

MÔN HỆ ĐIỀU HÀNH

Chương 1

TỔNG QUÁT VỀ HỆ ĐIỀU HÀNH

- 1.1 Định nghĩa sơ lược về hệ điều hành
- 1.2 Lịch sử phát triển hệ điều hành
- 1.3 Phân loại các hệ điều hành
- 1.4 Nhắc lại phần cứng máy tính
- 1.5 Các khái niệm cơ bản về hệ điều hành
- 1.6 Các lời gọi dịch vụ HĐH "System call"
- 1.7 Cấu trúc hệ điều hành

Tài liệu tham khảo : chương 1, sách "Modern Operating Systems",
Andrew S. Tanenbaum: , 2nd ed, Prentice Hall



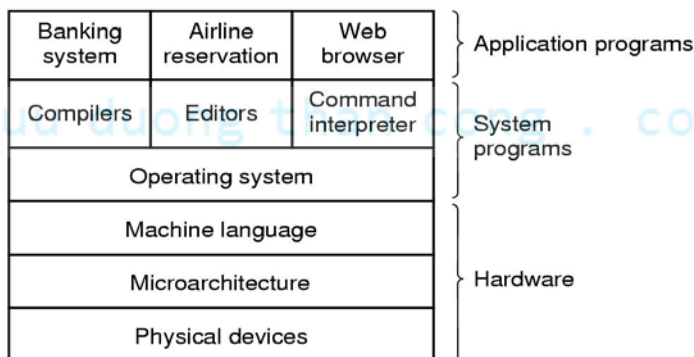
Khoa Công nghệ Thông tin
Trường ĐH Bách Khoa Tp.HCM

Môn : Hệ điều hành
Chương 1 : Tổng quát về hệ điều hành
Slide 1

cuu duong than cong . com

1.1 Định nghĩa sơ lược về hệ điều hành

- Máy tính số là máy nhiều cấp, trong đó 3 cấp chính yếu là :
 - vật lý (phần cứng - hardware)
 - chương trình hệ thống (system programs)
 - chương trình ứng dụng (application programs)



Khoa Công nghệ Thông tin
Trường ĐH Bách Khoa Tp.HCM

Môn : Hệ điều hành
Chương 1 : Tổng quát về hệ điều hành
Slide 2

Hệ điều hành là gì?

Hai định nghĩa được nhiều người đồng ý nhất :

1. **HĐH là 1 máy tính luận lý mở rộng (extended machine)** : đây là góc nhìn từ ngoài vào.

- dấu các chi tiết khó, phiền phức cần thực hiện.
- cung cấp cho người dùng 1 máy luận lý dễ dùng hơn và độc lập với phần cứng (thông qua các lệnh *system calls*)

2. **HĐH là 1 hệ quản lý các tài nguyên của máy** : đây là góc nhìn bên trong

- Phân chia việc dùng tài nguyên theo thời gian, mỗi chương trình dùng tài nguyên trong 1 khoảng thời gian rồi giao lại cho chương trình khác dùng (CPU, máy in).
- Phân chia tài nguyên theo không gian : mỗi chương trình dùng 1 vùng nhỏ tài nguyên (bộ nhớ, đĩa).



cuu duong than cong . com

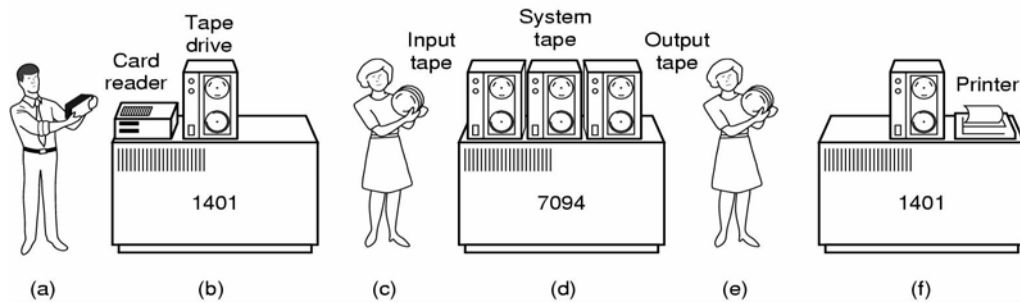
1.2 Lịch sử hệ điều hành

Vì HĐH nằm trên cấp phần cứng nên lịch sử HĐH gắn liền với lịch sử phát triển phần cứng máy tính. Ở đây chúng ta tổng kết lại lịch sử phát triển máy tính số gồm 4 thế hệ sau :

1. First generation 1945 - 1955
 - vacuum tubes, plug boards
 - Inventors : Aiken (USA), Zuse (Germany)
 - chưa cần HĐH
2. Second generation 1955 - 1965
 - transistors
 - batch systems
3. Third generation 1965 – 1980
 - ICs (Integrated Circuits)
 - multiprogramming, spooling, time-sharing
4. Fourth generation 1980 – present
 - LSI (Large Scale Integration)
 - Hệ điều hành cho PC



Lịch sử hệ điều hành - Thế hệ thứ 2



Early batch system (hệ thống xử lý bó)

- xuyên phiếu chuyển chương trình thành chồng card đục lỗ.
- để n chồng card theo thứ tự cho máy đọc card 1401 đọc và ghi lên băng từ.
- gắn băng từ cho máy 7094 xử lý tuần tự từng chương trình, kết quả của chương trình được ghi lên băng kết xuất.
- gắn băng kết xuất vào máy in 1401 để in ra giấy.

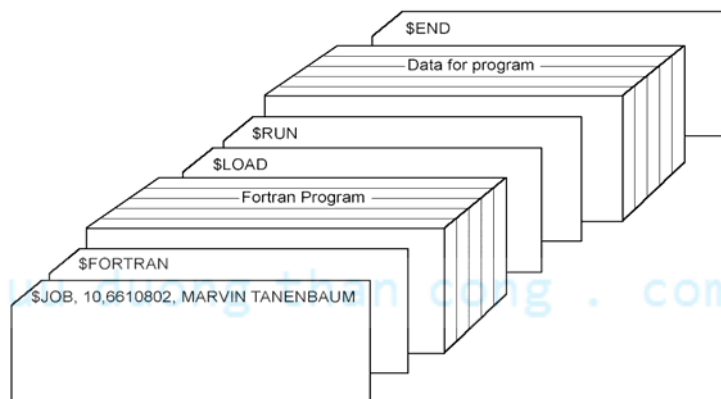


Khoa Công nghệ Thông tin
Trường ĐH Bách Khoa Tp.HCM

Môn : Hệ điều hành
Chương 1 : Tổng quát về hệ điều hành
Slide 5

cuu duong than cong . com

Lịch sử hệ điều hành - Thế hệ thứ 2



Cấu trúc điển hình của 1 job FMS

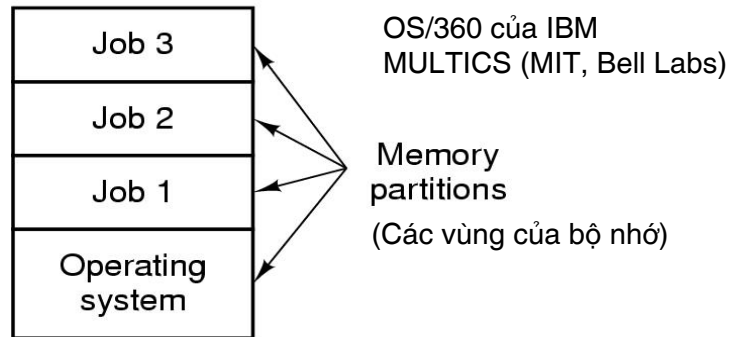
(FMS: Fortran Monitor System, hệ điều hành của IBM cho mainframe 7094)



