối với chương trình P.

-Giao diện trang chủ của chương trình P sẽ có menu gồm 2 lựa chọn là “Quản lý khung giờ sử dụng” và “Quản lý lịch sử sử dụng máy”. Ta click vào lựa chọn tương ứng để thực hiện thao tác mong muốn.

-Giao diện “Quản lý khung giờ sử dụng” có textarea đề hiển thị các khung giờ tương ứng trong tập dữ liệu Input. Phụ huynh có thể xem và thực hiện các thao tác thêm, xoá, sửa và sau đó upload lên google drive để lưu lại các thay đổi.

-Giao diện “Quản lý lịch sử sử dụng máy” sẽ hiển thị list gồm các ảnh chụp màn hình và list gồm các key mà máy thu thập được và ghi lại trong quá trình trẻ sử dụng máy.

-Về việc quản lý khung giờ được sử dụng máy của trẻ (nội dung file text):

* Đối với việc đọc nội dung file text thì nhiều phụ huynh truy cập cùng lúc được cho phép.
* Đối với việc chỉnh sửa nội dung file text thì chỉ có 1 phụ huynh được thực hiện vào 1 thời điểm xác định -> Hàm thực hiện chức năng chỉnh sửa nội dung file text là đoạn miền găng -> Chọn giải quyết bài toán miền găng bằng cách sử dụng ***semaphore***.
* Việc chọn giải pháp semaphore vì phương pháp này là một phương pháp giải quyết tốt bài toán đồng bộ. Phần kiến thức về semaphore được trình bày cụ thể hơn ở ***phụ lục 3***.

## 2.Phần cài đặt hệ thống.

### 2.1 Các màn hình giao diện.

### 2.2 Mã nguồn các hàm ứng với từng chức năng tương ứng.

# PHẦN III: Phụ lục

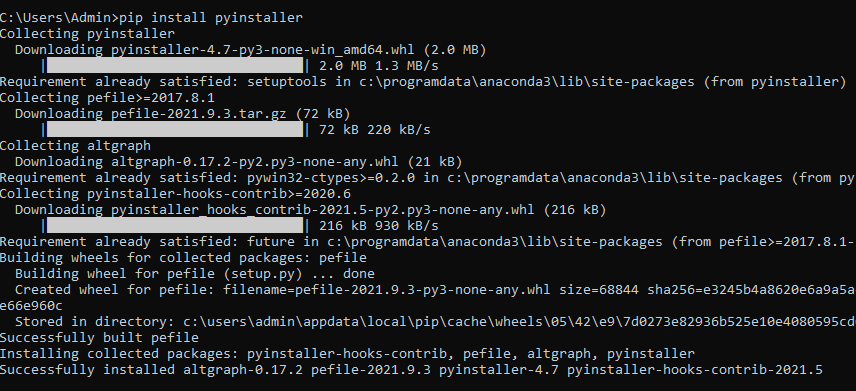
## 1.Trình bày cách chuyển python script sang exe bằng cách sử dụng Pyinstaller.

**Bước 1:** Thêm python vào Windows Path

**Bước 2:** Mở cửa sổ Command Prompt

**Bước 3:** Cài đặt Pyinstall Package

Trong cửa sổ cmd vừa mở ở ***Bước 2***, gõ dòng lệnh pip install pyinstaller và sau đó nhấn Enter



**Bước 4:** Chuẩn bị python script mà ta muốn convert qua file exe

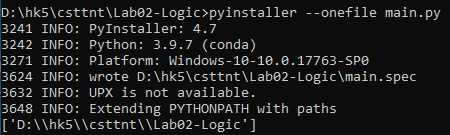
**Bước 5:** Tạo file exe sử dụng Pyinstaller

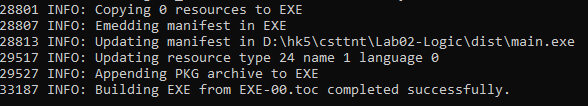
Tại thư mục chứa file python muốn convert ta mở cửa sổ cmd

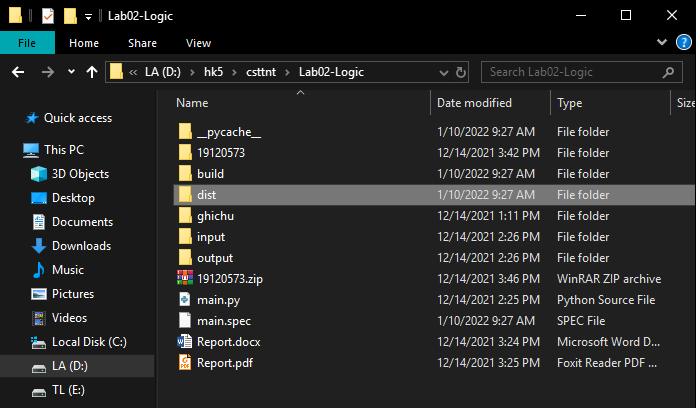
Tiếp theo, gõ mẫu dòng lệnh ***pyinstaller –onefile filename.py***

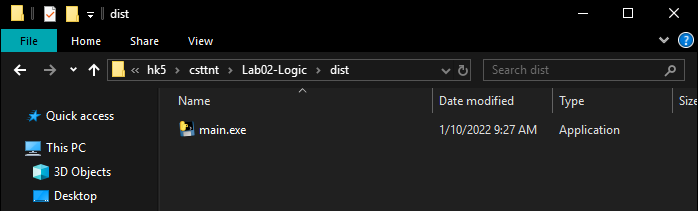
Sau khi chạy xong lệnh tập tin exe sẽ nằm trong thư mục dist

Các hình ảnh minh hoạ cho việc chuyển tập tin python sang exe. Cụ thể trong trường hợp này là tập tin main.py



