# khung doiTRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

# KHOA CÔNG NGHỆ THÔNG TIN

-----🙞🙜🕮🙞🙜-----



**BÀI TIỂU LUẬN**

**MÔN HỌC: NGUYÊN LÝ HỆ ĐIỀU HÀNH**

ĐỀ TÀI**:** Nghiên cứu tìm hiểu về Quản lý tiến trình trong

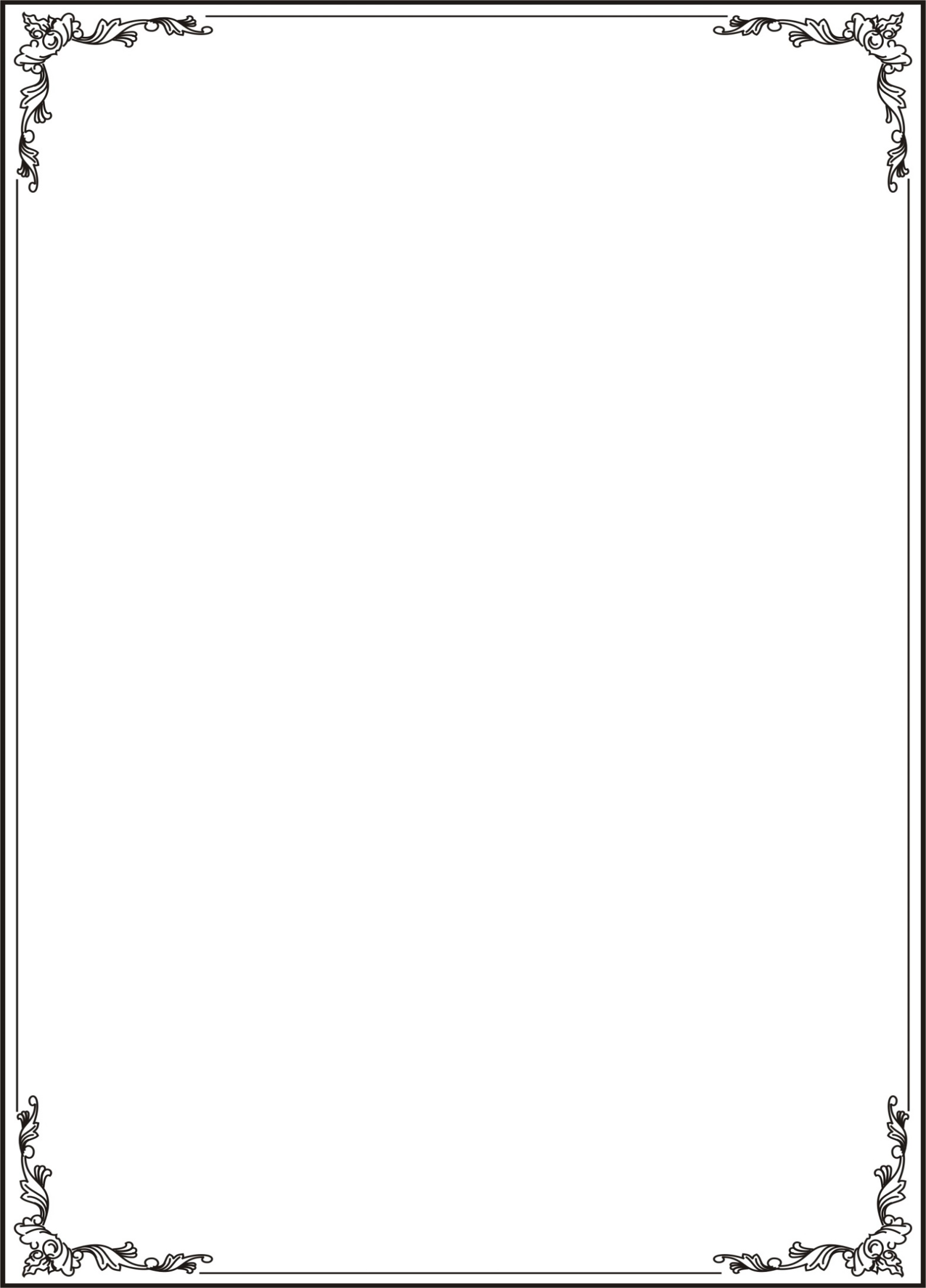
Hệ điều hành Windows

***Giáo viên hướng dẫn:* Ths. Nguyễn Thanh Hải**

***Lớp:* 20212IT6025008**

***Nhóm số:* 1**

**Hà Nội, 2022**



**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TIỂU LUẬN**

**MÔN HỌC: NGUYÊN LÝ HỆ ĐIỀU HÀNH**

ĐỀ TÀI**:** Nghiên cứu tìm hiểu về Quản lý tiến trình trong

Hệ điều hành Windows

***Giáo viên hướng dẫn:* Ths. Nguyễn Thanh Hải**

***Lớp:* 20212IT6025008**

***Nhóm thực hiện:***

**1.** **Phạm Minh Đức**

**2. Bùi Văn Thuận**

**3. Trương Quang Việt Anh**

**4. Nguyễn Văn Trúc**

**5. Trần Xuân Tùng**

**Hà Nội, 2022**

**MỤC LỤC**

[*LỜI NÓI ĐẦU 1*](#_Toc420298562)