Khoa Công nghệ Thông tin
Trường ĐH Bách Khoa Tp.HCM

Môn : Hệ điều hành
Chương 1 : Tổng quát về hệ điều hành
Slide 6

Lịch sử hệ điều hành - Thế hệ thứ 3



- ❑ Multiprogramming system
- ❑ Spooling (Simultaneous Peripheral Operation On Line)
- ❑ Time sharing



cuu duong than cong . com

Lịch sử hệ điều hành - Thế hệ thứ 4

- ❑ 1974, first microcomputer
 - Intel 8080, first general-purposed 8-bit CPU
 - floppy disk
 - CP/M (Control Program for Microcomputers)
- ❑ early 1980s, IBM PC
 - DOS (Disk Operating System)
 - MS-DOS (Microsoft Disk Operating System)
- ❑ 1983, IBM PC/AT (Intel 80286 CPU)
- ❑ 1985-1995, Windows on top of MS-DOS
- ❑ Pentium PC
 - UNIX, Linux, Windows 2000
 - X Windows system (UNIX, Linux)



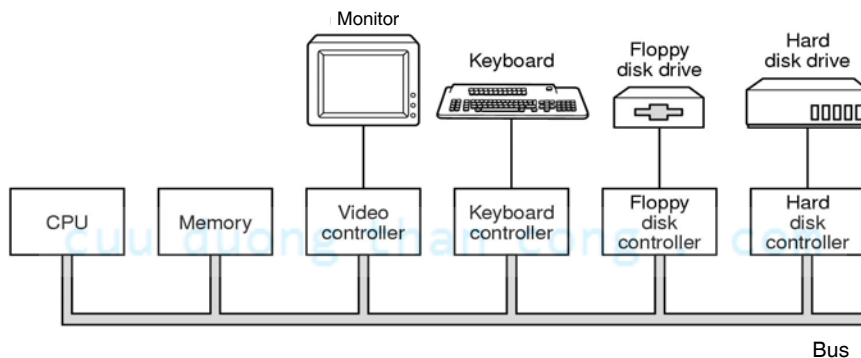
1.3 Phân loại các hệ điều hành

- | | |
|---------------------------------------|---------------------------|
| ❑ Mainframe operating systems | High-end Web servers |
| ▪ OS/390 | |
| ❑ Server operating systems | Web service, file service |
| ▪ UNIX, Linux, Windows 2000 | |
| ❑ Multiprocessor operating systems | |
| ❑ Personal computer operating systems | |
| ▪ Linux, Windows XP, Macintosh | |
| ❑ Real-time operating systems | Control systems |
| ▪ VxWorks, QNX | |
| ❑ Embedded operating systems | Mobile phones |
| ▪ uCLinux, PalmOS, Windows CE | |
| ❑ Smart card operating systems | Smart cards |



cuu duong than cong . com

1.4 Nhắc lại phần cứng máy tính



Các thành phần của một máy PC đơn giản



Nhắc lại phần cứng máy tính - Processors

- ❑ Special registers
 - Program counter
 - Stack pointer
 - Program Status Word (PSW)
 - o kernel mode
 - o user mode
- ❑ TRAP instruction
 - System call



cuu duong than cong . com

Nhắc lại phần cứng máy tính - Memory

Typical access time

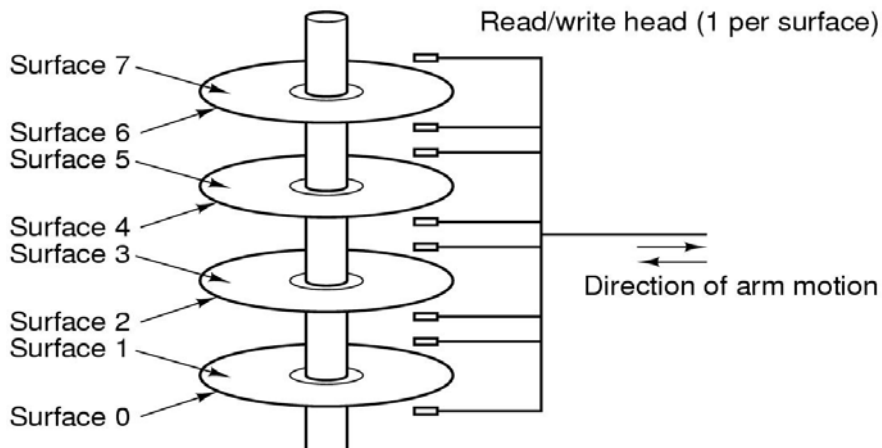
Typical capacity

1 nsec	Registers	<1 KB
2 nsec	Cache	1 MB
10 nsec	Main memory	64-512 MB
10 msec	Magnetic disk	5-50 GB
100 sec	Magnetic tape	20-100 GB

Phân cấp điển hình các loại bộ nhớ
các giá trị chỉ có ý nghĩa xấp xỉ



Nhắc lại phần cứng máy tính - Đĩa cứng



Cấu trúc của một ổ đĩa cứng



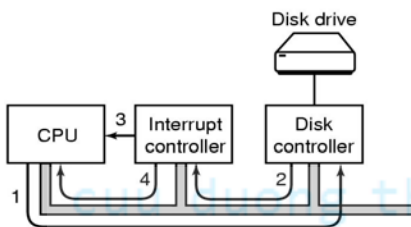
Khoa Công nghệ Thông tin
Trường ĐH Bách Khoa Tp.HCM

Môn : Hệ điều hành
Chương 1 : Tổng quát về hệ điều hành
Slide 13

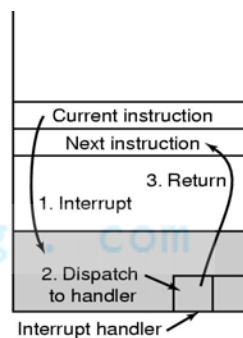
cuu duong than cong . com

Nhắc lại phần cứng máy tính - Thiết bị I/O

Device driver là gì?



