



## Hệ Điều Hành thi hết học phần

Hệ điều hành (Học viện Công nghệ Bưu chính Viễn thông)

### 3.1 Trình bày khái niệm hệ điều hành. Phân tích rõ hai chức năng cơ bản của hệ điều hành

Hệ điều hành là hệ thống phần mềm đóng vai trò trung gian giữa người sử dụng và phần cứng máy tính nhằm tạo môi trường giúp thực hiện chương trình thuận tiện và hiệu quả.

HDH gồm 2 chức năng chính là quản lý tài nguyên và quản lý việc chạy chương trình.

- Quản lý tài nguyên là đảm bảo cho tài nguyên hệ thống sử dụng có ích và hiệu quả.

Tài nguyên phần cứng bao gồm CPU, bộ nhớ chính, bộ nhớ thứ cấp, thiết bị vào ra. Quản lý tài nguyên trước hết là phân phối tài nguyên tới các ứng dụng hiệu quả yêu cầu được hdh thu nhận và đáp ứng bằng cách cấp phát tài nguyên cho chương trình tương ứng, để cấp phát đc tài nguyên hdh cần phải lưu giữ tình trạng tài nguyên để biết được hiện tại tài nguyên nào còn trống, tài nguyên nào đang được sử dụng. Quản lý tài nguyên còn đảm bảo cho chương trình không xâm phạm đến tài nguyên đã cấp cho chương trình khác

- Nhiệm vụ quan trọng của máy tính là chạy chương trình đang chạy gọi là tiến trình HDH giúp việc chạy chương trình dễ dàng hơn. Để môi trường thuận lợi cho chương trình hdh tạo ra các máy ảo. Máy ảo là máy logic với những tài nguyên ảo có tính chất và khả năng khác tài nguyên thực: dễ sd hơn, dữ liệu trình hơn, số lượng nhiều hơn khả năng có thể vượt qua khả năng tài nguyên thực. Tài nguyên ảo là bản mô phỏng tài nguyên thực đc thực hiện bằng phần mềm. Tài nguyên ảo cung cấp các dịch vụ giống tài nguyên thực nhưng nó dễ sd hơn và số lượng nhiều hơn. Để sử dụng tài nguyên phần cứng dễ dàng người ta trừu tượng hóa chúng bằng cách loại bỏ những chi tiết không quan trọng, chỉ giữ lại những khía cạnh mà người sd quan tâm, sau khi trừu tượng hóa tài nguyên phần cứng vẫn cung cấp chức năng như ban đầu nhưng dễ sd hơn vì những chi tiết cụ thể đã được giấu đi.

### 3.2 Dịch vụ của hệ điều hành là gì ? Trình bày những dịch vụ điển hình mà hệ điều hành cung cấp. Làm rõ về quá trình tải và chạy hệ điều hành khi mới khởi động.

Dịch vụ hdh là những công việc mà hdh đã thực hiện giúp người dùng hoặc chương trình ứng dụng.

Những dv chính mà hdh cc:

+ Tải và chạy chương trình: Để thực hiện chương trình từ đĩa vào bộ nhớ sau đó đc trao quyền thực hiện các lệnh. Khi thực hiện xong cần giải phóng bộ nhớ các tài nguyên.

→ HDH thực hiện công về này.

+ Giao diện với người dùng: giao diện dòng lệnh, giao diện đồ họa

+ Thực hiện vào ra dữ liệu

+ Làm việc với file

+ Phát hiện và xử lý lỗi phát hiện và xử lý kịp thời những lỗi xảy ra với phần cứng cũng như phần mềm đảm bảo cho hệ thống hoạt động ổn định an toàn Truyền thông các dịch vụ cho phép thiết lập liên lạc và truyền thông tuy cấp phát tài nguyên

