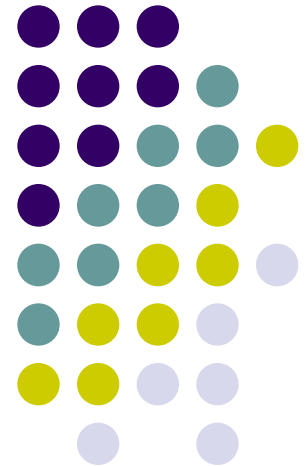


Nguyên lý hệ điều hành

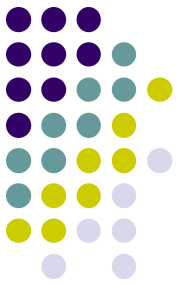
Lê Đức Trọng
DS&KTLab – Khoa CNTT
Trường Đại học Công nghệ
(Slide credit: PGS. TS. Nguyễn Hải Châu)





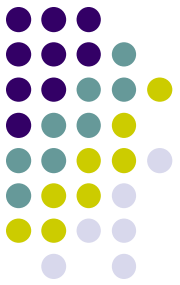
Mục tiêu của môn học

- Cung cấp những khái niệm cơ bản về hệ điều hành máy tính: phân loại, nguyên lý, cách làm việc, phân tích thiết kế và chi tiết về một số hệ điều hành cụ thể
- Yêu cầu sinh viên: Nắm vững các nguyên lý cơ bản, làm tốt các bài tập để lấy đó làm cơ sở - nguyên lý cho các vấn đề khác trong thiết kế và cài đặt các hệ thống thông tin
- Chú ý liên hệ nội dung môn học với các tình huống thực tế về khía cạnh quản lý, tổ chức



Nội dung

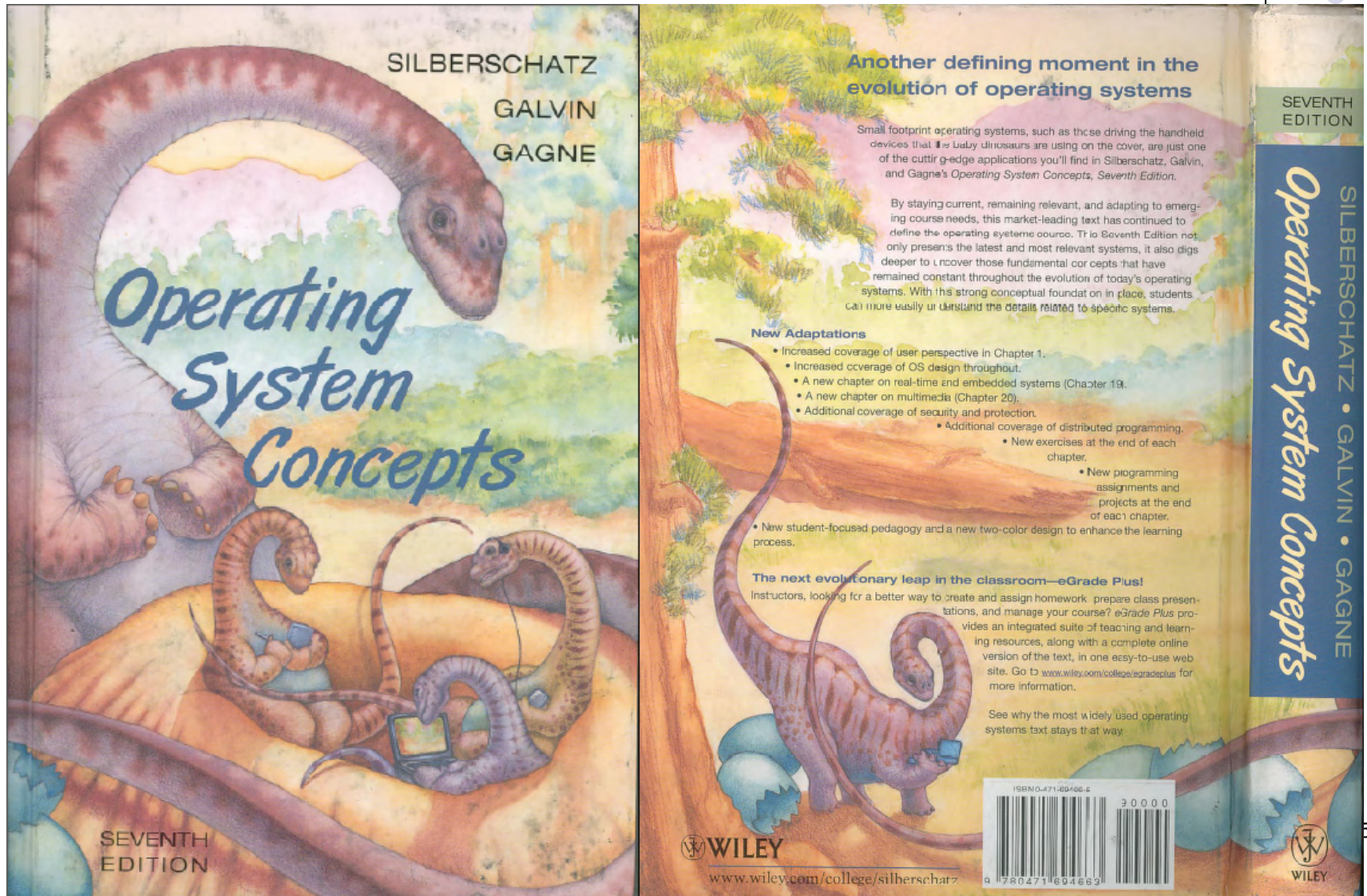
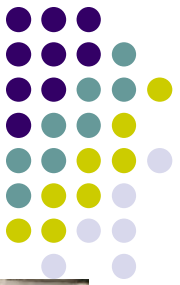
- Gồm có 6 phần chính:
 - Tổng quan (3 tiết)
 - Quản lý tiến trình (12 tiết)
 - Quản lý lưu trữ (12 tiết)
 - Hệ vào/ra (9 tiết)
 - Bảo vệ và an ninh (6 tiết)
 - Hệ điều hành Linux (*optional*) + Ôn tập (3 tiết)