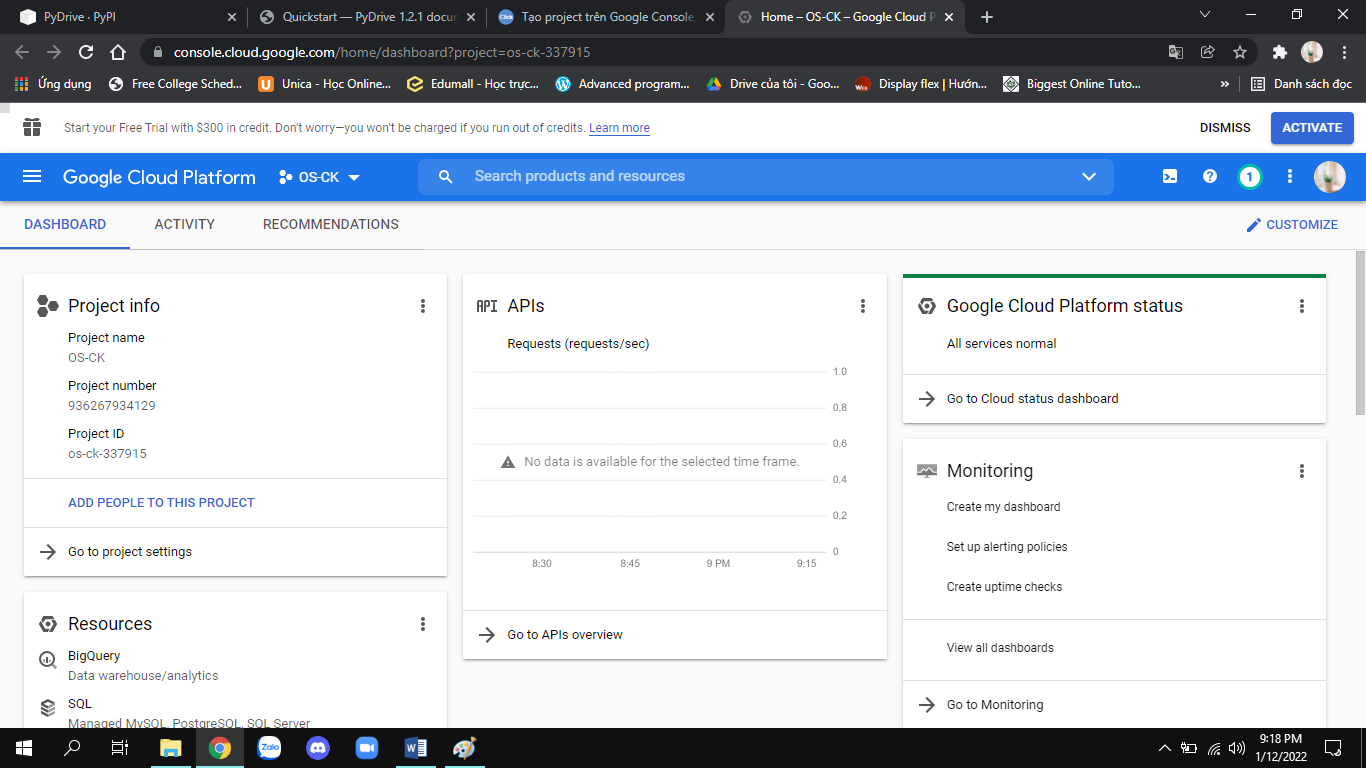




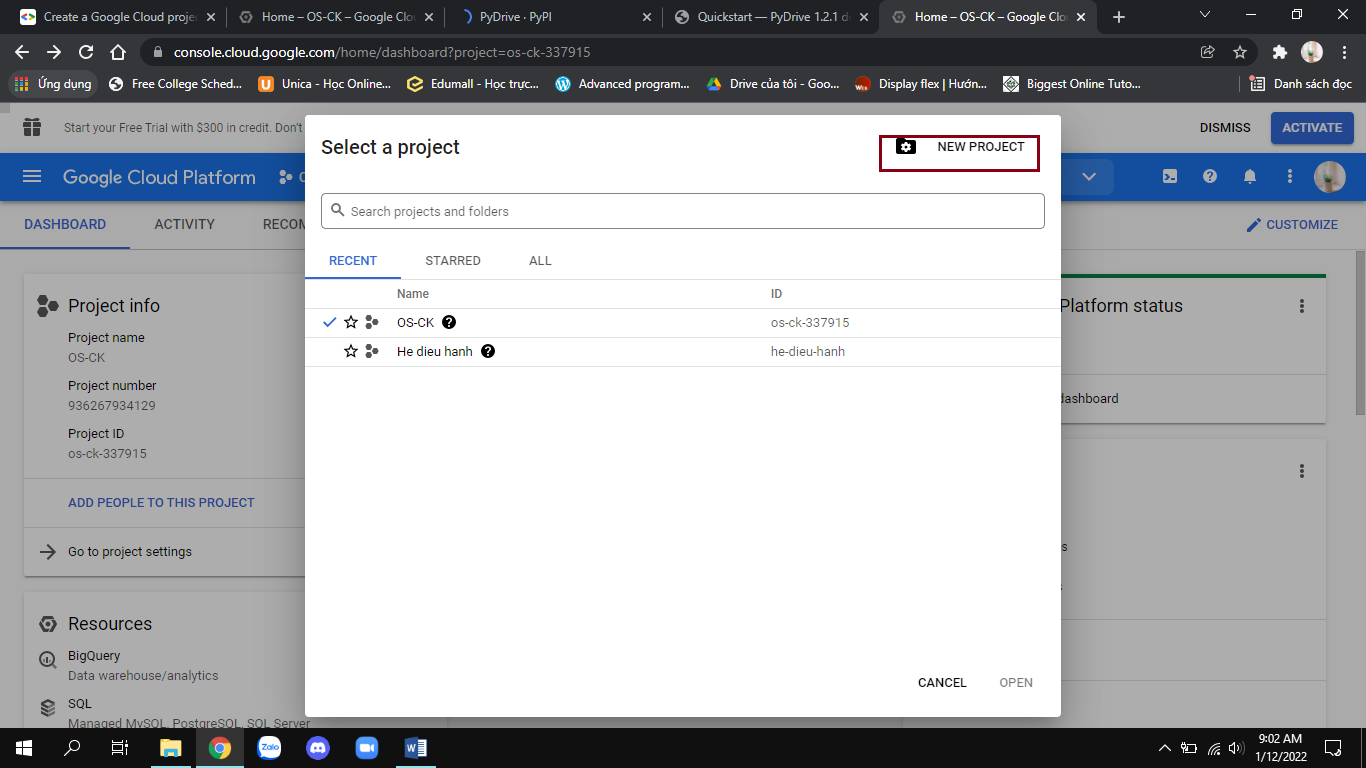


## 2.Tạo project trên google console.

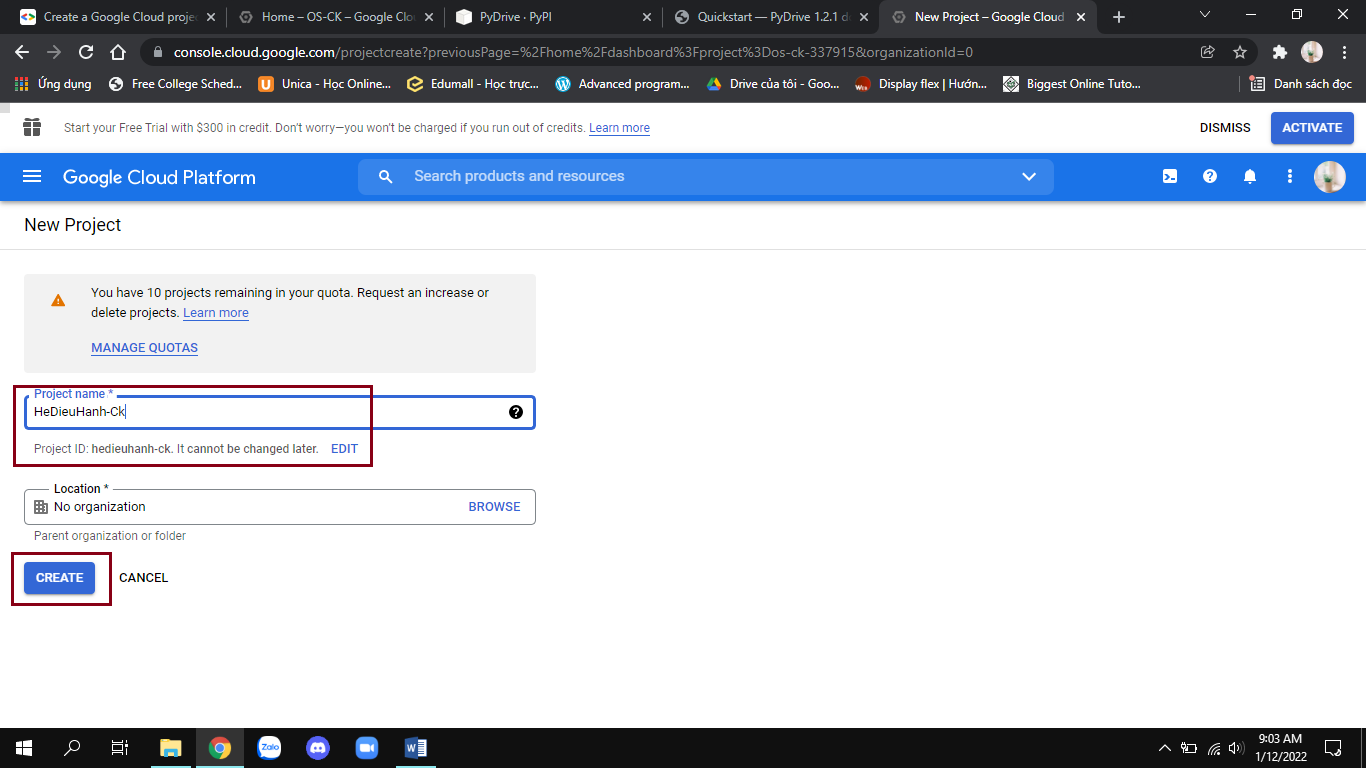
**-**Truy cập vào link <https://console.cloud.google.com/> bằng