[*CHƯƠNG 1: KHÁI NIỆM TIẾN TRÌNH VÀ CÁC VẤN ĐỀ LIÊN QUAN 2*](#_Toc420298563)

[*1. Tiến trình 2*](#_Toc420298565)

[*1.1. Khái niệm 2*](#_Toc420298566)

[*1.2. Các loại tiến trình 2*](#_Toc420298567)

[*1.3. Các trạng thái của tiến trình 2*](#_Toc420298568)

[*1.4. Quan hệ giữa các tiến trình 3*](#_Toc420298569)

[*2. Tài nguyên “Găng” 3*](#_Toc420298570)

[*2.1. Khái niệm 3*](#_Toc420298571)

[*2.2. Mục tiêu của quản lý tiến trình 4*](#_Toc420298572)

[*2.3. Nguyên tắc chung của các phương pháp giải quyết bài toán đoạn tới hạn 4*](#_Toc420298573)

*2.3.1. Phương pháp khóa trong................................................................4*

*2.3.2. Phương pháp kiểm tra và xác lập...................................................4*

*2.3.3. Phương pháp đèn hiệu....................................................................5*

*2.3.4. Phương pháp dùng trình thư kí.......................................................5*

*2.3.5. Phương pháp tổ chức liên lạc giữa các tiến trình.........................5*

[*3. Hiện tượng bế tắc 5*](#_Toc420298574)

[*3.1. Khái niệm bế tắc 5*](#_Toc420298575)

[*3.2. Điều kiện xảy ra bế tắc trong hệ thống 6*](#_Toc420298576)

[*3.3. Các mức phòng tránh bế tắc 6*](#_Toc420298577)

[*3.3.1. Ngăn ngừa bế tắc 6*](#_Toc420298578)

[*3.3.2. Dự báo và tránh bế tắc 6*](#_Toc420298579)

[*3.3.3. Xử lý bế tắc 7*](#_Toc420298580)

[*CHƯƠNG 2: QUẢN LÝ TIẾN TRÌNH VỚI TASK MANAGER( WINDOWS 10) 8*](#_Toc420298581)

[*1. Giới thiệu chung 8*](#_Toc420298583)

[*2. Chức năng của các tabs trong Task Manager 9*](#_Toc420298584)

[*2.1. Tab Processes( tab tiến trình) 9*](#_Toc420298585)

[*2.2. Tab Performance( Tab hiệu suất) 13*](#_Toc420298586)

[*2.3. Tab App History 17*](#_Toc420298587)

[*2.4. Tab Startup( Tab Khởi động) 18*](#_Toc420298588)

[*2.5. Tab Users( Tab người dùng) 19*](#_Toc420298589)

[*2.6. Tab Details( Tab chi tiết) 20*](#_Toc420298590)

[*2.7. Tab Services 21*](#_Toc420298591)

[*TÀI LIỆU THAM KHẢO 22*](#_Toc420298592)

# LỜI NÓI ĐẦU

Tất cả các hệ điều hành, từ các hệ điều hành đơn người sử dụng đến các hệ điều hành có thể hỗ trợ đến hàng ngàn người sử dụng, đều phải xây dựng dựa trên khái niệm tiến trình. Vì thế, một yêu cầu quan trọng trong thiết kế hệ điều hành là thành phần quản lý tiến trình của hệ điều hành phải đáp ứng tất cả những gì liên quan đến tiến trình:

* Hệ điều hành phải cho phép thực hiện nhiều tiến trình đồng thời để khai thác tối đa thời gian xử lý của processor nhưng cũng cung cấp được thời gian hồi đáp hợp lý.
* Hệ điều hành phải cấp phát tài nguyên để tiến trình hoạt động một cách hiệu quả với một chính sách hợp lý nhưng không xảy ra tình trạng tắc nghẽn trong hệ thống.
* Hệ điều hành phải có nhiệm vụ tạo ra tiến trình, điều khiển sự hoạt động của tiến trình và kết thúc tiến trình.

Mỗi hệ điều hành có những đặc trưng riêng trong sử dụng thuật toán cũng như xây dựng phần mềm quản lý tiến trình. Qua đề tài này, nhóm chúng em sẽ tập trung đề cập đến kiến thức cơ bản về quản lý tiến trình trong Hệ điều hành Windows. Với lượng kiến thức không quá lớn cùng với hình ảnh minh họa cụ thể, sinh động, nhóm chúng em sẽ mang đến cho độc giả cái nhìn khái quát nhất về quản lý tiến trình trong Hệ điều hành Windows mà cụ thể hơn là trình quản lý Task Manager.