+ dịch vụ an ninh và bảo mật

Để thực hiện 1 chương trình, chương trình đó phải đc tải từ đĩa vào bộ nhớ sau đó đc trao quyền thực hiện lệnh, sau khi kết thúc ctrình phải giải phóng bộ nhớ và tải nguyên mà ctrình chiếm giữ, hdh sẽ thực hiện các công việc này. Do hdh là ctrình đầu tiên đc thực hiện khi hệ thống khởi động nên hdh phải tự tải chính mình vào bộ nhớ trong và tải diễn ra như sau: hdh có 1 chương trình nhỏ gọi là ctrình tải hay ctrình môi, ctrình này nằm ở 1 vị trí xác định trên đĩa hoặc bộ nhớ ngoài sau khi hệ thống khởi động, 1 chương trình nằm sẵn trong bộ nhớ ROM được kích hoạt và đọc chương trình môi ở 1 vị trí quy ước trên đĩa vào bộ nhớ

trong sau đó ctrình mỗi sẽ chịu trách nhiệm tải những phần còn lại của hdh vào bộ nhớ và trao quyền đk hệ thống cho HDH.

### **3.3 : Trình bày khái niệm dòng (thread) và mô hình đa dòng. Vấn đề sở hữu tài nguyên của tiến trình và dòng. Phân tích trụ điểm của mô hình đa dòng.**

- Dòng : mỗi đv thực hiện của tiến trình tức là 1 chuỗi lệnh đc cấp CPU và thực hiện độc lập gọi là dòng thực hiện. Đa dòng là nhiều chuỗi lệnh đc thực hiện cùng 1 lúc trong phạm vi 1 tiến trình.

- Mô hình đa dòng mỗi dòng cần có khả năng quản lý con trỏ lệnh và nội dung thanh ghi mỗi dòng cũng có trạng thái riêng đc chứa trong khối quản lý lệnh, mỗi dòng có ngăn xếp riêng. Tất cả các các dòng chia sẻ chung không gian nhớ và tài nguyên của tiến trình. Các dòng có cùng không gian để có thể truy cập dữ liệu của tiến trình.

Tài nguyên của tiến trình và dòng trong hệ thống cho phép đa dòng tiến trình vẫn là đơn vị đc HDH phân phối tài nguyên, các tiến trình đề sở hữu chung 1 số tài nguyên không gian nhớ của tiến trình, và 1 số tài nguyên khác như file đang mở và thiết bị vào ra.

- Ưu điểm của mô hình đa dòng: tăng hiệu năng và tiết kiệm thời gian, tăng tính đáp ứng dễ dàng chia sẻ tài nguyên và thông tin tận dụng đc kiến trúc xử lý với nhiều CPU, thuận lợi cho việc tổ chức chương trình.

### **3.4: Phân tích các vấn đề cần quan tâm trong sử dụng và quản lý tiến trình đồng thời (concurrent processes) đối với ba dạng tiến trình : tiến trình độc lập có cạnh tranh tài nguyên, tiến trình hợp tác nhờ chia sẻ tài nguyên, và tiến trình hợp tác nhờ trao đổi thông điệp.**

Các tiến trình cạnh tranh tài nguyên lẫn nhau :

+ Loại trừ tương hỗ : Khi các tiến trình đồng thời truy cập tài nguyên mà khả năng ese của tài nguyên đó hạn chế đảm bảo khi tiến trình truy cập tài nguyên thì tiến trình khác k đc phép truy cập Các tài nguyên như vậy gọi là tài nguyên nguy hiểm, những đoạn chương trình yc sử dụng tài nguyên nguy hiểm gọi là đoạn nguy hiểm

+ Không để xảy ra bế tắc: Bế tắc là tình trạng 2 hay nhiều tiến trình k thể tiếp tục thực hiện do phải đợi chờ lẫn nhau

+ Không để đói tài nguyên là tình trạng tiến trình phải chờ đợi quá lâu mà chưa đến lượt cấp tài nguyên. Tiến trình hợp tác vs nhau thông qua tài nguyên chung. Các tiến trình hợp tác vs nhau trao đổi ttin vs nhau thông qua chia sẻ vùng nhớ dùng chung. Việc các tiến trình truy cập đồng thời d liệu dùng chung đòi hỏi đảm bảo tính nhất quán điệu. Việc cập nhập điệu chung cung ycau đbiệt loại trừ tương hỗ, ngoài ra có thể xảy ra tình trạng bế tắc và đói.