tài khoản google để vào Console Clound.



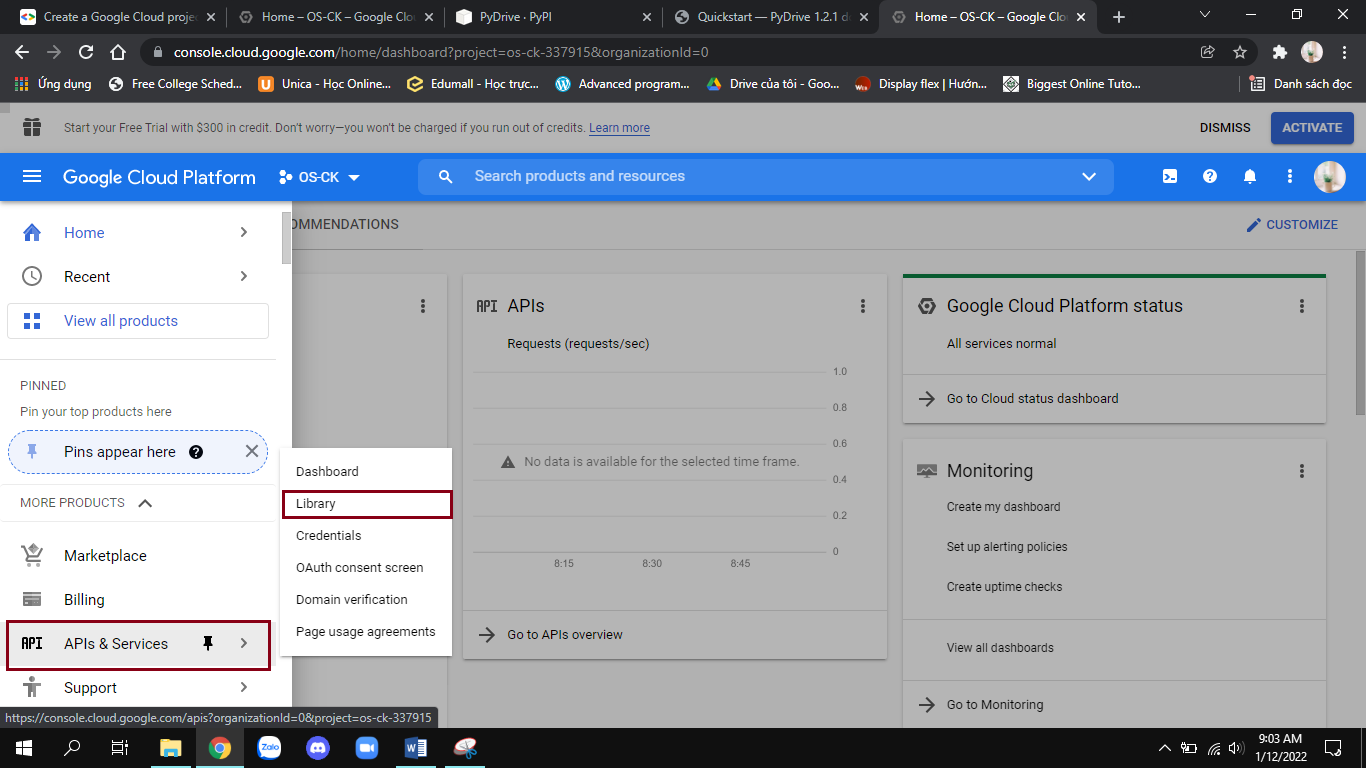
**-**Chọn “New Project” để tạo một project mới.



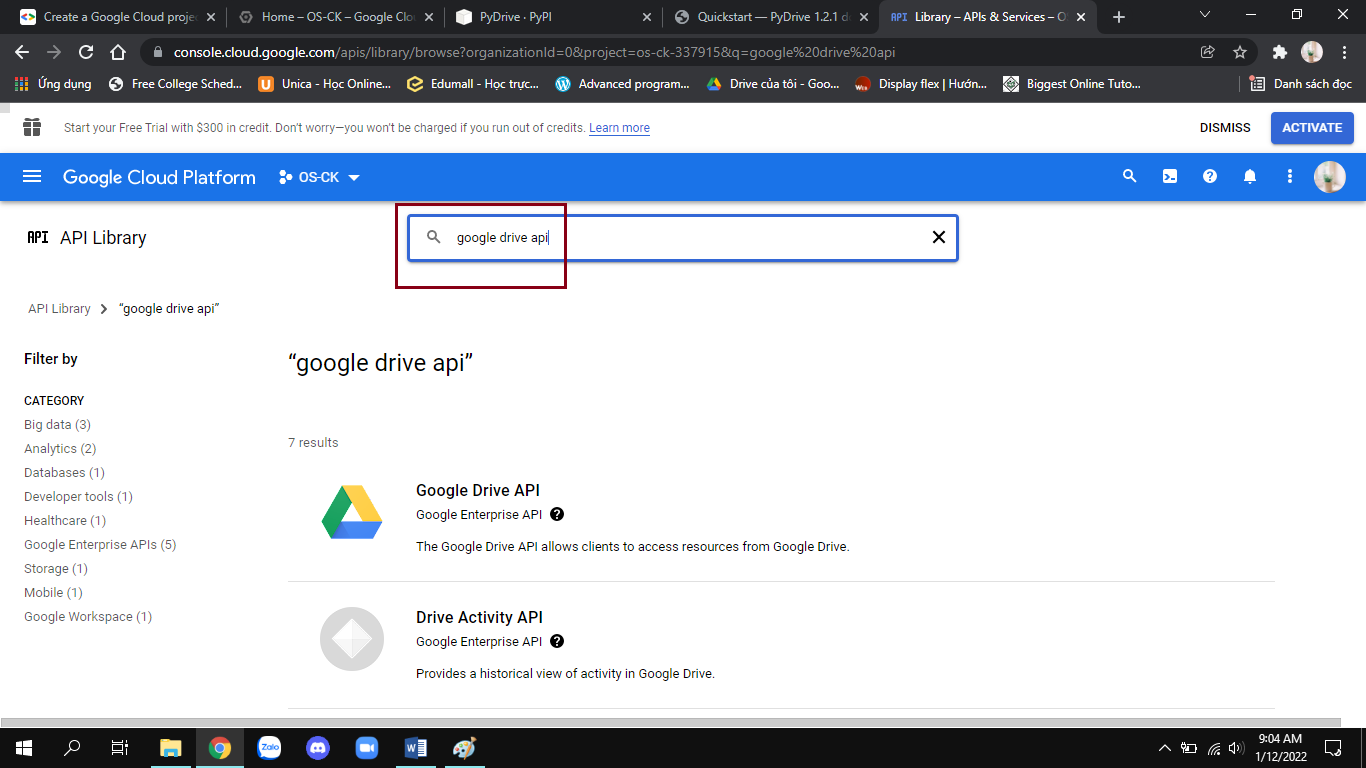
**-**Nhập tên của Project và sau đó chọn “Craete”.