Bài tiểu luận gồm 2 chương chính:

* ***Chương 1:*** Khái niệm tiến trình và các vấn đề liên quan: Giúp độc giả có kiến thức cơ bản nhất, tổng quát nhất về tiến trình, quá trình tạo ra 1 tiến trình, quan hệ giữa các tiến trình và 1 số khái niệm cơ bản liên quan khác.
* ***Chương 2:*** Quản lý tiến trình với Task Manager( Windows 10): Giới thiệu chi tiết các tab, cách sử dụng trong Task Manager và từ đó hiểu được cách mà Windows quản lý tiến trình các tiến trình của nó.

Sau cùng nhóm chúng em xin gửi lời cảm ơn trân thành Thầy Nguyễn Thanh Hải đã giúp chúng em rất nhiều trong việc hoàn thiện nội dung đề tài này. Bài tiểu luận do nhóm thực hiện sẽ khó tránh khỏi những sai thiếu sót, nhóm chúng em rất mong muốn nhận được những đóng góp ý kiến của bạn đọc và quý thầy cô để bài tiểu luận được hoàn thiện và chính xác hơn.   
 *Trân trọng cảm ơn !*

***Nhóm 1***

CHƯƠNG 1: KHÁI NIỆM TIẾN TRÌNH VÀ

CÁC VẤN ĐỀ LIÊN QUAN

* 1. **Tiến trình**
  2. **Khái niệm**

Là một chương trình đang xử lý, nó sở hữu một con trỏ lệnh, tập các thanh ghi và các biến. Để hoàn thành nhiệm vụ của mình, các tiến trình có thể còn yêu cầu một số tài nguyên hệ thống như: CPU, bộ nhớ và các thiết bị.

* 1. **Các loại tiến trình**

Các tiến trình trong hệ thống có thể chia thành 2 loại: Tiến trình tuần tự và tiến trình song song.

* *Tiến trình tuần tự*: là các tiến trình mà điểm khởi tạo của nó là điểm kết thúc của tiến trình trước đó.
* *Tiến trình song song:* là các tiến trình mà điểm khởi tạo của tiến trình này nằm ở thân của các tiến trình khác, tức là có thể khởi tạo một tiến trình mới khi các tiến trình trước đó chưa kết thúc. Tiến trình song song được chia thành nhiều loại:
* *Tiến trình song song độc lập*
* *Tiến trình song song có quan hệ thông tin*
* *Tiến trình song song phân cấp*
* *Tiến trình song song đồng mức*
  1. **Các trạng thái của tiến trình**

Trạng thái của tiến trình tại mỗi thời điểm được xác định bởi hoạt động hiện thời của tiến trình tại thời điểm đó. Trong suốt khoảng thời gian tồn tại trong hệ thống, một tiến trình có thể thay đổi trạng thái do rất nhiều nguyên nhân như: chờ đợi sự kiện nào đó xảy ra, đợi một thao tác vào/ra hoàn tất, hết thời gian xử lí...

Tại mỗi thời điểm, Tiến trình có thể nhận một trong các trạng thái sau:

* *Trạng thái khởi tạo( New)*: tiến trình đang được tạo lập.
* *Trạng thái sẵn sàng( Ready)*: tiến trình chờ được cấp phát CPU để xử lý.
* *Trạng thái thực hiện( Running)*: tiến trình được xử lý.
* *Trạng thái đợi( Waiting):* tiến trình phải dừng vì thiếu tài nguyên hoặc chờ 1 sự kiện nào đó.
* *Trạng thái kết thúc( Halt)*: tiến trình đã hoàn tất công việc xử lý.

Diagram

Description automatically generated

*Hình 1*. Các trạng thái của một tiến trình.

* 1. **Quan hệ giữa các tiến trình**

Các tiến trình hoạt động trong hệ thông tồn tại 2 mối quan hệ: Độc lập và hợp tác( Song hành).

* + - **Quan hệ độc lập**

Tiến trình được gọi là độc lập nếu hoạt động của nó không gây ảnh hưởng và không bị ảnh hưởng bởi các tiến trình khác cũng đang hoạt động của hệ thống. Tiến trình độc lập có những đặc trưng sau:

* Trạng thái của nó không bị chia sẻ với bất kì tiến trình nào khác.
* Việc thực hiện tiến trình là đơn định (kết quả chỉ phụ thuộc vào đầu vào).
* Tiến trình có thể tái hiện (lặp lại).
* Tiến trình có thể dừng hoặc bắt đầu lại mà không gây ảnh hưởng tới các tiến trình khác trong hệ thống.
  + - **Quan hệ hợp tác**

Tiến trình được gọi là hợp tác (song hành) nếu hoạt động của nó gây ảnh hưởng hoặc bị ảnh hưởng bởi các tiến trình khác cũng đang hoạt động trong hệ thống. Tiến trình hợp tác có những đặp trưng sau:

* Trạng thái của nó bị chia sẻ cho các tiến trình khác.
* Việc thực hiện tiến trình là không đơn định (kết quả phụ thuộc dãy thực hiện tương ứng và không dự báo trước).
* Tiến trình không thể tái hiện.