(a)



(b)

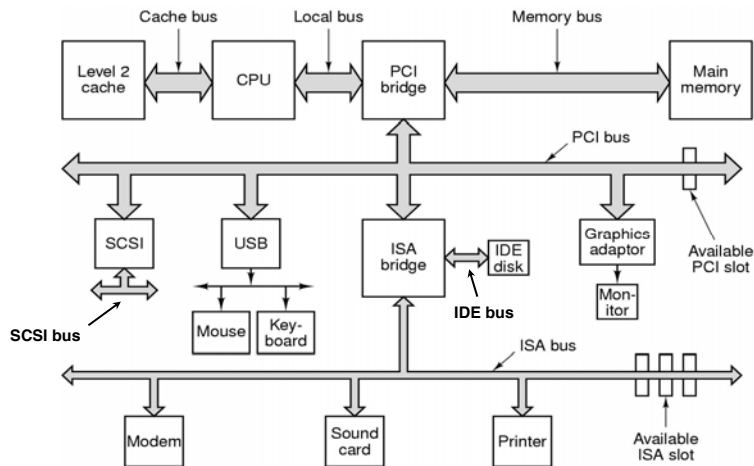
- (a) Steps in starting an I/O device and getting interrupt
- (b) Interrupt processing



Khoa Công nghệ Thông tin
Trường ĐH Bách Khoa Tp.HCM

Môn : Hệ điều hành
Chương 1 : Tổng quát về hệ điều hành
Slide 14

Nhắc lại phần cứng máy tính - Bus



Cấu trúc của một hệ thống Pentium



Khoa Công nghệ Thông tin
Trường ĐH Bách Khoa Tp.HCM

Môn : Hệ điều hành
Chương 1 : Tổng quát về hệ điều hành
Slide 15

cuu duong than cong . com

1.5 Các ý niệm chủ đạo trong hệ điều hành

- ❑ Các tài nguyên của máy
- ❑ Quá trình (process)
- ❑ Lập thời biểu cho các quá trình chạy đồng thời (Scheduler)
- ❑ Cho phép các quá trình đồng thời truy xuất chung tài nguyên
- ❑ Deadlock và giải quyết
- ❑ Quản lý bộ nhớ (memory management)
- ❑ Hệ thống file (tập tin)
- ❑ Giao tiếp với thế giới ngoài (input/output)
- ❑ An ninh dữ liệu (security)
- ❑ The shell



Khoa Công nghệ Thông tin
Trường ĐH Bách Khoa Tp.HCM

Môn : Hệ điều hành
Chương 1 : Tổng quát về hệ điều hành
Slide 16

Các ý niệm chủ đạo trong hệ điều hành - Tài nguyên

- ❑ Tài nguyên của 1 chương trình là bất kỳ thành phần nào của máy tính được sử dụng bởi chương trình đó.
 - Tài nguyên phần cứng : CPU, bộ nhớ, đĩa, CDROM, đĩa USB, màn hình, bàn phím, chuột, card mạng,...
 - Tài nguyên phần mềm : các file dữ liệu và các hệ thống phần mềm khác mà 1 chương trình cần truy xuất/tương tác.
- ❑ HĐH cần quản lý các tài nguyên sao cho việc sử dụng chúng bởi các chương trình được tin cậy, an toàn, hiệu quả và độc lập với tính chất vật lý của chúng.



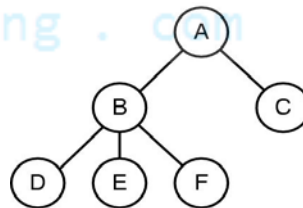
cuu duong than cong . com

Các ý niệm chủ đạo trong hệ điều hành - Process

File chương trình thường có 2 dạng : mã nguồn và mã thực thi. File thực thi (*.exe trên Windows) có thể được chạy trực tiếp bởi máy, nhưng nếu chưa chạy, nó vẫn là thành phần thụ động, ngủ yên và không tạo ra kết quả gì. Khi người dùng kích hoạt 1 file chương trình, nó được chạy bởi CPU, lúc này ta gọi nó bằng thuật ngữ "**Tiến trình**" (**Process**). Trong lúc hoạt động, process có thể tạo ra nhiều process khác (con) và cứ thế tiếp tục.

Một cây process (process tree)

- A đã tạo hai process con, B và C
- B đã tạo ba process con, D, E, và F



Các ý niệm chủ đạo trong hệ điều hành - Scheduler

- ❑ Để quản lý việc chạy các process đơn giản và dễ dàng nhất, người ta đã cho chúng chạy tuần tự, mỗi lần cho 1 chương trình chạy. Chỉ khi chương trình chạy xong (dù lâu hay mau) thì ta mới cho chương trình khác chạy,...
- ❑ Hầu hết các chương trình đều cần giao tiếp với người (hay I/O nói chung). Việc giao tiếp với I/O thường chậm hơn rất nhiều so với tốc độ của CPU, nghĩa là lúc chương trình dừng chờ I/O (chờ nhập phím), CPU phải ngủ chờ mất thời gian và hiệu suất làm việc của nó.
- ❑ Để sử dụng CPU hiệu quả hơn, người ta cố gắng cho nhiều chương trình chạy đồng thời. Cách thông thường nhất là dùng kỹ thuật phân chia thời gian (Time sharing) : chia trục thời gian làm nhiều khe nhỏ (quantum), cho mỗi chương trình chạy 1 khe nhỏ rồi dừng nó lại, chọn chương trình khác chạy trong khe nhỏ kế



cuu duong than cong . com

Các ý niệm chủ đạo trong hệ điều hành - Scheduler

- ❑ Module của HĐH quản lý việc phân chia thời gian cho các chương trình chạy được gọi là **trình lập thời biểu (Scheduler)**.

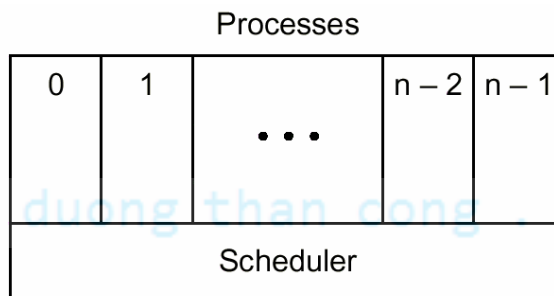


Figure 2-3. The lowest layer of a process-structured operating system handles interrupts and scheduling. Above that layer are sequential processes.