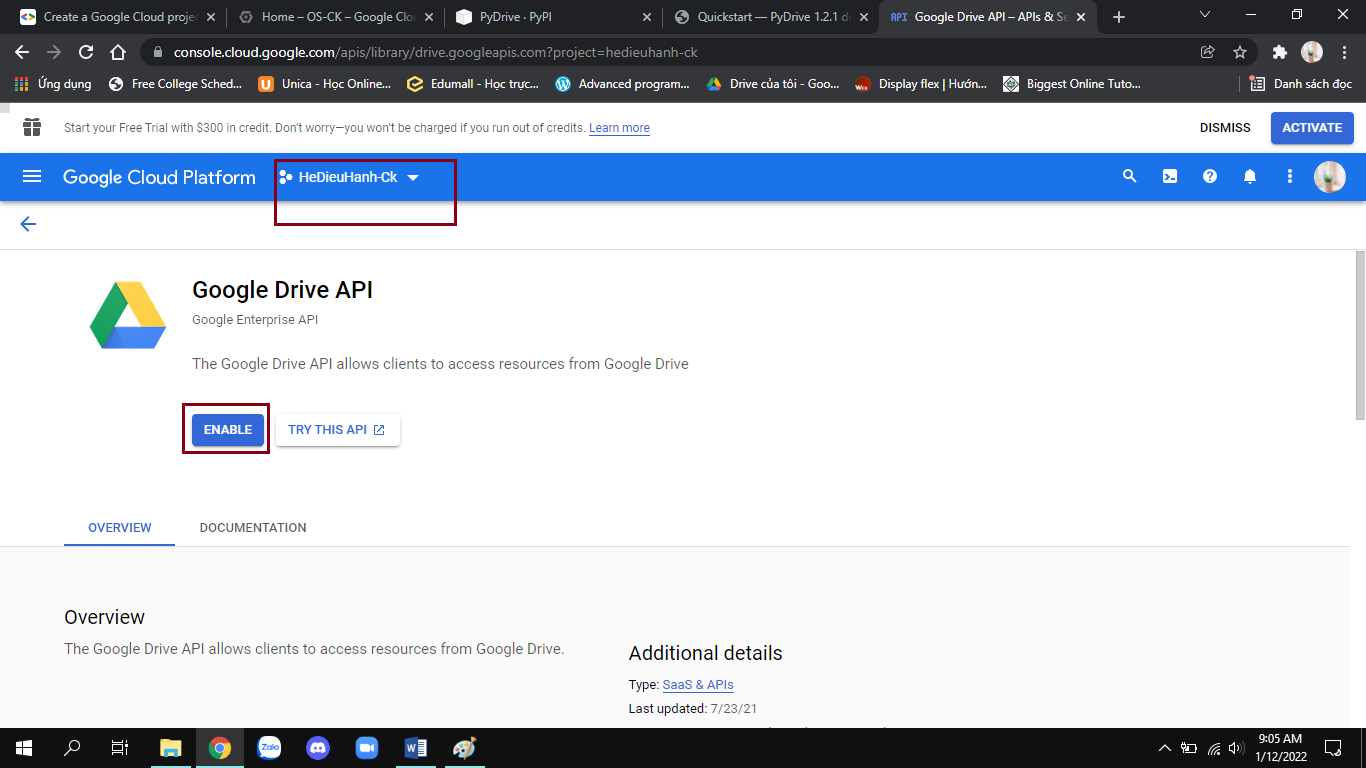
-Chọn “APIs & Services” -> “Library”.



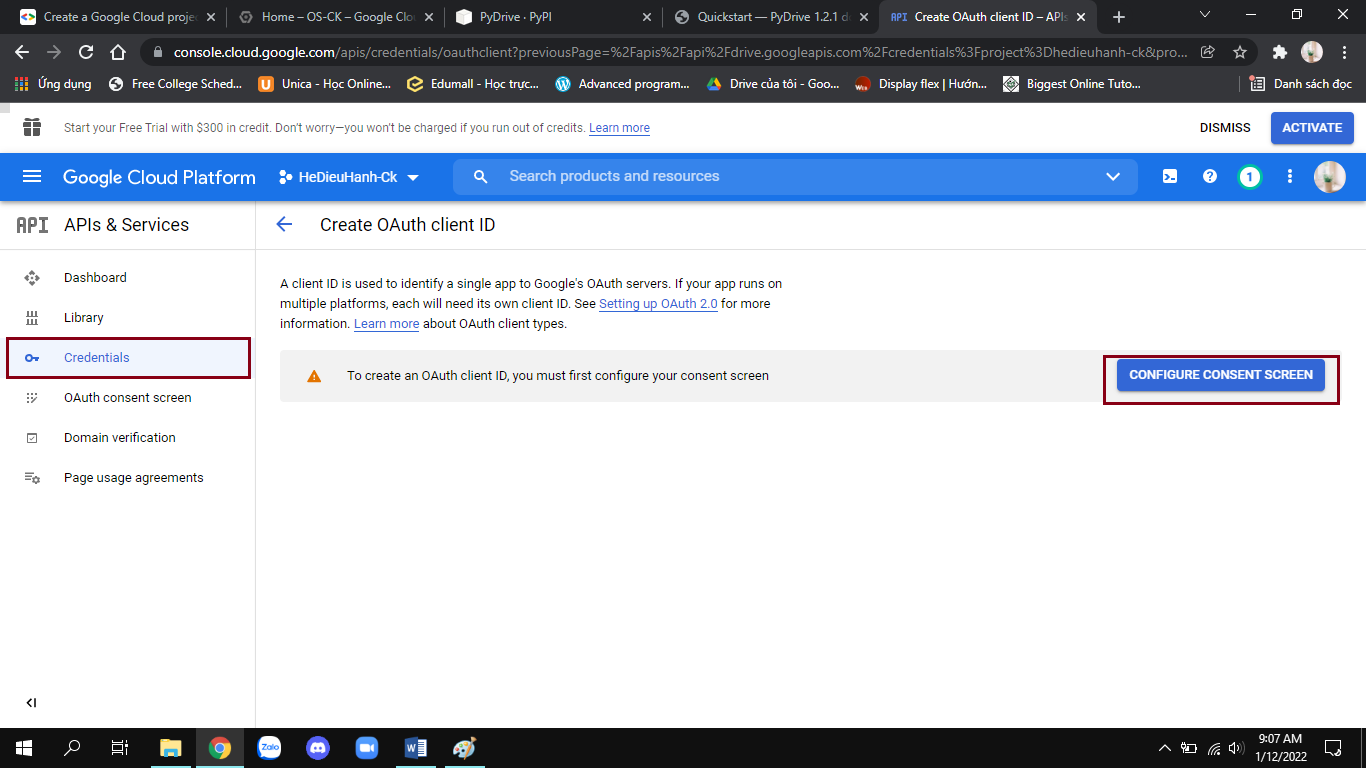
-Nhập key search “Google drive api” và thanh tìm kiếm và sau đó nhấn Enter.