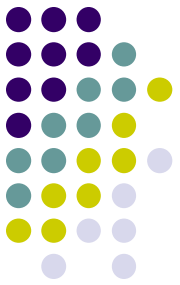


Tài liệu tham khảo

- **Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts, 7th edition, John Wiley & Sons, Inc., 2005.**
- William Stallings, Operating Systems: Internals and Design Principles 5th edition, Prentice-Hall, 2005.
- Andrew S. Tanenbaum, Modern Operating Systems, 2nd edition, Prentice-Hall, 2001.
- Andrew S. Tanenbaum, Albert S Woodhull, Operating Systems: Design and Implementation, 3rd edition, Prentice-Hall. 2006. (Có mã nguồn kèm theo).
- **Hà Quang Thụy, Nguyên lý hệ điều hành, NXB KHKT, 2002.**
- Robert Love, Linux Kernel Development, Sams Publishing, 2003.
- Daniel P. Bovet, Marco Cesati, Understanding Linux Kernel, 2nd edition, O'Reilly & Associates, 2002.
- W. Richard Stevens, Advanced Programming in the UNIX Environment, Addison-Wesley, 1992.

Giáo trình

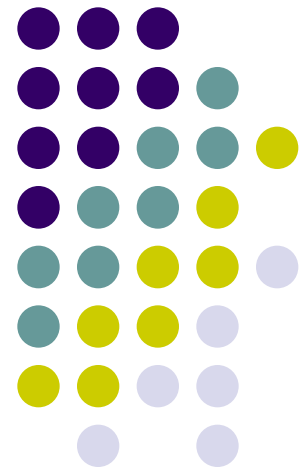




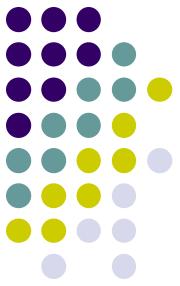
Đánh giá, thi và kiểm tra

- Chuyên cần, kiểm tra hàng tuần (15%)
- Kiểm tra giữa kỳ (25%)
 - Thi viết (45-60 phút)
 - Là điều kiện bắt buộc để được thi cuối kỳ
 - Tuần 8, sau phần quản lý bộ nhớ/lưu trữ
 - Được sử dụng tài liệu
- Thi cuối kỳ (60%)
 - Thi viết (60-90 phút)
 - Kết hợp trắc nghiệm khách quan và tự luận

Giới thiệu

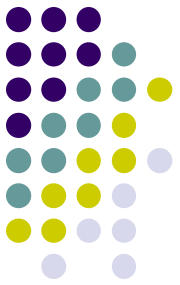


Máy tính - tài nguyên máy tính



- Tài nguyên:
 - CPU
 - Bộ nhớ trong
 - Ổ đĩa cứng
 - Thiết bị ngoại vi (máy in, màn hình, bàn phím, card giao tiếp mạng, USB...)

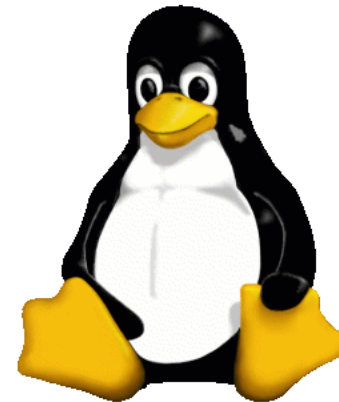




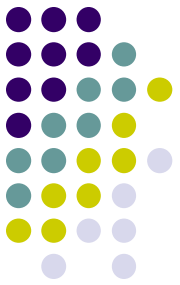
Hệ điều hành là gì?