Vấn đề truy xuất 1 tài nguyên dùng chung

- ❑ Như vậy, về mặt vật lý chi ly, các chương trình không bao giờ chạy đồng thời trên 1 máy có 1 CPU vì tại từng khe thời gian, chỉ có 1 chương trình được chạy, các chương trình khác đều bị dùng chờ.
- ❑ Tuy nhiên theo góc nhìn người dùng (góc nhìn luận lý, góc nhìn ngữ nghĩa) thì ta cảm nhận rằng các chương trình chạy đồng thời.
- ❑ Nếu 2 hay nhiều chương trình chạy đồng thời và nếu chúng muốn truy xuất 1 tài nguyên (dùng chung) nào đó thì ta sẽ giải quyết ra sao ?
- ❑ Về nguyên tắc, ta phải cho chương trình truy xuất, nhưng nếu không kiểm soát việc truy xuất đồng thời vào cùng 1 tài nguyên thì có thể dẫn đến trình trạng "Race". Race là hiện tượng lỗi bất định có thể xảy ra khi 2 hay nhiều process truy xuất 1 tài nguyên dùng chung đồng thời.



cuu duong than cong . com

Vấn đề truy xuất 1 tài nguyên dùng chung

Thí dụ 2 ứng dụng truy xuất tài khoản A đồng thời :

1. hiển thị giao diện & chờ người dùng ra lệnh
2. Người dùng ra lệnh nạp vào tài khoản A số tiền 700USD → xử lý :
 - 21a Đọc tài khoản A vào bộ nhớ,
 - 22a Tăng giá trị tài khoản trong bộ nhớ lên 700USD.
 - 23a Ghi lại giá trị mới.
3. Quay về bước 1

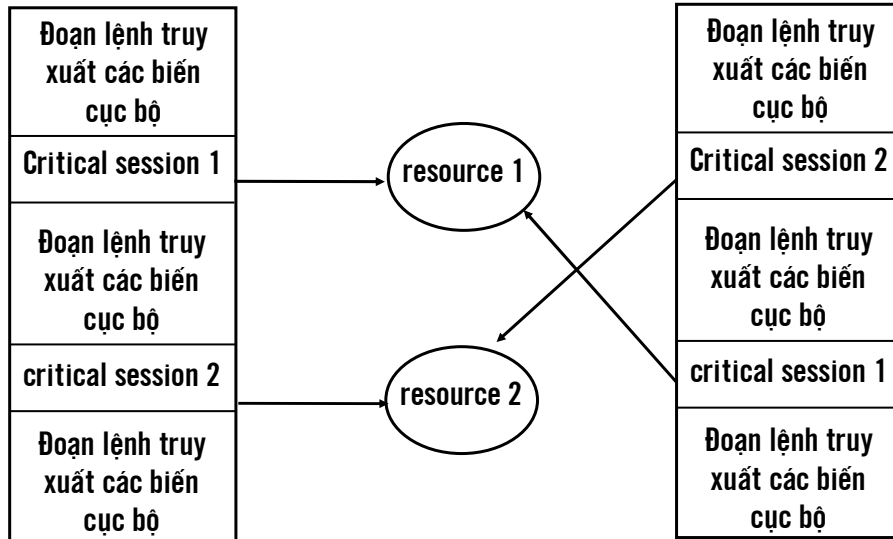
Tài khoản
A

1. hiển thị giao diện & chờ người dùng ra lệnh
2. Người dùng ra lệnh rút tiền từ tài khoản A 500USD → xử lý :
 - 21b Đọc tài khoản A vào bộ nhớ,
 - 22b Giảm giá trị tài khoản trong bộ nhớ đi 500USD.
 - 23b Ghi lại giá trị mới.
3. Quay về bước 1

Nếu tài khoản A là 1000USD và HĐH điều khiển chạy 2 process P1 và P2 theo thứ tự 21a→22a→21b→22b→23b→23a thì kết quả tài khoản A sẽ là 1700USD (giá trị đúng là 1200USD).



Vấn đề truy xuất 1 tài nguyên dùng chung



Vấn đề truy xuất 1 tài nguyên dùng chung

- ❑ Để tránh tình trạng "race", ta sẽ loại trừ tương hỗ (Mutual Exclusion) các vùng code "critical session" truy xuất cùng 1 tài nguyên dùng chung của các process, nghĩa là không cho các vùng CS này chạy đồng thời mà phải tuần tự hóa việc chạy của chúng.
- ❑ Vì vùng CS thường rất hiếm trong chương trình và rất ngắn, nên việc tuần tự hóa việc chạy chúng không ảnh hưởng nhiều đến việc chạy đồng thời các process tương ứng.



Deadlock & giải quyết

- ❑ Tuy nhiên kỹ thuật loại trừ tương hỗ các process lại thường dẫn đến mối nguy hiểm cho hệ thống mà người ta gọi là "deadlock".
- ❑ Deadlock là tình trạng của hệ thống mà ở đó có ít nhất 2 process đang dừng chờ lẫn nhau và bị kẹt mãi mãi ở trạng thái này. Trường hợp xấu nhất là mọi process đều bị dừng và chờ lẫn nhau, hệ thống sẽ bị tê liệt mãi mãi.



cuu duong than cong . com

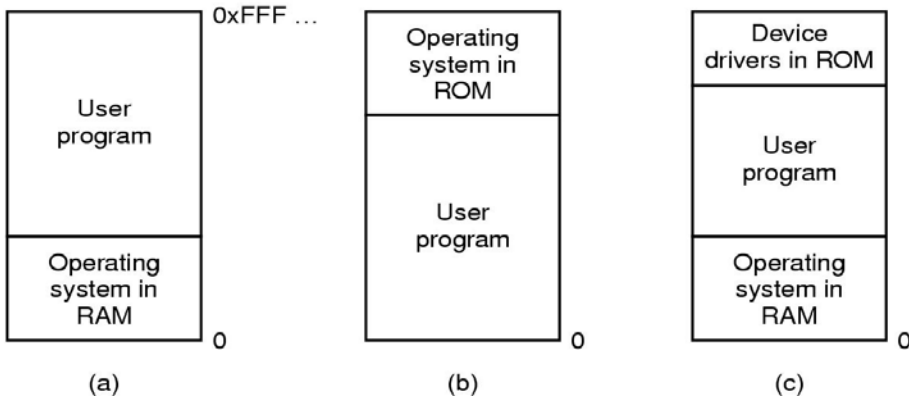
Deadlock & giải quyết

- ❑ Thí dụ giả sử có 2 process A và B đang chạy, theo giải thuật process A sẽ truy xuất file1 rồi file2, trong khi đó process B sẽ truy xuất file2 rồi file1 với tiến độ thời gian cụ thể như sau :
 - tại t1 : process A xin truy xuất file1 $\Rightarrow$ OK $\Rightarrow$ chạy tiếp.
 - tại t2 : process B xin truy xuất file2 $\Rightarrow$ OK $\Rightarrow$ chạy tiếp.
 - tại t3 : process A xin truy xuất file2 (vẫn còn truy xuất file1 nên chưa trả) $\Rightarrow$ không được $\Rightarrow$ phải dừng đợi process B.
 - tại t4 : process B xin truy xuất file1 (vẫn còn truy xuất file2 nên chưa trả) $\Rightarrow$ không được $\Rightarrow$ phải dừng đợi process A.
 - từ t4 trở đi : cả 2 process A và B đều bị dừng vì phải chờ lẫn nhau và chúng không bao giờ chạy được nữa.
- ❑ Cần phải giải quyết deadlock, chi tiết được trình bày trong môn HĐH.