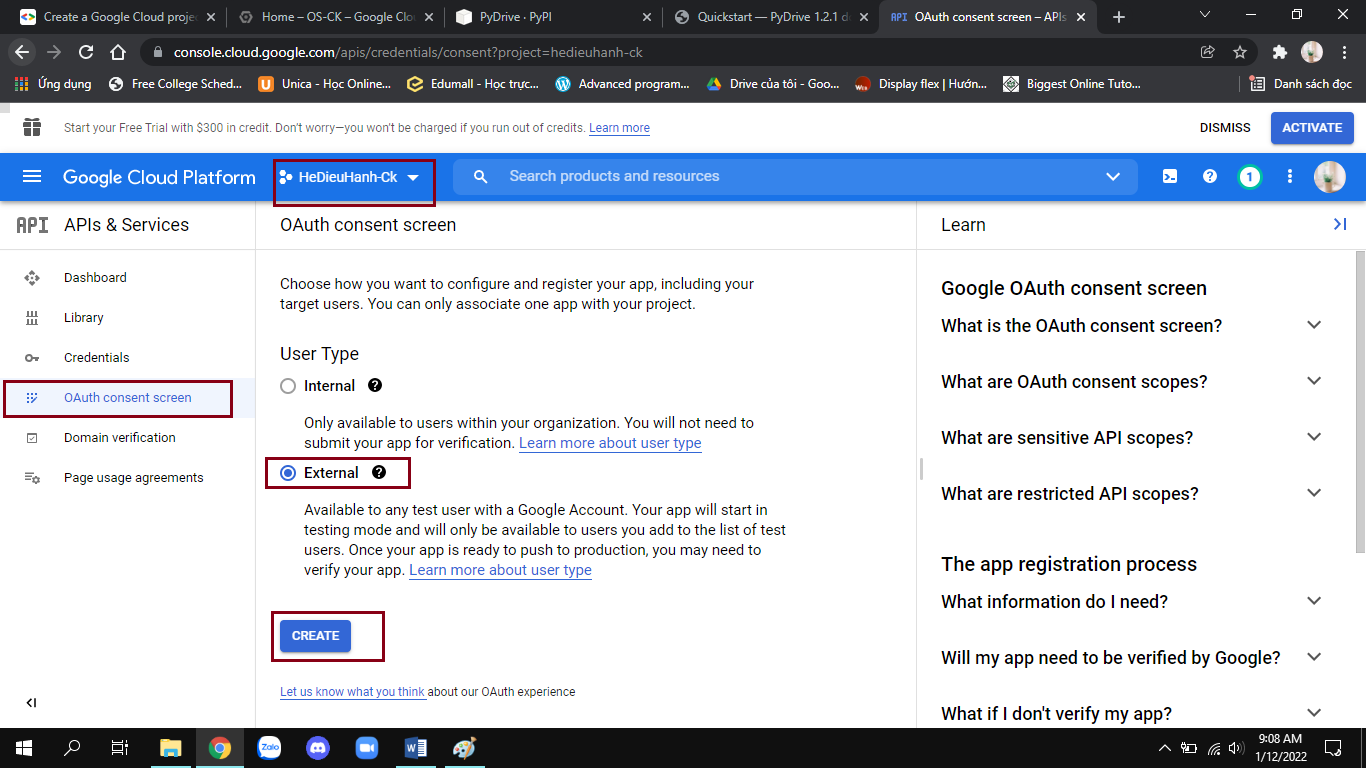
-Nhấn chọn “Google drive api”, chọn Project vừa tạo để kích hoạt dịch vụ của Google Drive api và sau đó nhấn vào “Enable” để thực hiện kích hoạt.



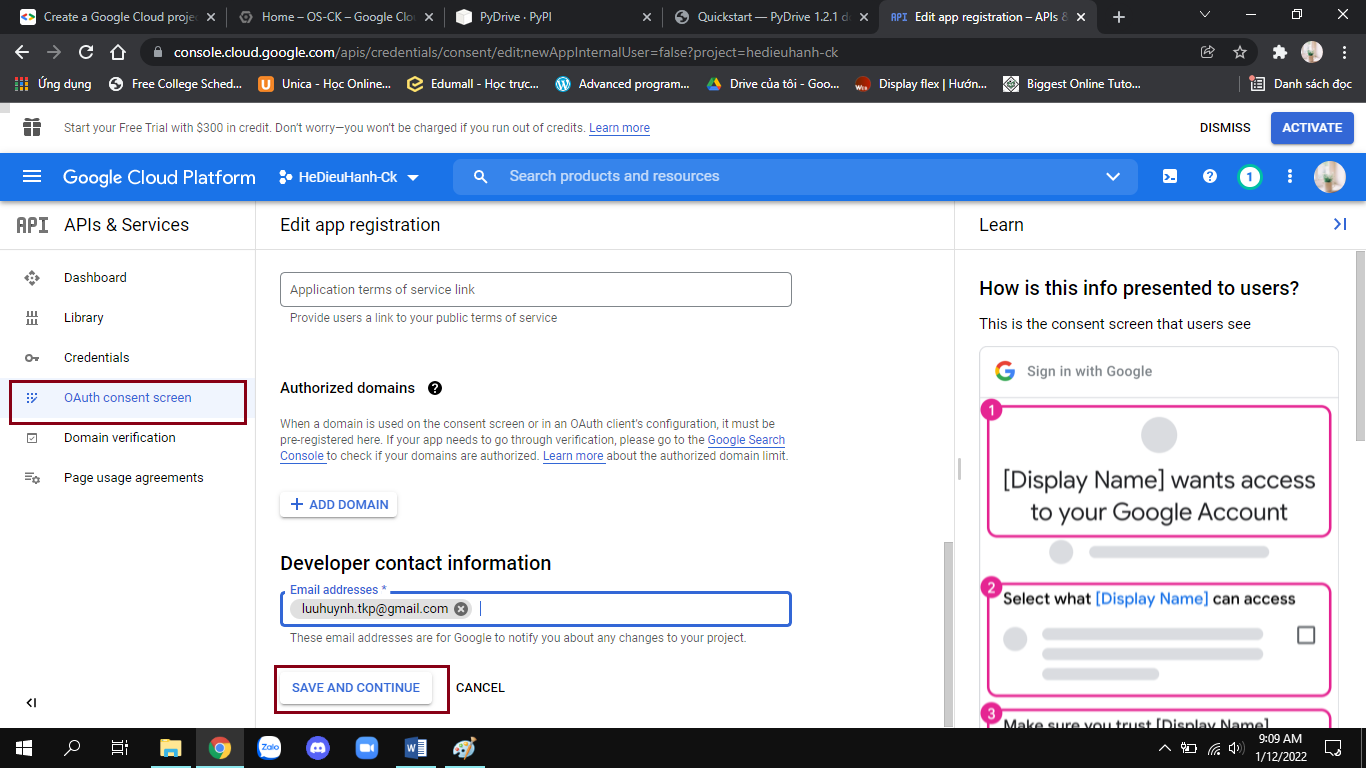
-Chọn “Credentials” -> “Configure consent screen”.