1. **Tài nguyên “Găng”**
   1. **Khái niệm**

* Các tài nguyên logic và vật lí phân bổ cho các tiến trình song hành là tài nguyên găng.
* Tài nguyên găng có thể là tài nguyên phần cứng hoặc tài nguyên phần mềm, có thể là tài nguyên phân chia được hoặc không phân chia được, nhưng đa số thường là tài nguyên phân chia được như là: các biến chung, các file chia sẻ.
* Các đoạn chương trình sử dụng tài nguyên găng được gọi là đoạn tới hạn (Critical

Section)

* 1. **Mục tiêu của quản lý tiến trình**
* Hạn chế tối đa xung đột và bế tắc xảy ra, đưa ra giải pháp nếu xảy ra các tình huống đó.
* Tận dụng tối đa khả năng của CPU (bài toán lập lịch)
* Điều kiện loại trừ lẫn nhau: tại mỗi thời điểm, chỉ có một tiến trình được phép thực hiện trong đoạn tới hạn của mình.
* Điều kiện tiến triển: không tiến trình nào được phép ở lâu vô hạn trong đoạn tới hạn của mình.
* Điều kiện chờ đợi có giới hạn: các tiến trình không phải chờ đợi vô hạn trước khi đi vào đoạn tới hạn của mình.
  1. **Nguyên tắc chung của các phương pháp giải quyết bài toán đoạn tới hạn**
     1. **Phương pháp khóa trong**
* Phương pháp này dựa trên cơ sở nếu hai hay nhiều tiến trình cùng định ghi vào một địa chỉ nào đó của bộ nhớ trong, thì giải thuật chỉ cho phép một tiến trình được thực hiện còn các tiến trình khác phải chờ.
* Mỗi tiến trình sử dụng một byte trong bộ nhớ làm khóa.
* Khi tiến trình vào đoạn tới hạn, byte khóa của nó được gán giá trị = 1 để thông báo cho các tiến trình còn lại biết tài nguyên găng đã được sử dụng.
* Khi ra khỏi đoạn tới hạn, byte khóa được gán giá trị = 0 để thông báo tài nguyên găng đã được giải phóng.
* Trước khi vào đoạn tới hạn các tiến trình phải kiểm tra byte khóa của các tiến trình khác. Nếu có byte chứa giá trị =1 thì tiến trình phải chờ cho tới khi byte đó nhận giá trị = 0.
  + 1. **Phương pháp kiểm tra và xác lập**
* Phương pháp này dựa vào sự hỗ trợ của phần cứng, có một lệnh thực hiện hai phép xử lí liên tục không bị tách rời.
* Giả sử ta gọi lệnh đó là TS (Test and Set), lệch này có 2 tham số: biến chung G là biến riêng L (biến chung G thông thường là một bit trong từ trạng thái hoặc trong thanh ghi cờ). Dạng thức thực hiện của lệnh TS(L) như sau:

L := G (gán giá trị biến chung cho biến riêng)

G := 1 (gán giá trị 1 cho biến chung)

* + 1. **Phương pháp đèn hiệu**
* Đèn hiệu S là một biến nguyên mà chỉ có hai phép xử lí WAIT và SINGNAL mới thay đổi được giá trị của nó.
* *Định nghĩa phép WAIT(S):*

S := S – 1;

Nếu S >= 0 thì tiếp tục thực hiện tiến trình;

Nếu S < 0 thì đưa tiến trình vào hàng đợi.

* *Định nghĩa phép SIGNAL(S):*

S := S + 1;

Nếu S <= 0 thì đưa tiến trình trong hàng đợi vào đoạn tới hạn.

* Chú ý:
* Phép WAIT và SIGNAL không bị phân chia trong tiến trình thực hiện.
* Nếu ban đầu S = 1 và cả hai tiến trình đều đưa ra phép WAIT(S) thì chỉ có một tiến trình được phép vào đoạn tới hạn, tiến trình còn lại sẽ được đưa vào hàng đợi.
  + 1. **Phương pháp dùng trình thư kí**
* Trình thư kí (monitor) là bộ các thủ tục và cấu trúc thông tin động, hoạt động trong chế độ phân chia thời gian, hỗ trợ việc thực hiện tiến trình. Mỗi thời điểm chỉ có một tiến trình làm việc được với monitor.

**2.3.5. Phương pháp tổ chức liên lạc giữa các tiến trình**

* Hệ điều hành xây dựng một hệ thông báo giữa các tiến trình dựa trên 3 thao tác cơ bản: gửi thông báo, nhận thông báo, tạo móc nối liên lạc giữa các tiến trình.
* Dựa vào hệ thông báo này, các tiến trình có thể phối hợp để vào đoạn tới hạn bằng cách trao đổi thông báo với nhau. Hệ thống sử dụng hai cách thức liên lạc: trực tiếp (bằng cách sửa hai thao tác gửi và nhận) và gián tiếp (dùng hộp thư trung gian).