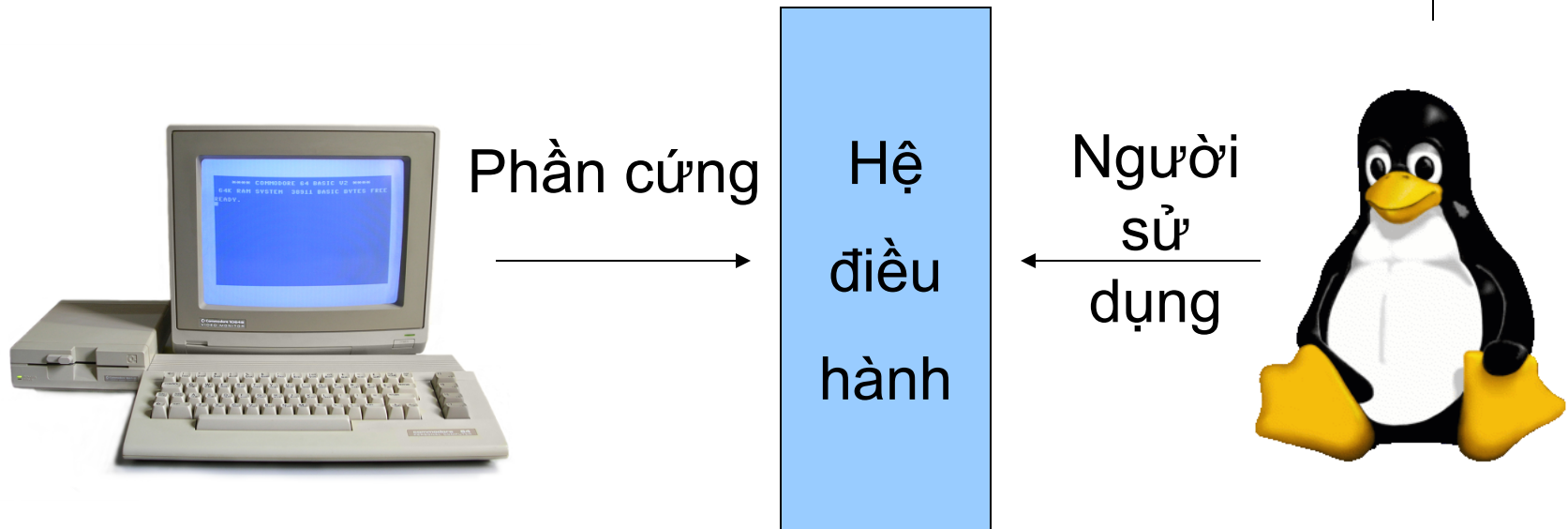
Hệ
điều
hành



- Hệ điều hành là một chương trình “*trung gian*” (nhân – kernel) giữa NSD và máy tính :
 - *Quản lý* phần cứng máy tính (các tài nguyên)
 - Cung cấp cho NSD môi trường làm việc *tiện lợi* và *hiệu quả*

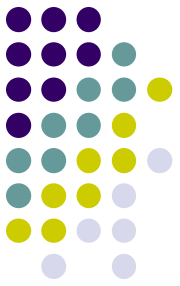


Hai cách nhìn hệ điều hành



- Phần cứng: Quản lý & cấp phát tài nguyên để sử dụng tối đa năng lực phần cứng
- Người sử dụng: Dễ sử dụng, hiệu quả, ứng dụng phong phú

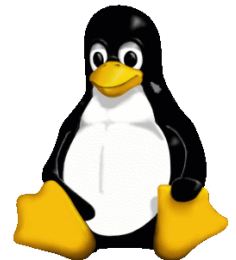
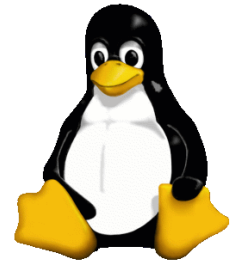
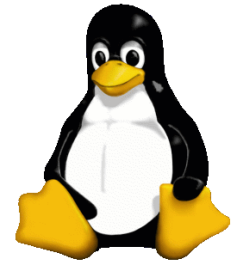
Hệ thống máy tính



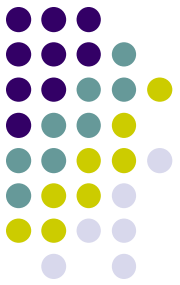
Phần cứng

Hệ
điều
hành

Các
chương
trình
hệ thống
và
ứng dụng



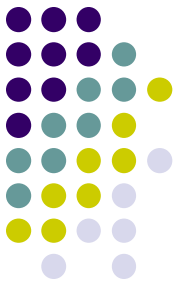
Người sử dụng



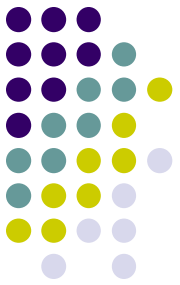
Một số loại hệ điều hành

- Xử lý theo lô (batch processing)
- Đa chương trình (multiprogramming)
- Phân chia thời gian (time-sharing/multitasking)
- Hệ điều hành cho máy cá nhân
- Xử lý song song (parallel)
- Thời gian thực (real-time)
- Nhúng (embedded)
- Cầm tay (portable)
- Đa phương tiện (multimedia)
- Chuyên dụng (special-purpose)

Các hệ xử lý theo lô đơn giản

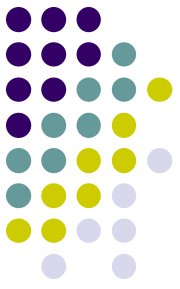


- Thuật ngữ: *Batch processing*
- Các chương trình được đưa vào hàng chờ
- Máy tính thực hiện tuần tự các chương trình của người sử dụng
- Chương trình không có giao tiếp với người sử dụng