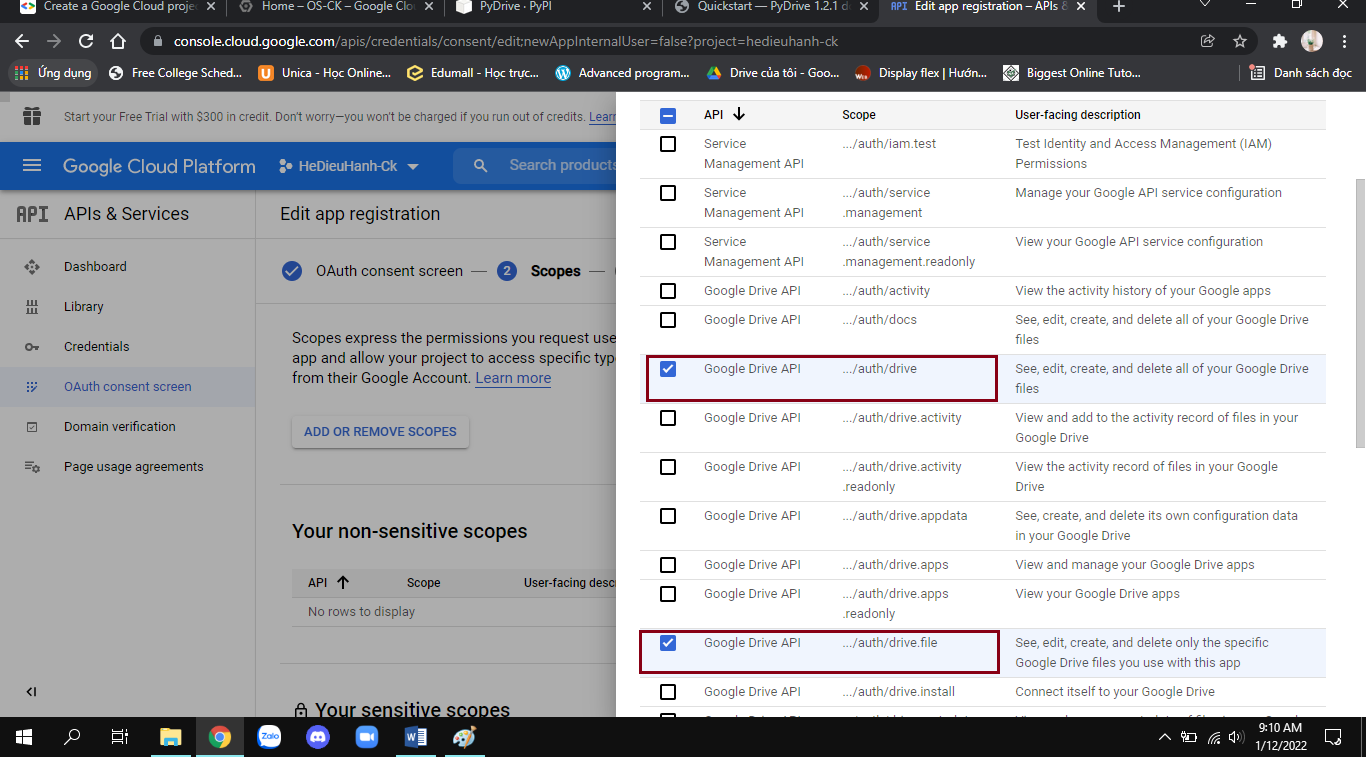
-Chọn User Type là External, sau đó nhấn chọn “Create”.



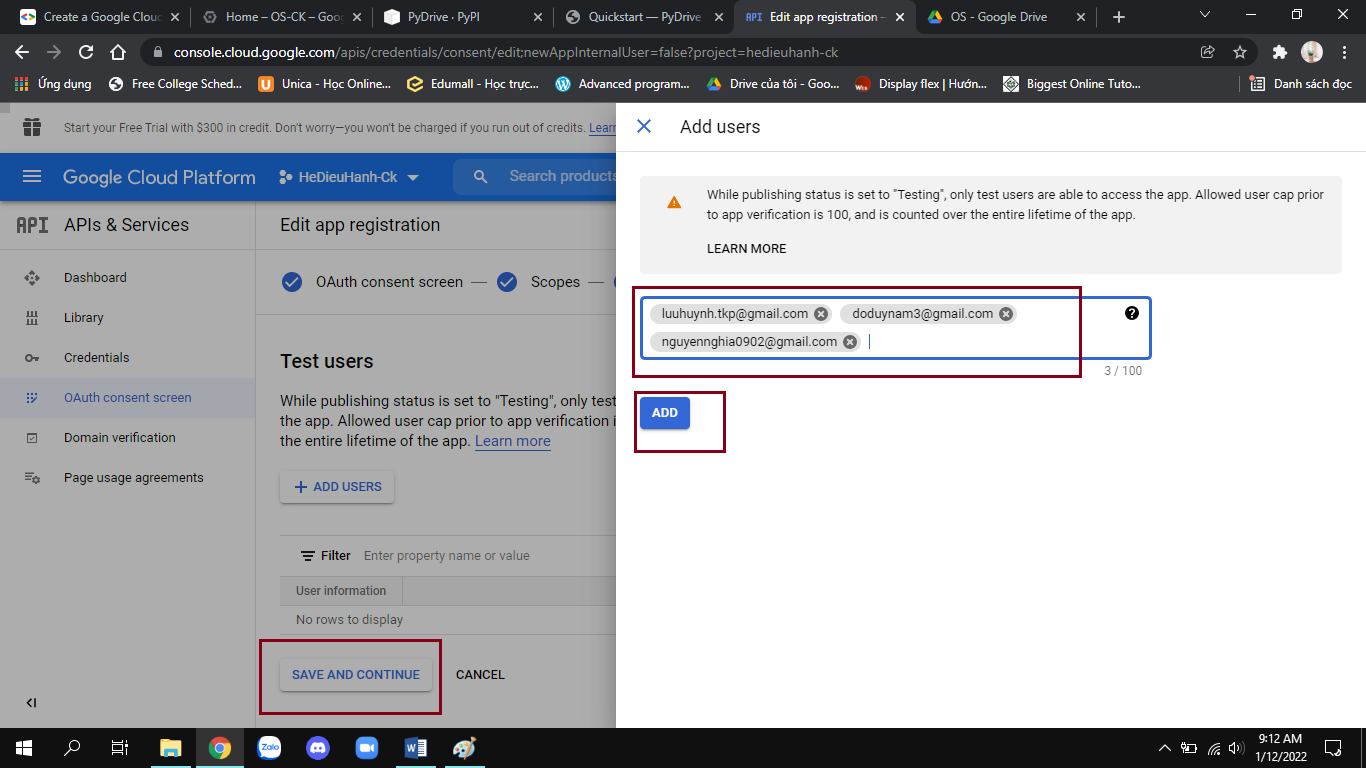
-Nhập các trường dữ liệu cho App information, sau đó nhấn chọn “Save and Continue”.