1. **Hiện tượng bế tắc**
   1. **Khái niệm bế tắc**

Bế tắc là trạng thái khi hai hoặc nhiều tiến trình cùng chờ đợi một số sự kiện nào đó và nếu không có tác động đặc biệt từ bên ngoài thì sự chờ đợi đó là vô hạn.

* 1. **Điều kiện xảy ra bế tắc trong hệ thống**

Hiện tượng bế tắc xảy ra khi và chỉ khi trong hệ thống tồn tại bốn điều kiện:

* Có tài nguyên găng
* Có hiện tượng giữ và đợi: có một tiến trình đag giữ một số tài nguyên và đợi tài nguyên bổ sung đang được giữ bởi các tiến trình khác.
* Không có hệ thống phân phối lại tài nguyên: việc sử dụng tài nguyên không bị ngắt.
* Có hiện tượng chờ đợi vòng tròn.
  1. **Các mức phòng tránh bế tắc**

Để tránh hiện tượng bế tắc, thông thường hệ thống áp dụng ba mức:

* Ngăn ngừa: áp dụng các biện pháp để hệ thống không rơi vào bế tắc.
* Dự báo và tránh bế tắc: áp dụng các biện pháp để kiểm tra các tiến trình có rơi và trạng thái bế tắc hay không. Nếu có thì thông báo trước khi bế tắc xảy ra.
* Nhận biết và khắc phục: tìm cách phát hiện và giải quyết.
  + 1. **Ngăn ngừa bế tắc**

Để ngăn chặn thì làm cho hệ thống không xảy ra đồng thời 4 điều kiện bế tắc:

- Loại bỏ tài nguyên găng: mô phỏng tài nguyên găng bằng các tài nguyên có thể dùng chung được và áp dụng kỹ thuật SPOOL.

- Loại bỏ yếu tố giữ và đợi: thực hiện phân bổ tài nguyên, tiến trình chỉ có thể thực hiện khi mọi tài nguyên mà nó yêu cầu đã được phân bổ đầy đủ. Tiến trình chỉ được phép đòi tài nguyên khi nó không giữ tài nguyên nào cả. Nếu tiến trình phải đợi thì mọi tài nguyên nó đang giữ tạm thời giải phóng.

- Xây dựng hệ thống ngắt tài nguyên:

+ Phương pháp 1: Nếu tiến trình đang giữ một số tài nguyên và yêu cầu tài nguyên bổ sung nhưng hệ thống không thể phân bố ngay được thì mọi tài nguyên tiến trình đang giữ sẽ bị ngắt và được bổ sung vào danh sách các tài nguyên tự do. Tiến trình sẽ được bắt đầu lại khi nó được phân bổ đầy đủ các tài nguyên cần thiết.

+ Phương pháp 2: Nếu tiến trình đang giữ một số tài nguyên và yêu cầu tài nguyên bổ sung nhưng hệ thống không thể phân bổ ngay, khi đó hệ thống sẽ kiểm tra xem liệu tài nguyên mà tiến trình yêu cầu có bị giữ bởi các tiến trình khác cũng đang đợi hay không. Nếu có thì ngắt các tiến trình này, thu hồi lại tài nguyên để phân bổ cho tiến trình yêu cầu; ngược lại thực hiện như phương pháp 1.

* + 1. **Dự báo và tránh bế tắc**

Nguyên tắc chung là mỗi lần phân bổ tài nguyên cho các tiến trình thì hệ thống sẽ kiểm tra xem việc phân bổ có đẩy hệ thống và tình trạng bế tắc hay không. Nếu có thì tìm ra cách giải quyết trước khi bế tắc xảy ra.

Hệ điều hành sử dụng 2 khái niệm là trạng thái và trạng thái an toàn để tránh được bế tắc trong công tác phòng chống bế tắc:

* Trạng thái của hệ thống: là sự cấp phát tài nguyên hiện tại cho các tiến trình.
* Trạng thái an toàn: Trạng thái A là an toàn nếu hệ thống có thể thỏa mãn các nhu cầu tài nguyên (đến tối đa) của mỗi tiến trình mà vẫn ngăn chặn được bế tắc.
  + 1. **Xử lý bế tắc**

Khi đã phát hiện được bế tắc, HĐH có thể áp dụng các phương pháp giải quyết:

* Thông báo cho Operator biết để tự xử lý.
* Đình chỉ hoạt động của các tiến trình có liên quan, thu hồi tất cả các tài nguyên bị kết thúc và sử dụng 2 phương pháp sau:
* Đình chỉ hoạt động của mọi tiến trình trong tình trạng bế tắc.
* Đình chỉ hoạt động lần lượt của từng tiến trình cho tới khi thoát khỏi tình trạng bế tắc( Khi đình chỉ hoạt động của tiến trình nào thì thu hồi tài nguyên của tiến trình đó).