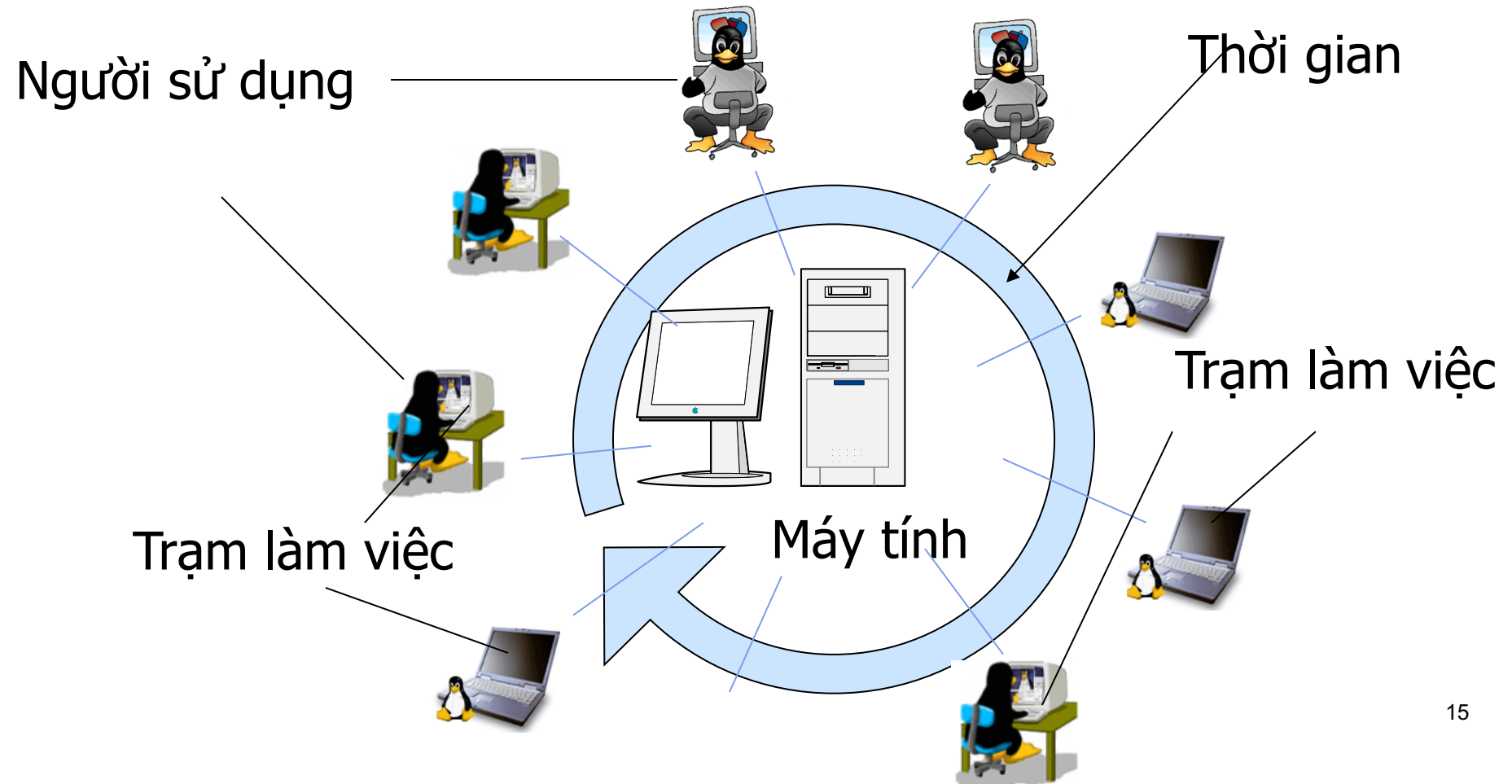
Đa chương trình

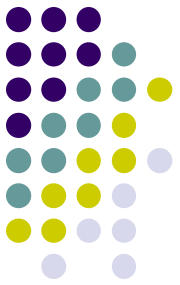
- Thuật ngữ: *Multiprogramming*
- Các chương trình được xếp vào hàng đợi
- Một chương trình được thực hiện và chiếm giữ CPU cho đến khi (1) có yêu cầu vào/ra, hoặc (2) kết thúc
- Khi (1) hoặc (2) xảy ra, chương trình khác sẽ được thực hiện
- Tận dụng CPU tốt hơn xử lý theo lô đơn giản



Phân chia thời gian/đa nhiệm

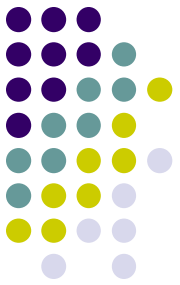
- Thuật ngữ: *time-sharing* hoặc *multitasking*





Một số hệ điều hành

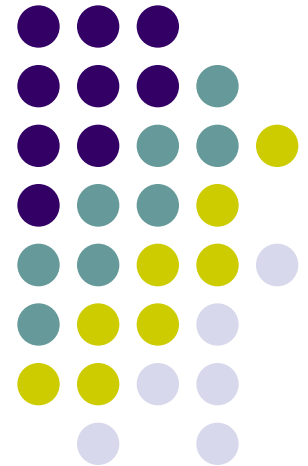
- UNIX (UNiplexed Information and Computing Service): (1) AT&T System V (2) Berkeley (BSD)
 - AIX dựa trên System V (IBM)
 - HP-UX dựa trên BSD (Hewlett-Packard)
 - IRIX dựa trên System V (Silicon Graphics Inc.)
 - Linux
 - Solaris, SunOS (Sun Microsystems)
 - Xenix (Microsoft)

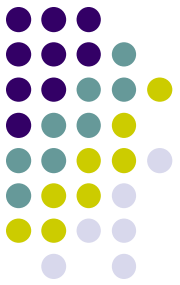


Một số hệ điều hành

- Windows (Microsoft): Windows 3.x, Windows 95, Windows 98, Windows 2000, Windows NT, Windows XP, Vista, 7, 10
- Apple: Mac OS, Mac OS X, iOS, iPadOS
- BeOS (Be Inc. 1991) => Haiku
- DOS
- PalmOS, Symbian
- Android

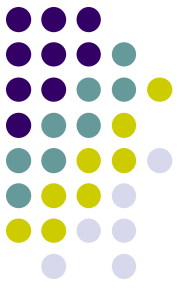
Cấu trúc hệ điều hành





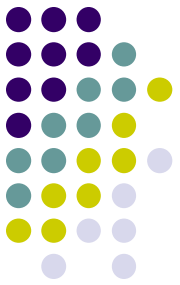
Các thành phần của hệ thống

- Quản lý tiến trình
- Quản lý bộ nhớ trong
- Quản lý tệp
- Quản lý vào/ra
- Quản lý lưu trữ trên bộ nhớ ngoài
- Liên kết mạng
- Bảo vệ và an ninh
- Thông dịch lệnh