Được chạy đua là tình huống 1 dòng or tiến trình đọc và ghi dữ và kết quả phụ thuộc vào thứ tự các thao tác vào ra vấn đề xảy ra khi có đik chạy đua có thể gọ bằng cách đặt toàn bộ thao tác vào đoạn nguy hiểm và sd loại trừ tương hỗ để thao tác này k bị tiến trình khác xen ngang Tiến trình có liên lạc nhờ gửi thông điệp Các tiến trình hợp tác vs nhau có thể II trực tiếp với nhau bằng cách gửi thông điệp ve gửi thông điệp các t.trình k c.sẽ hay cạnh tranh lẫn nhau nên k ycau loại trừ tương hỗ, nhưng có thể xảy ra bế tắc và đói.

### **3.5 : Trình bày giải thuật Peterson cho đoạn nguy hiểm. Phân tích tra nhược điểm của phương pháp này.**

Thuật toán peterson:

```

Bool flag[2];
Int turn;
Void POO{// tiến trình P0
    For{//lặp vô hạn
        Flag[0]=true;
        turn =1;
        while (flag[1] == true && turn == 1) ;//lặp đến khi dk không thỏa mãn
        <đoạn nguy hiểm>
        Flag[0]=false;
        <phần còn lại của tiến trình>
    }
}
Void P1{// tiến trình P1
    For{//lặp vô hạn
        Flag[1]=true;
        turn=0;
        while(flag[0] == true && turn == 0) ;//lặp đến khi dk k thỏa mãn
        <đoạn nguy hiểm>
        Flag[1]=false; <phần còn lại của tiến trình>
    }
}
Void main(){
    Flag[0]=flag[1]=0;
    Turn=0 ;//tắt tiến trình chính, chạy đồng thời P1 và P0
    Startprocess(PO);
    Startprocess(P1);}

```

- Ưu điểm :

Thỏa mãn các y/c :

- Đk loại trừ tương hỗ
- Đk tiến triển :

+ P0 chỉ có thể bị P1 ngăn cản vào đoạn nguy hiểm nếu flag[1] true và turn =1 luôn

đúng

+ Có 2 khả năng với P1 ở ngoài đoạn nguy hiểm :

- P1 chưa sẵn sàng vào đoạn nguy hiểm=>flag[1]=false, P0 có thể vào ngay đoạn nguy hiểm

- P1 đã đặt flag[1]=true và đang trong vòng lặp while => turn =1 hoặc 0

Turn = 0; P0 vào đoạn nguy hiểm ngay

Turn =1: P1 vào đoạn nguy hiểm, sau đó có đặt flag[1]=false=>quay lại kn 1

+ Chờ đợi giới hạn

- Nhược điểm:

- Sử dụng trên thực tế tương đối phức tạp

- Đòi hỏi tiến trình đang yêu cầu vào đoạn nguy hiểm phải nằm trong - trạng thái chờ đợi tích cực. Chờ đợi tích cực; tiến trình vẫn phải sd CPU để kiểm tra xem có thể vào đoạn nguy hiểm?=>lãng phí CPU

### 3.7 : Trình bày kỹ thuật tái trong quá trình thực hiện. Trình bày kỹ thuật liên kết động và thư viện dùng chung. Phân tích rõ ưu điểm mà từng phương pháp đem lại.

Tài trong quá trình thực hiện :

- Hàm chưa bị gọi thì chưa tải vào bộ nhớ
- Chương trình chính dc load vào bộ nhớ và chạy

Khi có lời gọi hàm:

- Chương trình sẽ ktra hàm đó dc tải vào chưa
- Nếu chưa, chương trình sẽ tiến hành tải sau đó ánh xạ địa chỉ hàm vào ko gian chung của chương trình và thay đổi bằng địa chỉ để ghi lại các ánh xạ đó.

Lập trình viên đảm nhiệm, HDH cung cấp các hàm thư viện cho tải động

- Liên kết động và thư viện dùng chung
- Liên kết tĩnh : các hàm và thư viện dc lk luôn và chương trình
- Lãng phí ko gian cả trên đĩa và MEM trong
- Trong gd lk, ko kết nối các hàm thư viện vào chương trình mà chỉ chèn các thông tin về hàm thư viện đó.
- Các modul thư viện dc lk trong quá trình thực hiện:

+ Ko giữ bản sao các modul thư viện mà tiến trình giữ stub chứa thông tin về modul thư viện.  
+ Khi stub dc gọi, nó ktra modul t/ứ đã có trong bộ nhớ chưa. Nếu chưa, thì tải phần còn lại về và dùng

