



HỆ ĐIỀU HÀNH

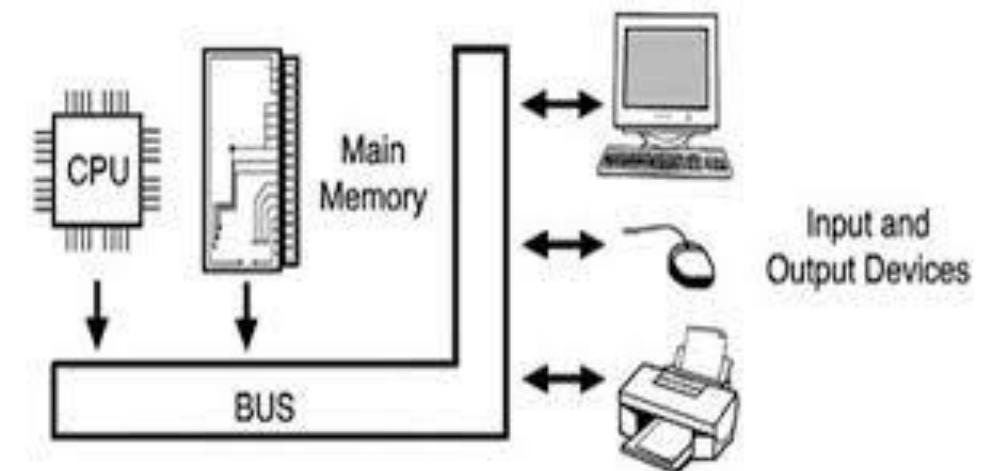
CHƯƠNG 9: HỆ ĐIỀU HÀNH LINUX VÀ HỆ ĐIỀU HÀNH WINDOWS

Trình bày các kiến thức cơ bản về hệ điều hành Linux và Windows



Các nội dung đã học

- Chương 1: Tổng quan về hệ điều hành
- Chương 2: Cấu trúc hệ điều hành
- Chương 3: Quản lý tiến trình
- Chương 4: Định thời CPU
- Chương 5: Đồng bộ hóa tiến trình
- Chương 6: Tắc nghẽn
- Chương 7: Quản lý bộ nhớ
- Chương 8: Bộ nhớ ảo
- **Chương 9: Hệ điều hành Linux và Hệ điều hành Windows**





MỤC TIÊU

1. Hiểu được các kiến thức cơ bản về hệ điều hành Linux và Windows.
2. Phân tích, so sánh, đánh giá các kiến thức đã học và cách các kiến thức đó được áp dụng vào hệ điều hành Linux và Windows.



NỘI DUNG

1. Lịch sử phát triển
2. Nguyên tắc thiết kế
3. Các thành phần chính
4. Quản lý tiến trình
5. Định thời
6. Giao tiếp liên tiến trình
7. Quản lý bộ nhớ



Hệ điều hành Linux

9.1.1 Tổng quan về Linux



1



9.1.1 Tổng quan về Linux

- Linux là một hệ điều hành hiện đại, miễn phí dựa trên UNIX.
 - Nhân Linux bắt đầu được phát triển bởi Linus Torvalds từ 1991, với mục đích ban đầu để tương thích với UNIX, được phát hành dưới dạng mã nguồn mở.
- Linux được thiết kế để hoạt động hiệu quả trên PC và nhiều nền tảng phần cứng khác, được phát triển và duy trì bởi nhiều người dùng trên thế giới.
- Phân biệt:
 - Nhân Linux: Phần mềm gốc được phát triển từ đầu bởi cộng đồng Linux.
 - Hệ thống Linux (Linux system): Tập hợp các thành phần được phát triển/hợp tác phát triển bởi các dự án và các cộng đồng khác.
- Linux có nhiều bản phân phối (distribution) khác nhau bao gồm nhân, các ứng dụng và công cụ quản lý.



9.1.1 Tổng quan về Linux

Quá trình phát triển nhân Linux

- Phiên bản 0.01 (5/1991) không có kết nối mạng, chỉ chạy trên PC với bộ xử lý Intel 80386.
- Phiên bản 1.0 (3/1994) bổ sung nhiều chức năng mới:
 - Hỗ trợ giao thức TCP/IP
 - Giao tiếp socket tương thích BSD
 - Cải thiện hệ thống tập tin
 - Hỗ trợ thêm nhiều phần cứng
- Phiên bản 1.2 (3/1995) là phiên bản cuối cùng chỉ dành cho PC.
- Cách thức đánh số phiên bản: số lẻ là các phiên bản phát triển, số chẵn là các phiên bản chính (production).