Quản lý bộ nhớ trong hệ đơn chương

- Trong hệ đơn chương : 3 cách tổ chức bộ nhớ gồm vùng nhớ HĐH và vùng nhớ của 1 process đang chạy.

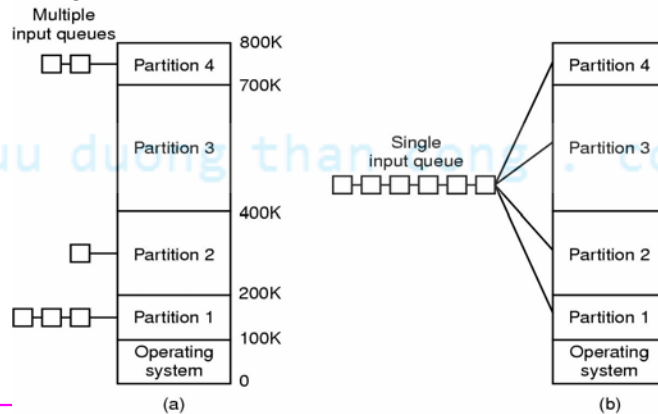


cuu duong than cong . com

Quản lý bộ nhớ - Phân vùng tĩnh

Chia bộ nhớ ra nhiều partition với độ lớn khác nhau để chạy nhiều process đòi hỏi kích thước khác nhau.

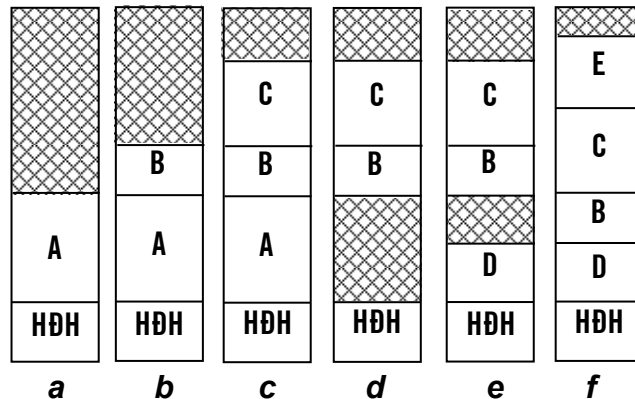
- (a) mỗi partition có hàng chờ các process đòi hỏi cùng dung lượng nhớ.
- (b) dùng 1 hàng chờ cho mọi process



Quản lý bộ nhớ - Phân vùng động

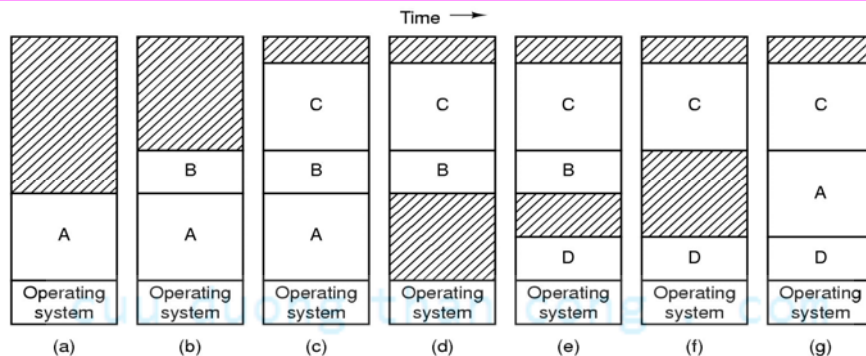
Vùng nhớ lúc đầu để nguyên. Mỗi khi có process xin cấp phát vùng nhớ, hệ thống sẽ tạo 1 partition có kích thước vừa đúng theo yêu cầu, phần còn lại để trống... Theo thời gian, bộ nhớ có thể bị băm nát bởi nhiều vùng nhớ được trả lại bởi các process.

Ta có thể khắc phục vấn đề này bằng cách sắp xếp lại các vùng nhớ sao cho vùng nhớ trống là duy nhất & liên tục (compactage).



cuu duong than cong . com

Quản lý bộ nhớ - Phân vùng động & Swapping



Các vùng bị thay đổi như sau :

- process B được cấp 1 vùng nhớ để chạy (Fig. b)
- process A is swapped lên disk hay trả lại, Fig. (d)
- process A is swapped vào từ disk khi cần chạy tiếp, Fig. (g)



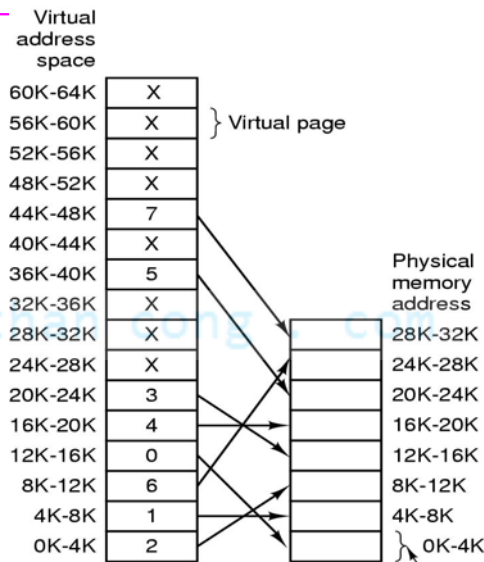
Quản lý bộ nhớ ảo (Virtual memory Man.)

- ❑ Việc swap toàn vùng nhớ của 1 process ra/vào đĩa gặp khá nhiều phiền hà do kích thước của mỗi process rất khác nhau. Tuy nhiên ý tưởng này dẫn đến cơ chế quản lý bộ nhớ tinh vi mà hiện nay các HĐH đều dùng, đó là cơ chế quản lý bộ nhớ ảo.
- ❑ Ý tưởng cơ bản là tại từng thời điểm chương trình chạy, ta không cần nội dung của toàn chương trình và dữ liệu của nó trong bộ nhớ, ta chỉ cần nội dung của lệnh đang cần chạy và dữ liệu mà lệnh này cần truy xuất, mọi thứ khác có thể để trên đĩa.
- ❑ Như vậy để chạy được 1 process, ta chỉ cần 1 vùng rất nhỏ bộ nhớ bất chấp kích thước của process đó lớn hay nhỏ.
- ❑ Có 3 kỹ thuật quản lý bộ nhớ ảo : quản lý theo phân trang, quản lý theo phân đoạn và quản lý theo phân đoạn và phân trang. Chi tiết sẽ được trình bày trong môn HĐH.