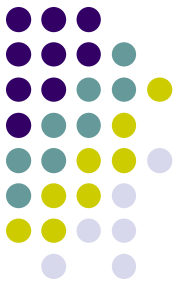
Các dịch vụ của hệ điều hành

- Giao diện với người sử dụng
- Thực hiện các chương trình
- Thực hiện các thao tác vào/ra
- Quản lý hệ thống tệp
- Kết nối
- Phát hiện lỗi
- Cấp phát tài nguyên
- Đưa ra các cơ chế bảo vệ và an ninh



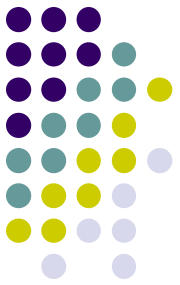
Các hàm hệ thống

- Các *hàm hệ thống* (system calls) cung cấp giao diện lập trình tới các dịch vụ do hệ điều hành cung cấp
- Ví dụ trong hệ điều hành Unix:
 - Tạo một tiến trình mới: **fork()**;
 - Thoát khỏi tiến trình đang thực hiện: **exit(1)**;
 - **fork** và **exit** là các hàm hệ thống (Hàm HT)



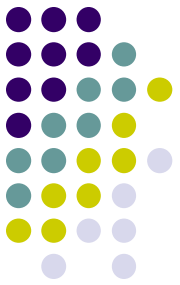
Hàm HT điều khiển tiến trình

- Kết thúc tiến trình bình thường/bất thường
- Nạp, thực hiện tiến trình
- Tạo, kết thúc tiến trình
- Đọc hoặc thiết lập các thuộc tính cho tiến trình
- Yêu cầu tiến trình vào trạng thái chờ
- Cấp phát và giải phóng bộ nhớ
- Xử lý các sự kiện không đồng bộ



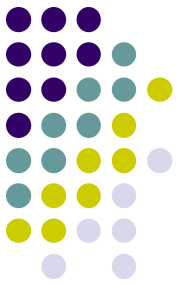
Hàm HT quản trị tệp

- Tạo, xóa tệp
- Đóng, mở tệp
- Đọc, ghi, định vị con trỏ tệp
- Đọc, thiết lập thuộc tính của tệp



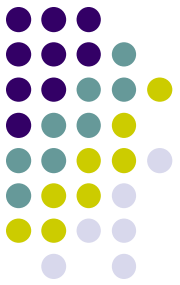
Hàm HT quản trị thiết bị

- Yêu cầu sử dụng hoặc thôi sử dụng thiết bị
- Đọc, ghi, định vị con trỏ
- Đọc, thiết lập thuộc tính cho thiết bị
- Attach/detach thiết bị về mặt logic



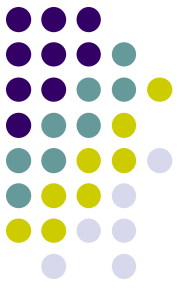
Hàm HT bảo trì thông tin

- Đọc, thiết lập thời gian hệ thống
- Đọc, ghi dữ liệu về hệ thống
- Đọc thuộc tính tệp, thiết bị, tiến trình
- Thiết lập thuộc tính tệp, thiết bị, tiến trình



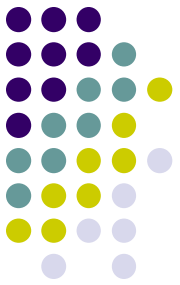
Hàm HT về kết nối

- Tạo, hủy các kết nối mạng
- Truyền nhận các thông điệp
- Lấy thông tin trạng thái kết nối
- Attach/detach các thiết bị ở xa



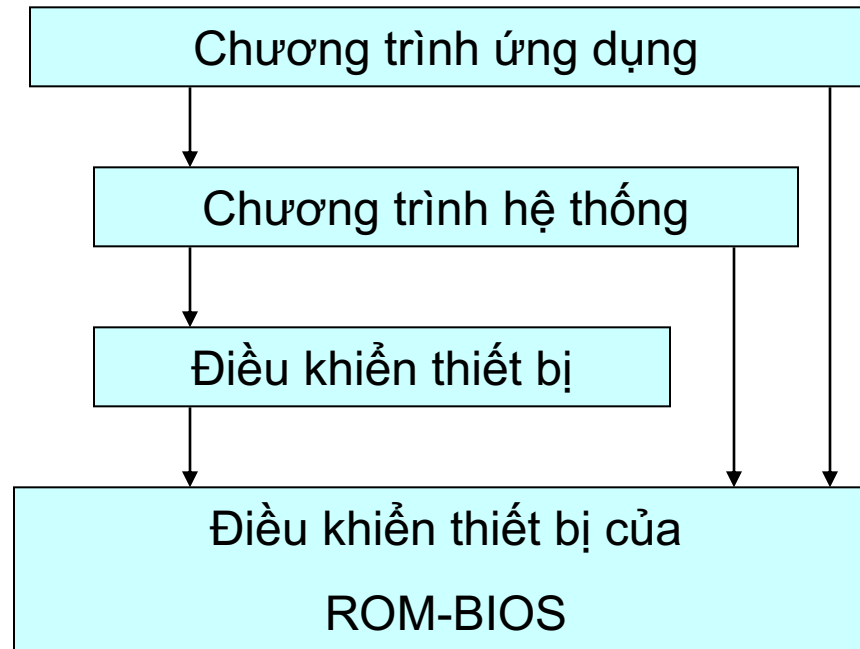
Các chương trình hệ thống

- Các chương trình hệ thống cung cấp môi trường thuận tiện cho việc thực hiện và phát triển chương trình, được phân loại như sau:
 - Thao tác với tệp
 - Thông tin về trạng thái của hệ thống
 - Sửa đổi tệp
 - Hỗ trợ ngôn ngữ lập trình
 - Nạp và thực hiện chương trình
 - Kết nối
 - Cách nhìn HĐH của NSD được xác định qua các chương trình hệ thống, không thực sự qua các hàm hệ thống (system calls).