9.1.1 Tổng quan về Linux

Quá trình phát triển nhân Linux

- Linux 2.0 được phát hành vào 6/1996 với hai cải tiến lớn:
 - Hỗ trợ nhiều kiến trúc, bao gồm cả 64 bit Alpha
 - Hỗ trợ các kiến trúc đa bộ xử lý
- Linux 2.0 hoạt động trên nhiều hệ thống khác nhau: Sun Sparc, PC, PowerMac, ...
- Các phiên bản 2.4 và 2.6 tiếp tục tăng cường hỗ trợ SMP, cải thiện hệ thống quản lý bộ nhớ với sự hỗ trợ bộ nhớ 64 bit.
- Linux 3.0 phát hành vào 7/2011 với sự cải thiện khả năng ảo hóa, quản lý bộ nhớ và định thời.
- Linux 4.0 phát hành vào 4/2015 hỗ trợ thêm nhiều kiến trúc mới, cải thiện các chức năng di động và nhiều cải tiến khác.



9.1.3 Tổng quan về Linux

Hệ thống Linux

- Các phần mềm tạo nên hệ thống Linux thường không chỉ dành riêng cho Linux mà còn được sử dụng trên nhiều hệ điều hành dựa trên UNIX khác.
 - Linux sử dụng nhiều công cụ được phát triển bởi hệ điều hành Berkeley BSD, MIT's X Window System và dự án Free Software Foundation's GNU.
- Việc sử dụng chung các công cụ này diễn ra theo cả hai hướng:
 - Linux sử dụng, đồng thời cải tiến công cụ đó
 - Hệ thống thư viện chính của Linux được bắt đầu từ dự án GNU với nhiều cải tiến được thực hiện bởi cộng đồng Linux.
 - Các dự án khác sử dụng lại công cụ được xây dựng/cải tiến bởi cộng đồng Linux
 - Các công cụ quản lý mạng trên Linux được kế thừa từ 4.3BSD, nhưng các hệ điều hành dựa trên BSD sau này như FreeBSD mượn trở lại các mã nguồn từ Linux.



9.1.1 Tổng quan về Linux

Các bản phân phối Linux

- Các bản phân phối là tập hợp các gói phần mềm đã được biên dịch và tiêu chuẩn hóa, bao gồm hệ thống Linux cơ bản, hệ thống cài đặt, các công cụ quản lý và các gói công cụ UNIX phổ biến.
- Các bản phân phối phổ biến hiện nay là RedHat (thương mại) và Debian (miễn phí). Một số bản phân phối thường gặp khác là Canonical và SuSE.
- Định dạng gói RPM có tính tương thích cao và được sử dụng bởi nhiều bản phân phối.





9.1.1 Tổng quan về Linux

Giấy phép Linux

- Nhân Linux được phân phối dưới giấy phép GNU General Public License (GPL).
- Bất kỳ ai sử dụng Linux, hoặc tạo ra phiên bản phái sinh của Linux, không được để sản phẩm đó là độc quyền, phần mềm được phát hành dưới giấy phép GPL không thể được tái phân phối chỉ dưới dạng nhị phân.
 - Có thể bán các bản phân phối, nhưng phải cung cấp kèm mã nguồn.



9.1.1 Tổng quan về Linux

Nguyên tắc thiết kế Linux

- Linux là một hệ thống nhiều người dùng, đa tác vụ với một tập đầy đủ các công cụ tương thích với UNIX.
 - Hệ thống tập tin tuân thủ theo UNIX. Linux cũng cài đặt đầy đủ mô hình mạng tiêu chuẩn của UNIX.
 - Giao diện lập trình tuân thủ theo SVR4 UNIX.
- Các mục tiêu thiết kế chính là tốc độ, hiệu quả và tiêu chuẩn hóa.
- Linux được thiết kế để đáp ứng các yêu cầu POSIX liên quan.
 - Hỗ trợ Pthreads và một tập con của thư viện quản lý tiến trình real-time POSIX.