- + Lần tiếp theo cần sd, modul thư viện sẽ dc chạy trực tiếp
- + Mỗi modul thư viện chỉ có 1 bản sao duy nhất chứa trong bộ nhớ
  - Cần hỗ trợ từ HDH

### **3.8 : Trình bày kỹ thuật phân chương động bộ nhớ. Phân tích ưu nhược điểm của phương pháp này so với phân chương cố định. Khi di chuyển chương sang vị trí khác cần thay đổi thông tin gì trong khối ánh xạ địa chỉ.**

- + Kỹ thuật :
  - Kích thước, số lượng và vị trí chương đều có thể thay đổi
  - Khi có y/c, HDH cấp cho tiến trình 1 chương có kích thước đúng bằng tiến trình đó.
  - Khi tiến trình kt sẽ tạo vùng trống trong bộ nhớ.
  - Các vùng trống nằm cạnh nhau dc nhập lại thành vùng lớn hơn
  - Sd các chiến lược cấp chương :
    - Chọn vùng thích hợp đầu tiên
    - Vùng thích hợp nhất
    - Vùng ko thích hợp nhất
- + Ưu điểm : Tránh phân mảnh trong
- + Nhược điểm : Gây phân mảnh ngoài : dồn những vùng trống nhỏ thành lớn(nén). Nếu chương bị di chuyển thì nội dung của thanh ghi cơ sở bị thay đổi chứa địa chỉ vị trí mở

### **3.9 : Trình bày kỹ thuật phân chương sử dụng phương pháp kẻ cận (buddy). Phân tích rõ các điểm giống khác nhau và ưu nhược điểm của phương pháp này so với phân chương cố định và phân chương động (lưu ý không cần trình bày lại hai phương pháp sau).**

- + Trình bày kỹ thuật :
  - Các chương và khối trống có kích thước là lũy thừa của  $2^k (L < k \leq H)$ :  $2^L$  : kích thước nhỏ nhất của chương ;  $2^H$  ; kích thước MEM
  - Đầu tiên, toàn bộ ko gian nhớ là  $2^H$ , y/c cấp vùng nhớ S

- $2^{H-1} < S \leq 2^H$  ; cấp cả  $2^H$
  - Chia đôi thành 2 vùng  $2^{H-1}$
  - Tiếp tục chia đôi tới khi tìm đc vùng tháo  $m \cdot 2^{k+1} < s \cdot 2^k$
- Sau 1 thời gian xuất hiện các khối trống có kích thước  $2^k$
- Tạo danh sách móc nối các vùng có cùng kích thước
- Nếu có 2 khối trống cùng kích thước và kề nhau thì ghép lại thành 1
- Khi cần cấp, sẽ tìm trong danh sách khối phù hợp nhất; nếu ko tìm khối lớn hơn và cắt đôi.

### 3.10 : Trình bày kỹ thuật phân đoạn bộ nhớ bao gồm cả cấu trúc và cách ánh xạ địa chỉ. Phân tích so sánh ưu nhược điểm của phân đoạn với phân trang.

Kỹ thuật phân đoạn bộ nhớ :

- Cấu trúc :
  - + Chương trình thường đc chia thành nhiều phần : dữ liệu, lệnh, ngăn xếp
  - + Chia chương trình thành các đoạn theo cấu trúc logic
  - + Mỗi đoạn đc phân vào 1 vùng nhớ, có kích thước ko bằng nhau.
  - + Mỗi đoạn Vứ với ko gian địa chỉ riêng, đc phân biệt bởi tên (STT) và độ dài của mình.
  - + Các vùng nhớ thuộc các đoạn khác nhau có thể nằm ở vị trí khác nhau.
  - + Bộ nhớ đc cấp phát theo từng vùng kích thích thay đổi giống phân chương động)
  - + Chương trình có thể chiếm nhieu hơn 1 đoạn và ko cần liên tiếp nhau trong MEM khác phân chương động)
  - + Tránh hiện tượng phân trang mảnh trong
  - + Có phân mảnh ngoài
  - + Dễ sắp xếp bộ nhớ
  - + Dễ chia sự các đoạn giữa các tiến trình khác nhau
  - + Kích thước mỗi đoạn có thể thay đổi mà ko ảnh tới các đoạn khác
- Ánh xạ địa chỉ:
 