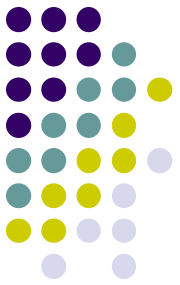


Cấu trúc HĐH: Đơn giản

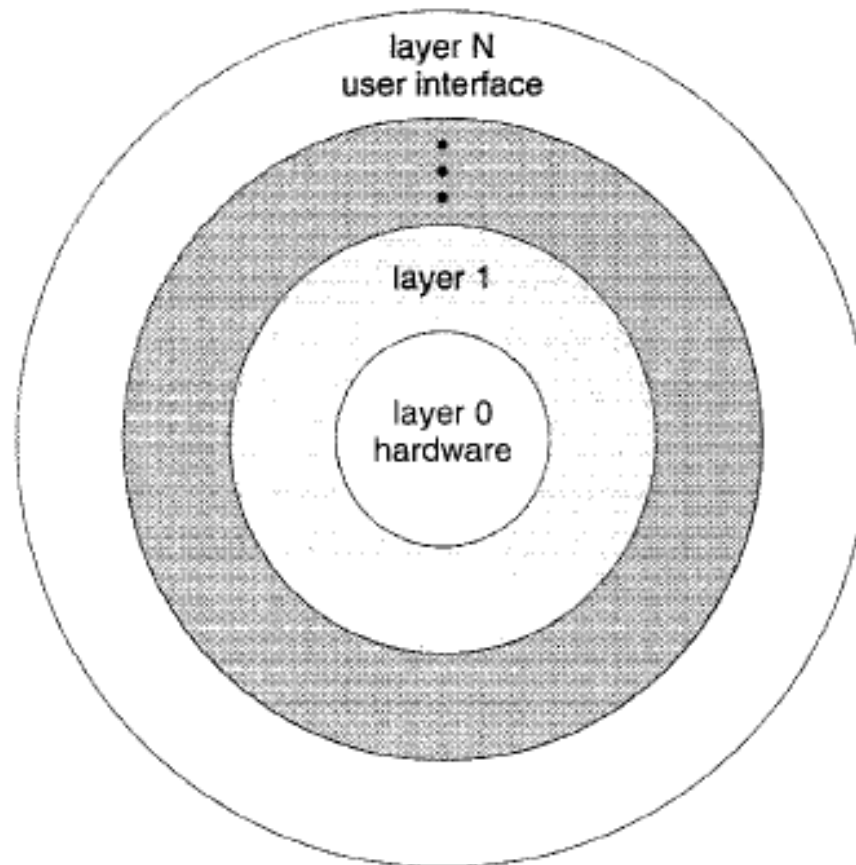
- Thuật ngữ: Simple approach
- Ví dụ MS-DOS. (tương tự: UNIX thời gian đầu)