CHƯƠNG 2: QUẢN LÝ TIẾN TRÌNH VỚI

TASK MANAGER( WINDOWS 10)

1. **Giới thiệu chung**

***Windows Task Manager*** là ứng dụng quản lý tiến trình kèm theo trong họ hệ điều hành [Microsoft Windows](http://vi.wikipedia.org/wiki/Microsoft_Windows). Nó cung cấp thông tin đầy đủ về hiệu năng [hệ thống](http://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng), các [ứng dụng](http://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_%E1%BB%A9ng_d%E1%BB%A5ng) đang chạy, các tiến trình và sự phân chia [CPU](http://vi.wikipedia.org/wiki/CPU), thông tin về bộ nhớ, sự hoạt động và thống kê mạng, người dùng đăng nhập và các [dịch vụ hệ thống](http://vi.wikipedia.org/w/index.php?title=D%E1%BB%8Bch_v%E1%BB%A5_windows&action=edit&redlink=1).Task manager cũng có thể dùng để thiết lập mức độ ưu tiên cho các tiến trình, chọn nhân CPU cho phép chạy một ứng dụng chỉ định, buộc dừng một tiến trình và tắt máy, khởi động lại, ngủ đông, đăng xuất. Windows Task Manager được giới thiệu với [Windows NT 4.0](http://vi.wikipedia.org/w/index.php?title=Windows_NT_4.0&action=edit&redlink=1). Những phiên bản trước đó của Windows NT thì có ứng dụng [Task List](http://vi.wikipedia.org/w/index.php?title=Task_List&action=edit&redlink=1), với một vài chức năng như hiện danh sách các tiến trình, ngắt hoặc tạo một tiến trình mới.

Graphical user interface, table

Description automatically generated

*Hình 2. Task Manager trên Windows 10*

***Task Manager*** có thể gọi bằng một trong bốn cách sau:

* Nháy phải chuột vào [taskbar](http://vi.wikipedia.org/w/index.php?title=Taskbar&action=edit&redlink=1) và chọn "Task Manager"
* Dùng tổ hợp phím Ctrl+Shift+Esc.
* Trong [Windows NT](http://vi.wikipedia.org/w/index.php?title=Windows_NT&action=edit&redlink=1), [Windows 2000](http://vi.wikipedia.org/wiki/Windows_2000), and [Windows Vista](http://vi.wikipedia.org/wiki/Windows_Vista), dùng tổ hợp phím [Ctrl+Alt+Del](http://vi.wikipedia.org/w/index.php?title=Control-Alt-Delete&action=edit&redlink=1) để mở hộp thoại *Windows Security*, sau đó chọn "Task Manager". Ở [Windows](http://vi.wikipedia.org/wiki/Windows_XP) XP, nhấn trực tiếp Ctrl+Alt+Del hoặc Ctrl+Shift+Esc để gọi Task Manager, trừ khi bạn tắt Welcome screen.
* Gọi lệnh "Taskmgr.exe" từ [command line](http://vi.wikipedia.org/w/index.php?title=Command_line&action=edit&redlink=1), [GUI](http://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_ng%C6%B0%E1%BB%9Di_d%C3%B9ng_%C4%91%E1%BB%93_h%E1%BB%8Da) hay là shortcut.

1. **Chức năng của các tabs trong Task Manager**
   1. **Tab Processes( tab tiến trình)**

Đây là tab quản lý chung các tiến trình. Gồm các Apps đang chạy, các tiến trình chạy nền và các tiến trình của Windows. Nếu muốn biết thông tin về một Apps nào đó, click chuột phải vào Apps đó chọn Expand, ví dụ Google Chrome đang duyệt trang web nào….

Graphical user interface, application, table

Description automatically generated

*Hình 3. Expand Proccesses*

Khi gặp hiện tượng tiến trình đang bị bế tắc( Not responding) mà không muốn chờ hệ thống tự xử lý, người dùng có thể làm theo một trong 2 cách sau:

* Đình chỉ hoạt động của mọi tiến trình trong tình trạng bế tắc: người dùng chỉ cần chọn các tiến trình bị bế tắc và nhấn End Task.
* Đình chỉ hoạt động lần lượt của tiến trình cho tới khi thoát khỏi tình trạng bế tắc: người dùng chọn tiến trình bị bế tắc và nhấn End Task lần lượt từng tiến trình cho tới khi hệ thống hết bế tắc thì dừng lại.

Ngoài ra, trong trường hợp bế tắc không xảy ra mà người dùng vẫn muốn kết thúc 1 tiến trình nào đó thì vẫn có thể dùng chức năng End Task để kết thúc tiến trình.

Tuy nhiên, nếu máy tính được sử dụng với nhiều user khác nhau thì 1 tiến trình chỉ được kết thúc bởi user tạo ra tiến trình đó, các user khác sẽ không có quyền can thiệp đến việc kết thúc tiến trình đó.