Sdung hàng đoạn cho mỗi tiến trình. Mỗi ô tương ứng với 1 đoạn chứa:

  - Địa chỉ cơ sở: vị trí bắt đầu của đoạn trong bộ nhớ
  - Địa chỉ giới hạn độ dài đoạn, sd để chống truy cập trái phép ra ngoài đoạn
- + Địa chỉ logic gồm 2 thành phần(so):
  - S: số thứ tự tên đoạn
  - O: độ dịch trong đoạn XII :

### 3.11: Trình bày kỹ thuật nạp trang theo nhu cầu dùng cho bộ nhớ ảo. Phân tích rõ cùng một lệnh có thể xảy ra nhiều sự kiện lỗi trang không?

Trình bày kỹ thuật:

- Tiến trình do phân trang và chứa trên đĩa
  - Khi thực hiện, nạp tiến trình vào MEM : chỉ nạp những trang cần dùng
  - Tiến trình gồm các trang trên đĩa và trong MEM : thêm bit P trong khoản mục bảng trang để phân biệt(P=1 : đã nạp vào MEM)
  - Quá trình kiểm tra và nạp trang :
  - + Tiến trình truy cập tới 1 trang, kiểm tra bit P. Nếu P=1, truy cập diễn ra bình thường Nếu P=0, xảy ra sự kiện thiếu trang.
- Ngắt xử lý thiếu trang:
- + HDH tìm 1 khung trống trong MEM

- + Đọc trang bị thiếu vào khung trang vừa tìm đc
- + Sửa lại khoản mục tương ứng trong bảng trang: đổi bút P=1 và số khung đã cấp cho trang
- + Khôi phục lại trạng thái tiến trình và thực hiện tiếp lệnh bị ngắt Nẹp trang hoàn toàn theo nhu cầu:
- + Bắt đầu 1 tiến trình mà ko nạp bất kì trang nào vào bộ nhớ
- + Khi con trở lệnh đc HDH chuyển tới lệnh đầu tiên của tiến trình để thực hiện, sự kiện thiếu trang sẽ sinh ra và trang tự đc nạp vào
- + Tiến trình sau đó thực hiện bt cho tới lần thiếu trang tiếp theo

### **3.12: Trình bày cấu trúc thư mục dạng cây và dạng đồ thị không có của trình. Cấu trúc thư mục dạng không chu trình có ưu điểm gì so với động cây ? Thế nào là đường dẫn tuyệt đối và đường dẫn tương đối.**

Thư mục cấu trúc cây:

- + Thư mục con có thể chứa các thư mục con khác và các file +Hthog thư mục de bd phân cấp như 1 cây: cành là thư mục, lá là file
- + Phân biệt khoản mục file và khoản mục của thư mục con: thêm bit db trong khoản mục
  - 1 : khoản mục của thư mục dưới
  - 0 : khoản mục của file
- + Tại mỗi thời điểm, ng dùng làm việc với thư mục hiện thời
- + Tổ chức cây trong thư mục cho từng đĩa:
- Trong hệ thống file như FAT của ĐÓ, cây thư mục đc xây cho từng đĩa. Hệ thống thư mục đc coi là rừng, mỗi cây trên 1 đĩa.
- Linux: toàn ht chỉ gồm 1 cây thư mục Thư mục cấu trúc ko tuần hoàn(acyclic graph):
  - + Chia sẻ file và thư mục để có thể xuất hiện ở nhiều thư mục riêng khác nhau
  - + Mở rộng của cấu trúc cây: là và cành có thể đồng thời thuộc về những cành khác nhau!
  - + Triển khai:
    - SD liên kết: con trỏ tới thư mục or file khác
    - tạo bản sao của file và thư mục cần chia sẻ và chứa vào các thư mục khác nhau=> phải đảm bảo tính đồng bộ và nhất quán.
- + Mềm dẻo nhưng phức tạp
  - Cấu trúc thư mục dạng ko chu trình có ưu điểm là mềm dẻo hơn so với cấu trúc dạng cây.
  - Đường dẫn tuyệt đối :
- + Đg dẫn từ gốc của cây thư mục, đi qua các thư mục trung gian, dan toi file.
- + C:\bc\bin\bc.exe
- Đường dẫn tương đối:
  - + Tính từ thư mục hiện thời
  - + Thêm 2 khoản mục đặc biệt trong thư mục “.”, “..”