-Chọn “Add or remove scopes”, chọn các scope như mong muốn, sau đó nhấn chọn “Save and continue”.



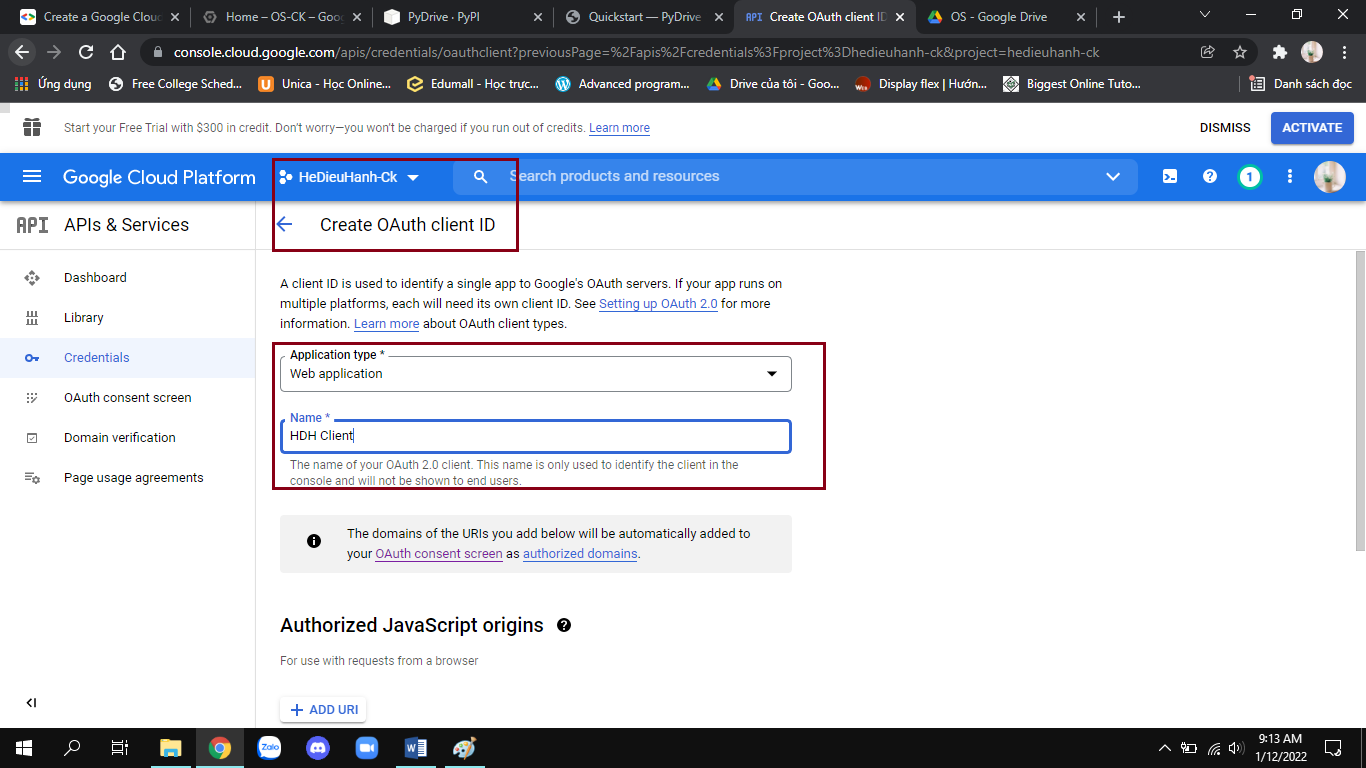
-Chọn “Add User” -> Nhập các email của user, sau đó nhấn chọn “Save and Continue”.



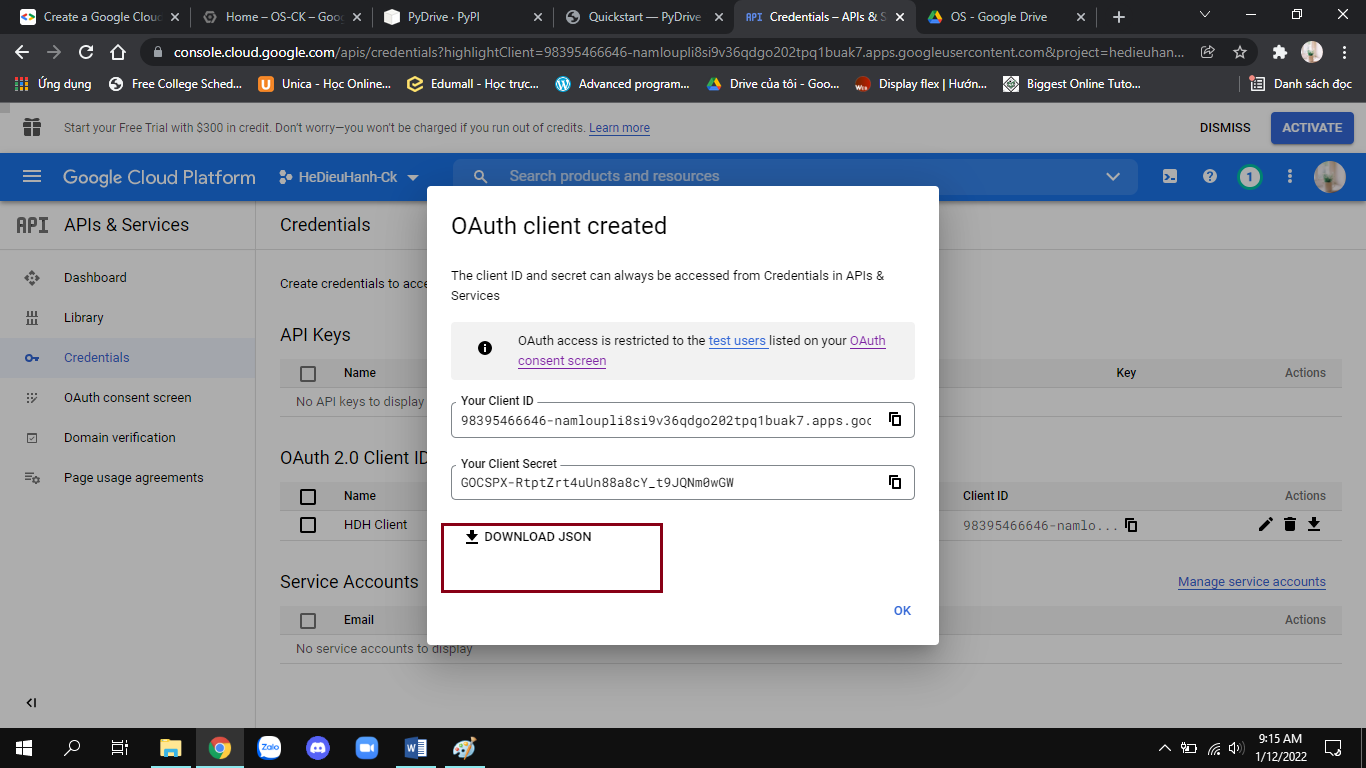
-Trở về Dashboard, chọn “Create Credentials” -> chọn “Oauth client ID”. Sau đó cung cấp các trường dữ liệu tương ứng:

* Application type: Web application
* Name: HDH Client
* Authorized JavaScript origins: <http://localhost:8080>
* Authorized redirect URIs: <http://localhost:8080/>

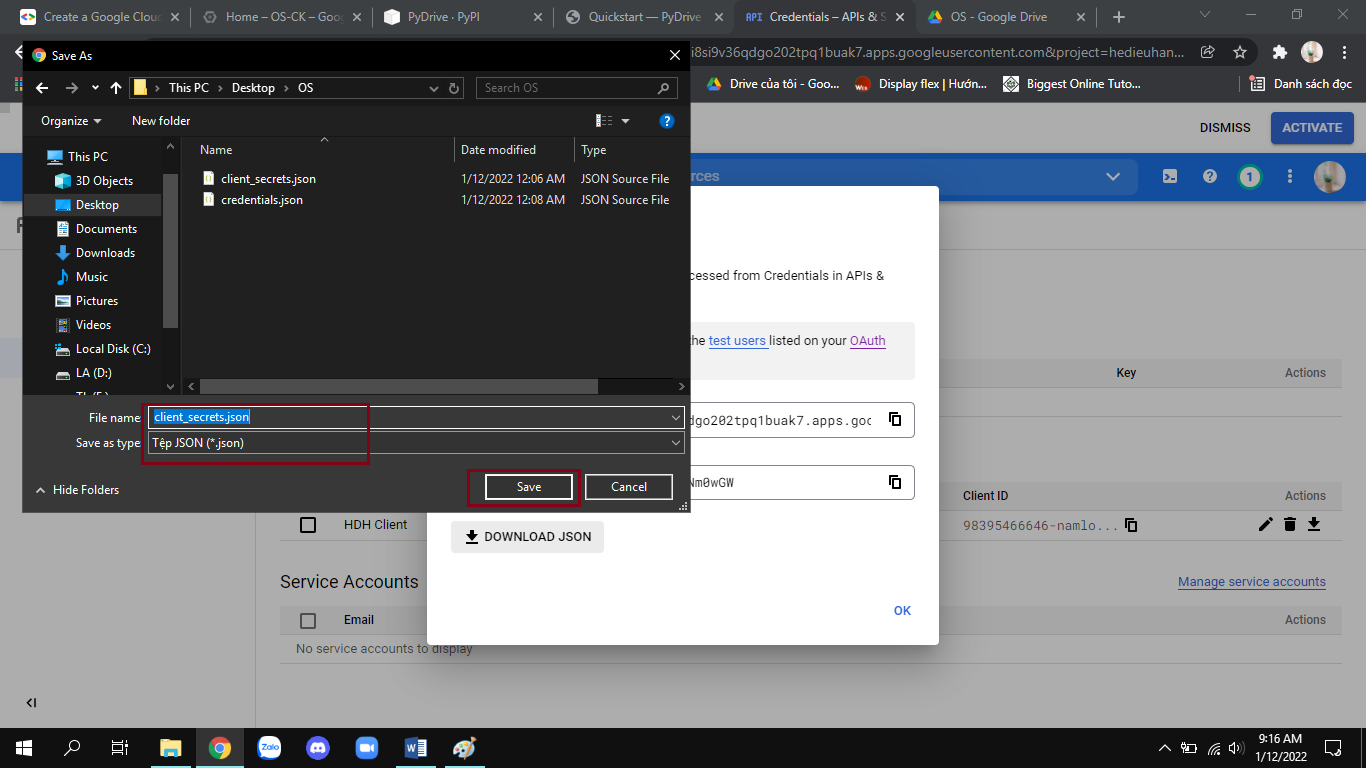
-Cuối cùng, nhấn chọn “Create”.



-Nhấn chọn “Download json”.



-Chọn vị trí lưu file json là folder chưa source code, rename file json thành “client\_secrets.json” và nhấn chọn “Save”.



## 3.Trình bày về semaphore.

Semaphore là phương pháp thoả mãn 4 điều kiện của một phương pháp giải quyết tốt bài toán đồng bộ:

* Mutual Exclusion: Không có hai tiến trình cùng ở trong miền găng cùng lúc.
* Progress: Một tiến trình tạm dừng ở bên ngoài miền găng không được ngăn cản các tiến trình khác vào miền găng.
* Bound Waiting: Không có tiến trình nào chờ vô hạn để được vào miền găng (Deadlock).
* Không có giả thiết nào đặt ra cho sự liên hệ về tốc độ của các tiến trình,, cũng như về số lượng bộ xử lý trong hệ thống.

**Cấu trúc semaphore.**

Semaphore gồm 2 thao tác sau:

-Thao tác Down(s): giảm giá trị của semaphore s xuống 1 đơn vị nếu semaphore có giá trị lớn hơn 0, và tiếp tục xử lý. Ngược lại, nếu giá trị của s đang nhỏ hơn 0, tiến trình phải chờ đến khi giá trị của s lớn hơn 0.

-Thao tác Up(s): tăng giá trị của semaphore lên 1 đơn vị. Nếu có 1 hoặc nhiều tiến trình đang chờ trên semaphore s, thì hệ thống sẽ chọn 1 trong các tiến trình này để cho tiếp tục thực thi.

**Các bước giải quyết bài toán độc quyền truy xuất sử dụng semaphore.**

|  |
| --- |
| Down(s) |
| Miền găng (đoạn mã có khả năng gây race condition) |
| Up(s) |

Hai hàm Down và Up là các thao tác dùng cơ chế điều phối độc quyền (chiếm CPU đến khi hoàn thành xong mới trả lại CPU cho Hệ điều hành).

# PHẦN IV: Các trích dẫn tham khảo được sử dụng trong bài báo cáo