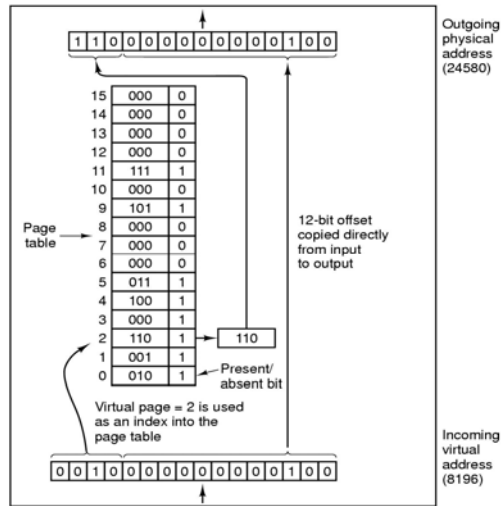


Quản lý bộ nhớ ảo (Virtual memory Man.)

Mỗi process có bảng quản lý trang, bảng này chứa thông tin về việc ánh xạ từng trang ảo của process vào từng trang thật bộ nhớ tại từng thời điểm theo thời gian.



Đổi địa chỉ ảo ra địa chỉ thật



cuu duong than cong . com

Hệ thống file (File System)

- ❑ disk vật lý là không gian dữ liệu 3 chiều : disk = nhiều cylinder, mỗi cylinder gồm nhiều track (head - vòng tròn chứa tin cùng bán kính), mỗi track chứa nhiều cung nhỏ chứa tin được truy xuất độc lập (sector). Sector là đơn vị truy xuất tin nhỏ nhất ở cấp vật lý, từ ngoài ta không thể truy xuất từng byte rời rạc trên đĩa được.
- ❑ Muốn truy xuất 1 sector, ta phải xác định được bộ ba chỉ số (C,H,S) $\Rightarrow$ rất khó dùng.
- ❑ Hơn nữa, dữ liệu có nghĩa cần lưu trên đĩa thường có kích thước rất khác nhau $\Rightarrow$ cần nhiều sector mới chứa đủ. Nếu việc quản lý 1 dữ liệu có nghĩa được chứa trên bao nhiêu sector đĩa và chỉ số cụ thể là gì được giao cho người dùng thì họ sẽ gặp rất nhiều rắc rối $\Rightarrow$ cần 1 giao tiếp sử dụng khác để sử dụng đĩa dễ dàng hơn.



Hệ thống file (File System)

- ❑ disk luận lý cấp #1 là không gian dữ liệu 1 chiều : disk = danh sách nhiều đơn vị chứa tin có độ dài cố định, mỗi đơn vị được gọi là cluster (hay block, sector luận lý). Độ dài của cluster cần đọc lập với đĩa vật lý.
- ❑ Ở cấp độ này, muốn truy xuất 1 cluster, ta chỉ cần xác định 1 chỉ số của nó.
- ❑ Tuy nhiên, dữ liệu có nghĩa cần lưu trên đĩa thường có kích thước rất khác nhau $\Rightarrow$ cần nhiều cluster mới chứa đủ. Nếu việc quản lý 1 dữ liệu có nghĩa được chứa trên bao nhiêu cluster đĩa và chỉ số cụ thể là gì được giao cho người dùng thì họ sẽ gặp rất nhiều rắc rối $\Rightarrow$ cần 1 giao tiếp sử dụng khác để sử dụng đĩa dễ dàng hơn.



cuu duong than cong . com

Hệ thống file (File System)

- ❑ disk luận lý cấp #2 là không gian dữ liệu 1 chiều : disk = danh sách nhiều đơn vị chứa tin có độ dài thay đổi theo yêu cầu của người dùng, mỗi đơn vị được gọi là file và được nhận dạng bằng tên gọi nhớ chứ không phải là chỉ số khó nhớ.
- ❑ Ở cấp độ này, muốn truy xuất 1 file, ta chỉ cần xác định tên gọi nhớ của nó.
- ❑ Dù dữ liệu có nghĩa cần lưu trên đĩa thường có kích thước rất khác nhau, nhưng chỉ cần 1 file là đủ để lưu 1 dữ liệu có nghĩa $\Rightarrow$ Việc quản lý dữ liệu trên đĩa trở nên dễ dàng hơn nhiều so với trước.
- ❑ Tuy nhiên vì 1 đĩa chứa 1 số rất lớn file (hàng triệu file) $\Rightarrow$ nếu dùng không gian phẳng để tổ chức các file thì cũng còn nhiều khó khăn trong việc đặt tên file, việc phân biệt các file của chương trình nào, của người nào $\Rightarrow$ cần 1 giao tiếp sử dụng khác để sử dụng đĩa dễ dàng hơn nữa.



Hệ thống file (File System)

- ❑ disk luận lý cấp #3 là không gian dữ liệu dạng cây phân cấp : disk = thư mục gốc chứa nhiều phần tử con, mỗi phần tử con có thể là file hay thư mục khác...
- ❑ Trong cấp độ này, ta nhận dạng 1 phần tử bằng khái niệm đường dẫn (pathname). Có 2 loại pathname : tuyệt đối và tương đối. Tùy thuộc vào nhu cầu sử dụng cụ thể mà dạng nào sẽ thích hợp hơn.



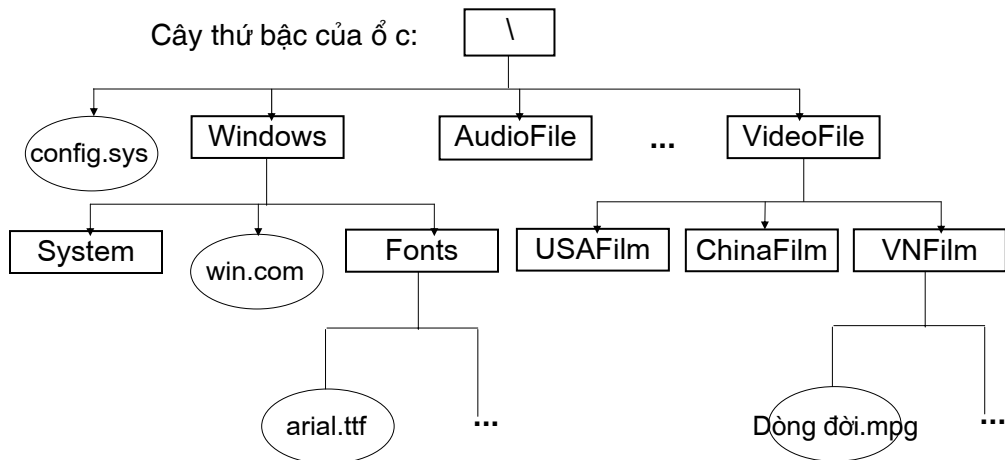
cuu duong than cong . com

Đường dẫn tuyệt đối và tương đối

- Đường dẫn (pathname) là thông tin để tìm kiếm (xác định) 1 phần tử từ 1 vị trí nào đó, nó chứa danh sách chính xác các tên gọi nhớ của các phần tử mà ta phải đi qua xuất phát từ vị trí đầu để đến phần tử cần tìm.
- ta dùng 1 dấu ngăn đặc biệt để ngăn cách 2 tên gọi nhớ liên tiếp nhau trong đường dẫn (trong Windows, dấu ngăn là '\')
- Tên thư mục gốc luôn là '\'.
cuu duong than cong . com
- Có 2 khái niệm đường dẫn : đường dẫn tuyệt đối và đường dẫn tương đối. Đường dẫn tuyệt đối là đường dẫn xuất phát từ thư mục gốc, đường dẫn tương đối xuất phát từ thư mục làm việc (working directory).
- Trước khi **ứng dụng** bắt đầu chạy, hệ thống sẽ khởi động thư mục làm việc cho ứng dụng (theo cơ chế nào đó). Trong quá trình thực thi, ứng dụng có quyền thay đổi thư mục làm việc theo yêu cầu riêng.