Một số tiến trình chạy nền, tiến trình hệ thống… không thể tắt bằng chức năng End Task. Khi tắt các tiến trình này có thể gây ảnh hưởng tới hoạt động của hệ thống hoặc tiến trình khác.

Graphical user interface, application, table

Description automatically generated

*Hình 4. End Task Proccesses*

Ngoài ra, tab này còn hiển thị cho người dùng thấy rằng các tiến trình đang chạy đó tốn hết bao nhiêu tài nguyên máy (CPU, RAM, Disk, Network). Vì vậy nếu thấy máy tính chậm chạp bất thường, người dùng có thể kiểm tra các tiến trình trong đây và đưa ra giải pháp hợp lý nhất.

Graphical user interface, table

Description automatically generated

*Hình 5. Index Proccesses*

***Background proccess:*** là các proccess chạy dưới tầng nền giống như service, hầu hết các proccess này user có quyền End task

Graphical user interface, application, table

Description automatically generated

*Hình 6. Background Proccesses*

*Windows Proccesses****:*** Là các proccesses nền tảng của hệ thống, 1 số proccess user không có quyền. Tắt các proccesses này có thể ảnh hưởng tới hoạt động của hệ thống và các proccesses khác

**Graphical user interface, table

Description automatically generated with medium confidence**

*Hình 7. Windows Proccesses*

* 1. **Tab Performance( Tab hiệu suất)**

Tab này hiển thị thông tin tổng quan nhất về hiệu suất hoạt động của máy tính.

Graphical user interface, table

Description automatically generated

*Hình 8. Tab Performance*

Dưới đây là cái nhìn chi tiết về tab hiệu suất:

+ **CPU**

* Biểu đồ sử dụng CPU.
* Tổng số xử lý và hiệu suất các tiến trình đang chạy.
* Tốc độ bộ nhớ.
* Nếu CPU đạt 100% thì xảy ra hiện tượng bế tắc( cách xử lý đã nêu ở mục tab processes).

Graphical user interface, table

Description automatically generated

*Hình 9. CPU*

+ **Memory**

* Sử dụng đồ thị bộ nhớ
* Tổng số lượng bộ nhớ sử dụng và có sẵn
* Dung lượng bộ nhớ cache

Graphical user interface, table

Description automatically generated*Hình 10. Memory*

+ **Disk**

* Biểu đồ sử dụng đĩa
* Tốc độ đọc / ghi
* Thời gian hoạt động và thời gian phản ứng trung bình
* Dung lượng đĩa (size)

Table

Description automatically generated

*Hình 11. Disk*

+ **Ethernet**

* Tốc độ tải lên và tải về
* Thông tin kết nối (tên adapter, tên miền, kiểu kết nối, địa chỉ IP)
* Hình ảnh này cho thấy *Ethernet* tab. Máy có thêm kết nối (ví dụ như WiFi, Bluetooth) sẽ có các tab bổ sung.

Table

Description automatically generated

*Hình 12. Ethernet*

* 1. **Tab App History**

Tab này thống kê tổng tài nguyên mà các apps đã sử dụng trong 1 khoảng thời gian xác định. Nó giúp người dùng có cái nhìn tổng quan về các apps trong máy tính và tìm ra được các apps không dùng đến hoặc sử dụng quá nhiều tài nguyên của máy để có cách xử dụng hợp lý hơn.

Dưới đây là cái nhìn chi tiết về tab History:

* **CPU time**(tổng thời gian mà CPU phục vụ cho 1 tiến trình hoạt động)
* **Network**( tổng dung lượng dữ liệu download và upload qua mạng của tiến trình)
* **Metered Network**( đồng hồ đo dữ liệu mạng mà tiến trình đang sử dụng)
* **Tiles updates**( tổng dữ liệu gói).

Nếu click chuột phải vào một ứng dụng nào đó và chọn Switch To thì Windows sẽ tự động chuyển qua ứng dụng đó.

Table

Description automatically generated

*Hình 13. App History*

* 1. **Tab Startup( Tab Khởi động)**

Tab này dùng để quản lý các ứng dụng sẽ khởi động cùng Windows 10. Ở các phiên bản  Windows trước thì việc quản lý các ứng dụng tự khởi động trong thư mục Startup hoặc trong trang Msconfig. Nhưng với Windows 10, nó đã được tích hợp trong Task Manager.

* **Disable**: ngăn nó khởi chạy cùng Windows 10 ở những lần khởi động sau.
* **Open file location**: mở thư mục chứa phần mềm đó.
* **Search online**: tìm kiếm thông tin về nó trực tuyến.
* **Properties**: hiển thị thông tin ứng dụng.