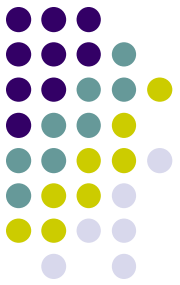
Cấu trúc HĐH: Phân tầng



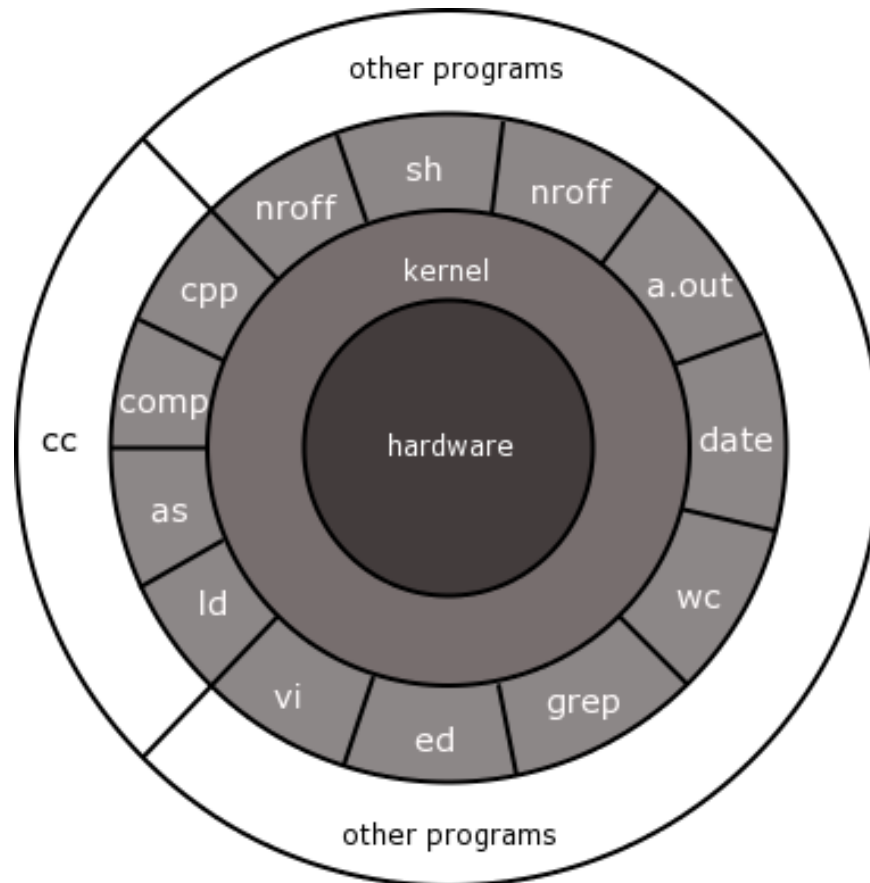
- Thuật ngữ: Layered approach



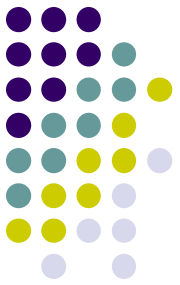
Cấu trúc HĐH: Phân tầng



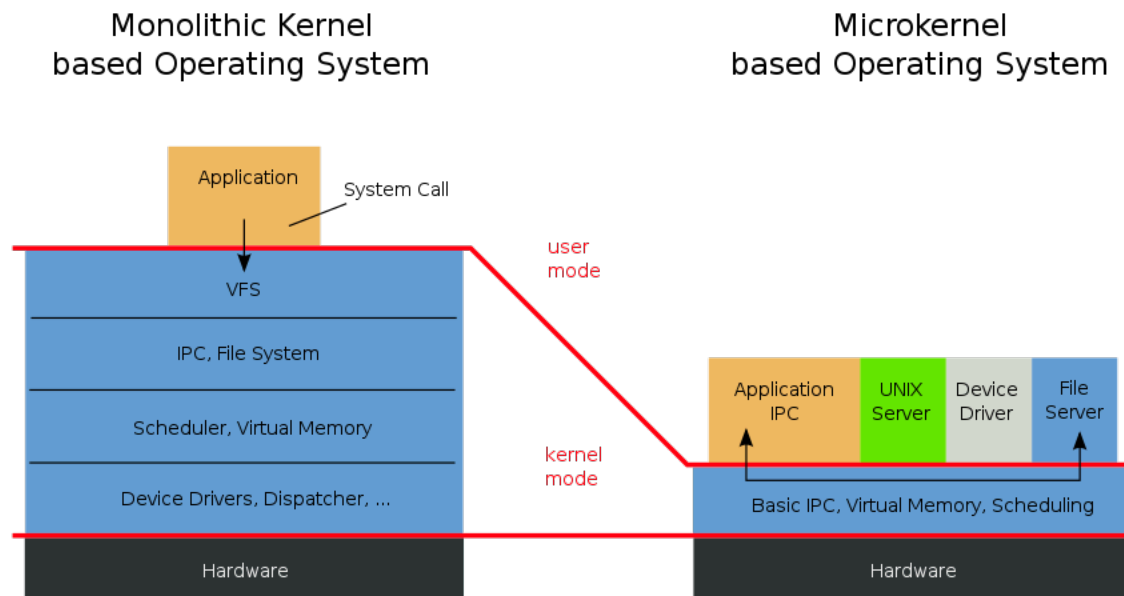
- Ví dụ UNIX



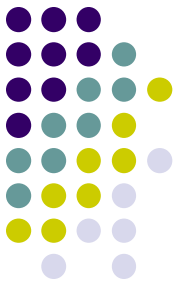
Cấu trúc HĐH: Vi nhân



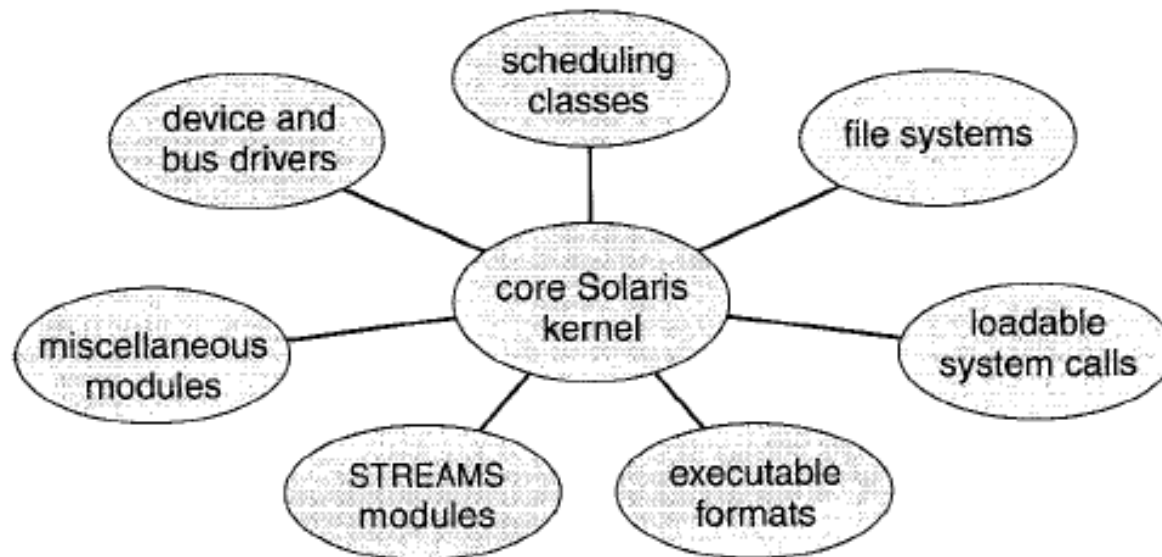
- Thuật ngữ: Microkernel
- Giữ cho nhân có các đủ các chức năng thiết yếu nhất để giảm cỡ
- Các chức năng khác được đưa ra ngoài nhân