Thí dụ về hệ thống file



cuu duong than cong . com

Đường dẫn tuyệt đối và tương đối (tt)

- Xét cây thứ bậc của ổ c: trên slide trước, đường dẫn tuyệt đối sau sẽ nhận dạng chính xác file arial.ttf trong thư mục 'Fonts' :
c:\Windows\Fonts\arial.ttf
- Nếu thư mục working của chương trình hiện là c:\Windows\Fonts thì ta có thể dùng đường dẫn tương đối sau đây để xác định file arial.ttf :
arial.ttf
- Đường dẫn tuyệt đối thường dài hơn đường dẫn tương đối nhưng nó luôn có giá trị bất chấp ứng dụng đang ở thư mục working nào.
- Đường dẫn tương đối thường gọn hơn (đa số chỉ chứa tên file cần truy xuất vì ứng dụng sẽ thiết lập thư mục working là thư mục chứa các file mà ứng dụng truy xuất) nhưng chỉ có giá trị với 1 thư mục working cụ thể.
- Trong 1 vài trường hợp đặc biệt, ta phải dùng đường dẫn tương đối ngay cả nó dài và phức tạp hơn đường dẫn tuyệt đối!



Giao tiếp với thế giới ngoài

- ❑ chương trình khi hoạt động thỉnh thoảng phải giao tiếp với thế giới ngoài (thí dụ cần in ra máy in, cần giao tiếp mạng, cần truy xuất thông tin của các sensor đo thông số,..). Máy tính sẽ dùng 1 card chức năng (card I/O) để giao tiếp với thế giới ngoài.
- ❑ Có rất nhiều hãng sản xuất, mỗi hãng sản xuất rất nhiều model card I/O khác nhau, để đoạn code chương trình giao tiếp với I/O độc lập hoàn toàn với tính chất phần cứng của card I/O, ta sẽ xây dựng 1 module phần mềm đặc biệt : **device driver**. Mỗi card I/O có device driver riêng. Device driver phải chứa n hàm chức năng theo qui định của HĐH, chi tiết của từng hàm chức năng sẽ phụ thuộc vào phần cứng, còn việc sử dụng các hàm chức năng trong chương trình thì hoàn toàn độc lập với phần cứng.



cuu duong than cong . com

An ninh hệ thống

- ❑ Máy tính có rất nhiều tài nguyên và cho phép nhiều người truy xuất ⇒ Cần phải có cơ chế đảm bảo việc dùng tài nguyên bởi các người dùng, không cho phép việc truy xuất bất hợp pháp.
- ❑ **An ninh hệ thống** bao gồm 3 vấn đề chính :
 - **Bảo mật dữ liệu** : mỗi người chỉ được phép truy xuất 1 số tài nguyên qui định, không có khả năng truy xuất các tài nguyên khác.
 - **Toàn vẹn dữ liệu** : việc truy xuất tài nguyên của người dùng không được làm hư hỏng dữ liệu, dù chỉ 1 phần nhỏ.
 - **Sẵn sàng dữ liệu** : việc truy xuất tài nguyên của người dùng hợp pháp phải luôn được phục vụ trong khoảng thời gian ngắn nhất, bất cứ lúc nào, bất cứ ở đâu.
- ❑ Các biện pháp để bảo mật dữ liệu là quản lý người dùng theo account và mật mã hóa dữ liệu.



1.6 Các lời gọi dịch vụ HĐH "System call"

- ❑ HĐH cung cấp 1 giao tiếp sử dụng được gọi là "System Call", mỗi system call là 1 hàm thực hiện 1 chức năng xác định.
- ❑ Thường chỉ có code chương trình mới gọi System call, còn người dùng đầu cuối không thể gọi system call trực tiếp được.
- ❑ Người dùng đầu cuối sử dụng các dịch vụ HĐH gián tiếp thông qua từng ứng dụng cụ thể. Thí dụ để thực hiện các chức năng quản lý hệ thống file, người dùng trên Windows sẽ dùng trình "Windows Explorer", thông qua giao tiếp sử dụng đồ họa trực quan của chương trình, người dùng sẽ thực hiện các thao tác quản lý hệ thống file rất dễ dàng (duyệt file, tạo/hiệu chỉnh/xóa file/thư mục, di chuyển file/thư mục từ nơi này đến nơi khác,...)



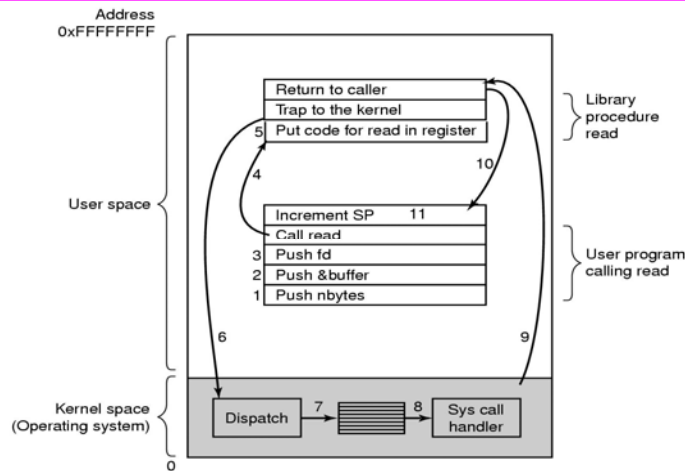
cuu duong than cong . com

Các lời gọi dịch vụ HĐH "System call"

- ❑ Gọi system call gần giống với gọi hàm bình thường, sự khác biệt lớn nhất là có sự thay đổi quyền truy xuất tài nguyên :
 - trước khi gọi system call, các lệnh của chương trình ứng dụng có quyền thấp.
 - khi gọi system call, quyền sẽ thay đổi thành rất cao (quyền hệ thống) để đoạn code của hàm "system call" có thể thực thi được chức năng đặc biệt của mình.
 - Sau khi gọi system call xong, quyền truy xuất được trả về mức cũ (thấp) của ứng dụng để đoạn code đi theo sau lệnh gọi system call chạy như cũ.