### **3.13. Trình bày phương pháp cấp phát không gian cho file sử dụng danh sách kết nối và sử dụng khối chỉ số (I-node). Hai phương pháp này có điểm 5 tầng và khác nhau.**

Giống nhau: Khoản mục của file trong thư mục chứa con trỏ tới khối này.  
khác nhau:



| So sánh     | DS kết nối   | Khối chỉ số  |
|-------------|--|--|
| Phương pháp | <ul style="list-style-type: none"> <li>- Các khối thuộc 1 file có thể nằm bất kỳ trên đĩa</li> <li>- Khi file đc cấp thêm khối mới, khối đó được thêm vào cuối danh sách</li> <li>- Các khối được kết nối thành danh sách</li> <li>- Phần đầu mỗi khối chứa con trỏ tới khối tiếp theo</li> <li>- HDH đọc lần lượt từng khối và sd con trỏ để xd khối tiếp theo</li> </ul> | <ul style="list-style-type: none"> <li>- Tất cả con trỏ tới các khối thuộc về 1 file đc tập trung 1 chỗ</li> <li>- Mỗi file có 1 mảng riêng của mình chứa trong 1 khối gia khối chỉ mục (I-node)</li> <li>- Mảng chứa thuộc tính của file kết nối và vị trí của khối trên đĩa</li> <li>- Ô thứ 1 của mảng chứa con trỏ tới khối thứ 1 của file.</li> <li>- Chọn kích the I-node: <ul style="list-style-type: none"> <li>+ Nhỏ: Tiết kiệm kngian nhưng k đủ con trỏ các khối nếu file lớn</li> <li>+ Lớn: với file nhỏ chỉ chiếm 1 vài o thi lãng phí</li> </ul> </li> <li>- Giải pháp: thay đổi size I-node = sd ds liên kết; sd I-node có ctruc nhiều mức.</li> </ul> |
| Ưu điểm     | <ul style="list-style-type: none"> <li>- K bị phân mảnh</li> <li>- K y/c biết trê của file lúc tạo.</li> <li>- dễ tìm vị trí cho file, khoản mục đơn giản</li> </ul>   | <ul style="list-style-type: none"> <li>- Cho phép truy cập trực tiếp.</li> <li>- Các khối thuộc về 1 file k cần nằm liên tiếp nhau.</li> </ul>   |
| Nhược điểm  | <ul style="list-style-type: none"> <li>- K hỗ trợ truy cập</li> <li>- Tốc độ truy cập k cao</li> <li>- Giảm độ tin cậy và tính toàn vẹn của ht file.</li> </ul>  | <ul style="list-style-type: none"> <li>- Tốc độ truy cập chậm</li> </ul>   |

### 3.14 : Trình bày về yêu cầu phải đảm bảo tính toàn vẹn của hệ thống file và các phương pháp đảm bảo tính toàn vẹn.

+ Yêu cầu phải đảm bảo tính bảo toàn :

- Hệ thống file chứa nhieu csdl có mối lk => thông tin về liên kết bị hư hại, tính toàn vẹn của hệ thống bị phá vỡ.
- Các khối ko có mặt trong danh sách các khối trống, đồng thời cũng ko có mặt trong 1 file nào đó.
- 1 khối có thể vừa thuộc về 1 file nào đó vừa có mặt trong ds trống
- HDH có các chương trình ktr tính toàn vẹn của ht file đc chạy khi ht khởi động, do là sau sự cố

+ Các pp đảm bảo tính toàn vẹn

- Giao tác là 1 tập hợp các thao tác cần phải đc thực hiện trọn vẹn
- Với ht file : mỗi giao tác sẽ bao gồm những thao tác thay đổi lk cùng nhau. cần thực hiện cùng nhau
- Toàn bộ trạng thái ht file đc ghi lại trong file log
- Nếu giao tác ko đc thực hiện trọn vẹn, HDH sd thông tin từ log để khôi phục ht file về trạng thái ko lỗi trc khi thực hiện giao tác