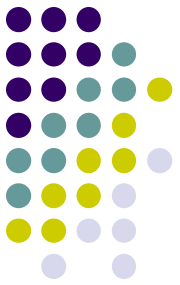
Cấu trúc HĐH: Module



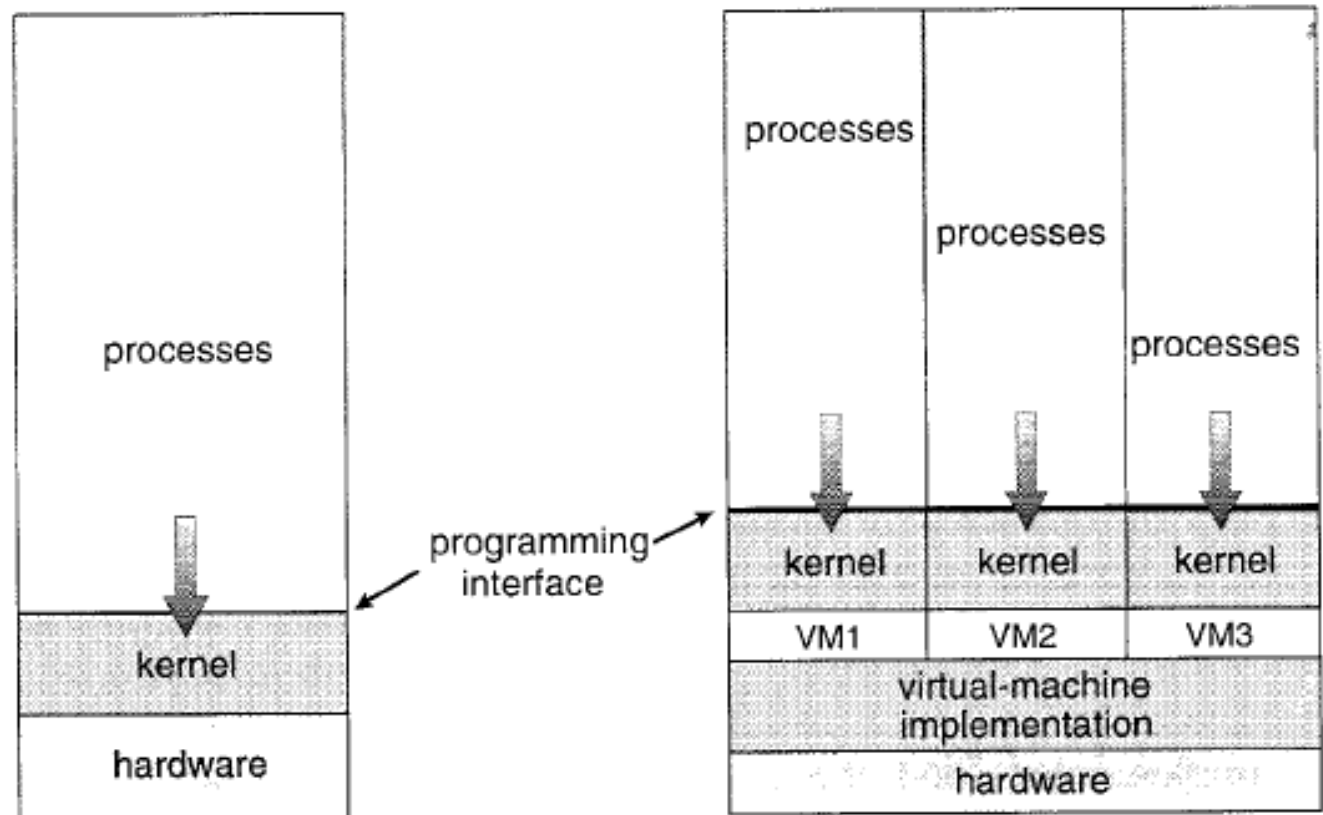
- Thuật ngữ: Module approach
- Hiện tại đây là cách tiếp cận tốt nhất (sử dụng được các kỹ thuật lập trình hướng đối tượng). Ví dụ: Solaris của Sun Microsystem:

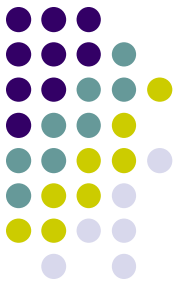


Máy ảo



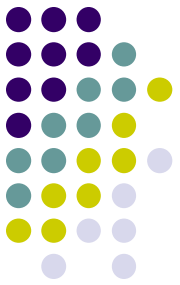
- Thuật ngữ (Virtual Machine)
- Ví dụ: VMware (sản phẩm thương mại)





Tóm tắt

- Khái niệm HĐH, nhân
- Hai cách nhìn HĐH từ NSD và hệ thống
- Các khái niệm xử lý theo lô, đa chương trình và phân chia thời gian
- Các thành phần và dịch vụ của HĐH
- Các hàm hệ thống
- Một số cấu trúc phổ biến của HĐH
- Máy ảo



Tìm hiểu thêm

- **Không bắt buộc**
- Bổ sung một hàm hệ thống mới vào nhân Linux và sử dụng hàm đó:
 - Đọc hướng dẫn trong giáo trình từ trang 74-78
 - Thử nghiệm trên RedHat Fedora hoặc Ubuntu/Debian