Các lời gọi dịch vụ HĐH "System call"



There are 11 steps in making the system call
read(fd, buffer, nbytes)



cuu duong than cong . com

Các lời gọi dịch vụ HĐH "System call"

Thí dụ 1 số
lời gọi dịch
vụ HĐH
"System
call" trên
Windows
và Linux :

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time



1.7 Kiến trúc của HĐH

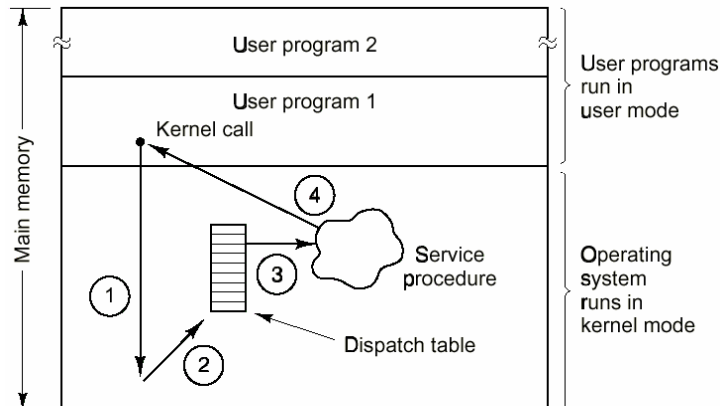
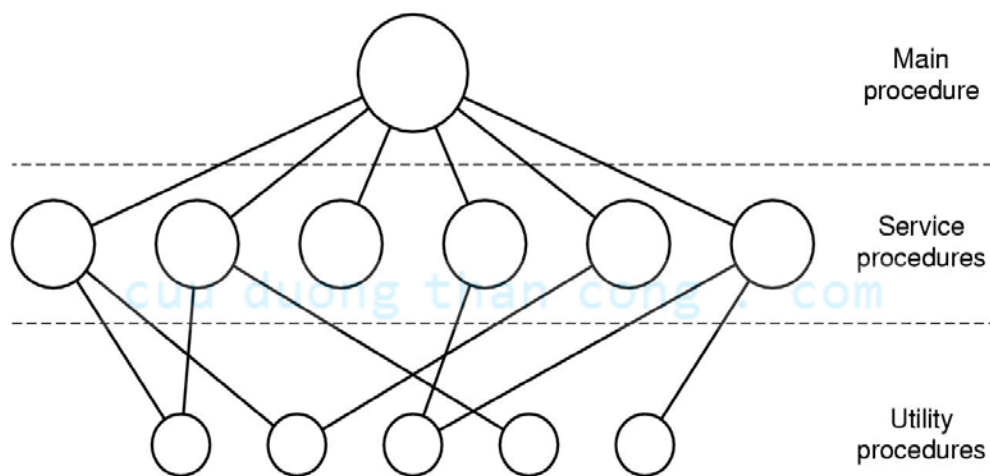


Figure 1-16. How a system call can be made: (1) User program traps to the kernel. (2) Operating system determines service number required. (3) Operating system calls service procedure. (4) Control is returned to user program.

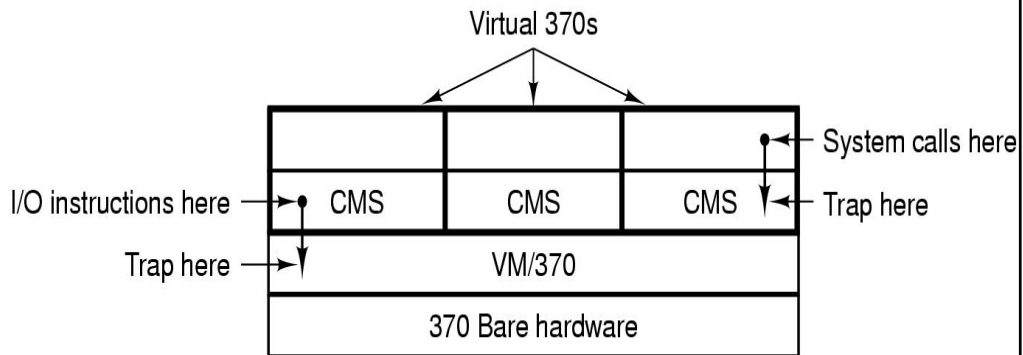


cuu duong than cong . com

Kiến trúc phân cấp

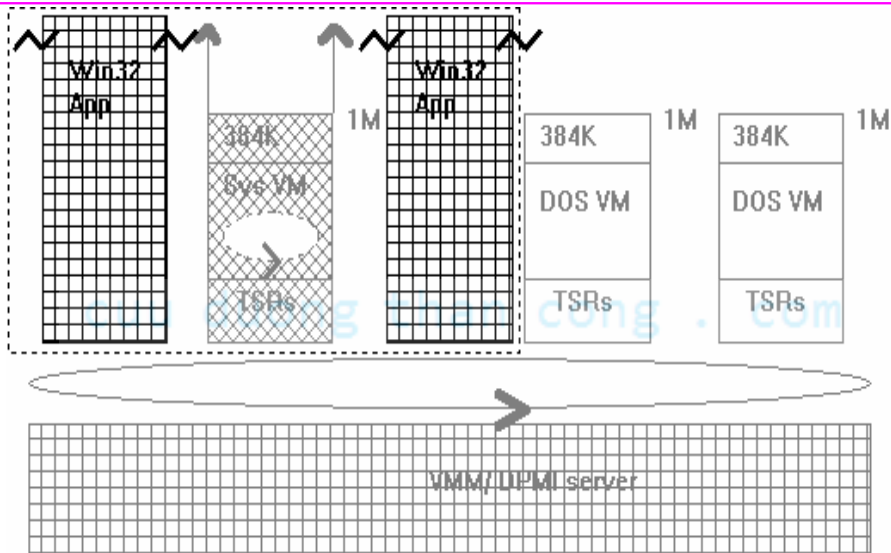


Kiến trúc máy ảo

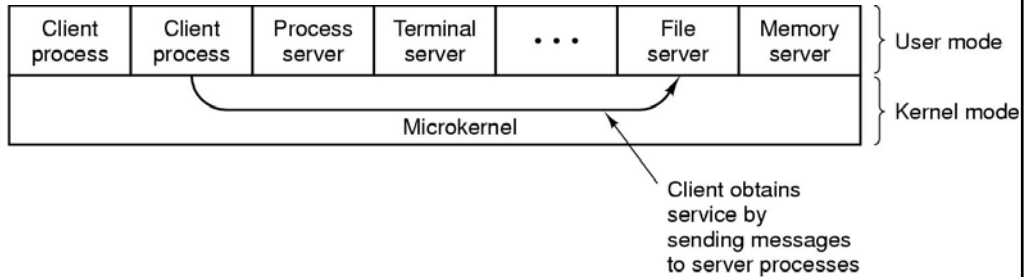


cuu duong than cong . com

Thí dụ về kiến trúc máy ảo



Kiến trúc client-server



cuu duong than cong . com

Kiến trúc client-server