Dưới đây là cái nhìn chi tiết về các tab Startup:

* ***Name:*** Tên tiến trình khởi động cùng hệ thống
* ***Pulisher:*** Nhà sản xuất ứng dụng
* ***Status:*** Trạng thái của ứng dụng( enable (cho phép) hoặc disable( vô hiệu hóa))
* ***Startup impact:*** Tác động của tiến trình đến tốc độ khởi động của các ứng dụng và được phân loại như sau:
* *High:* sử dụng hơn 1 s (1000 ms) của thời gian CPU
* *Medium:* sử dụng 300 - 1000 ms của thời gian CPU
* *Low:* sử dụng ít hơn 300 ms của thời gian CPU
  + *Not measured:* ứng dụng không chạy lúc khởi động

*Graphical user interface, application

Description automatically generatedHình 14. Startup*

* 1. **Tab Users( Tab người dùng)**

Tab này hiển thị cho người dùng biết có bao nhiêu tài khoản đang truy cập máy, mỗi tài khoản đó đang chạy những ứng dụng và tiến trình nào.

Tab user chứa các thẻ CPU (phần trăm CPU mà mỗi user sử dụng), Memory (bộ nhớ đang sử dụng của mỗi user), Disk (phần trăm bộ nhớ và tốc độ đọc/ghi của ổ đĩa cho mỗi user), Network (tốc độ sử dụng dữ liệu mạng).

Graphical user interface

Description automatically generated with medium confidence

*Hình 15. Users*

Ở trên ta có thể thấy có 1 user đang truy cập.

Người dùng cũng có thể chọn một user đã đăng nhập để Sign off (Đăng xuất) hoặc Switch user (Chuyển sang sử dụng) (Chú ý: Phải có nhiều hơn một người dùng đăng nhập trong Hệ thống).

Để xem chi tiết các tiến trình đang hoạt động của user, hãy click chuột phải vào user đó chọn Expand, sẽ có các chi tiết hiển thị cho mỗi tiến trình là CPU, Memory, Disk, Netwwork.

Table

Description automatically generated

*Hình 16. Users( details)*

* 1. **Tab Details( Tab chi tiết)**

Là một tính năng nâng cao của tab Process. Nó liệt kê tất cả các tiến trình đang chạy bao gồm các tiến trình của người sử dụng mở và các tiến trình chạy nền hoặc của Windows. Tab này tương tự với tab Process của Task Manager trong Windows 7.

* **Name:** Tên tiến trình
* **PID:** Mỗi tiến trình mang một định danh gọi là Process ID (PID). PID là một con số lớn hơn 0 và là duy nhất. Hệ thống sẽ dựa vào các PID này để quản lý các tiến trình.
* **Status:** Trạng thái của mỗi tiến trình( Running, Not responding, Suspended( bị đình chỉ)…)
* **User name:** Tên người dùng ứng với mỗi tiến trình( personal, administrator, system, network services...)
* **CPU:** tình trạng của CPU mà mỗi tiến trình sử dụng
* **Memory:** tình trạng bộ nhớ mà mỗi tiến trình sử dụng
* **Description:** Mô tả tên chương trình

Table

Description automatically generated

*Hình 17. Details*

* 1. **Tab Services**

Tab này giúp người dùng quản lý các dịch vụ chạy kèm với Windows 10. Các dịch vụ này giúp máy tính hoạt động ổn định. Nếu như không cần thiết hoặc không am hiểu nhiều về các Services của Windows thì người dùng không nên thay đổi gì nhiều trong tab này. Vì việc này có thể gây ra những vấn đề trong quá trình sử dụng Windows.

* **Name:** Tên service
* **PID:** Tương tự thẻ PID trong Tab Details
* **Description:** Mô tả về service
* **Status:** Trạng thái hoạt động của mỗi dịch vụ( Running hoặc Stopped)
* **Group:** Tên nhóm của mỗi dịch vụ

Graphical user interface, text, application

Description automatically generated

*Hình 18. Services*

# TÀI LIỆU THAM KHẢO

1. Tổ KTMT; *Giáo trình nguyên lý hệ điều hành*; Khoa CNTT Trường Đại học Công nghiệp Hà Nội, 2014.
2. [Ts. Hà Quang Thụy](http://www.vinabook.com/tac-gia/ts-ha-quang-thuy-i10458) ; Giáo Trình Nguyên Lý Các Hệ Điều Hành; NXB Khoa học và kỹ thuật, 2005.
3. [Hồ Đắc Phương](http://tado.vn/product-list.aspx?tacgia=H%E1%BB%93%20%C4%90%E1%BA%AFc%20Ph%C6%B0%C6%A1ng); Giáo trình Nguyên lý Hệ điều hành; NXB [Giáo dục](http://tado.vn/product-list.aspx?nhaxuatban=1004) ,2012.
4. Abraham Silberschatz, Peter B.Galvin, Greg Gagne; *Operating System Concepts*; 2008.

[1] <http://vi.wikipedia.org/wiki/Windows_Task_Manager>

[2] http://sinhvienit.net/forum/he-dieu-hanh.8651.html#post25427

[3] <http://www.cnet.com/how-to/how-to-use-windows-10-task-manager/>

[4] <http://cdth10a.forumbuild.com/viewtopic.php?f=8&t=80>

[5] http://windows.microsoft.com/en-us/windows/what-task-manager-memory-columns-mean#1TC=windows-7