Hệ điều hành Linux

9.1.2 Các thành phần của hệ thống Linux

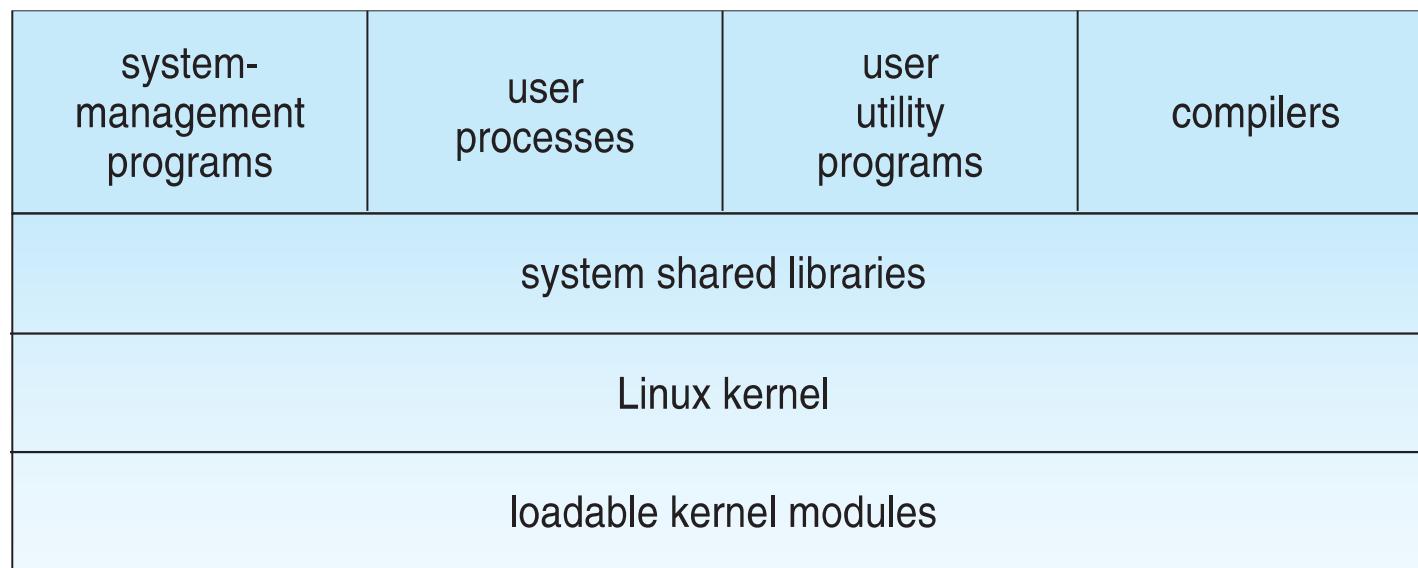


1



9.1.2 Các thành phần của hệ thống Linux

- Như phần lớn các hệ thống UNIX, hệ thống Linux bao gồm 3 thành phần chính: nhân, thư viện hệ thống và công cụ hệ thống.





9.1.2 Các thành phần của hệ thống Linux

- Nhân chịu trách nhiệm duy trì các hoạt động ở mức trừu tượng của hệ điều hành, như bộ nhớ ảo và tiến trình.
 - Các mã của nhân thực thi ở kernel mode với đầy đủ quyền truy xuất đến tài nguyên vật lý của máy tính.
 - Tất cả các mã của nhân và cấu trúc dữ liệu được lưu trên cùng một không gian địa chỉ
- Các thư viện hệ thống (system libraries) định nghĩa một tập chuẩn các hàm mà thông qua đó các ứng dụng có thể tương tác với nhân. Các thư viện này cài đặt nhiều chức năng của hệ điều hành mà không cần đầy đủ quyền (privileges) như các mã của nhân.
- Các công cụ hệ thống (system utilities) là các chương trình thực hiện các chức năng quản lý cụ thể.



Hệ điều hành Linux

9.1.3 Module nhân (kernel modules) của Linux



1



9.1.3 Module nhân (kernel modules) của Linux

- Module nhân: Các phần mã của nhân có thể biên dịch, nạp và gỡ độc lập với phần còn lại của nhân.
- Một module nhân có thể cài đặt một trình điều khiển thiết bị, một hệ thống tập tin hoặc một giao thức mạng.
- Các module nhân cho phép thiết lập một hệ thống Linux với một nhân Linux tiêu chuẩn tối thiểu mà không cần bắt cứ trình điều khiển thiết bị đi kèm.
- Các giao thức của module có thể cho phép bên thứ ba viết và phân phối trình điều khiển thiết bị hoặc hệ thống tập tin của họ, vốn không thể phân phối dưới giấy phép GPL.



9.1.3 Module nhân (kernel modules) của Linux

- Linux hỗ trợ 4 loại module sau:
 - module-management system: cho phép module nạp vào bộ nhớ và giao tiếp với phần còn lại của nhân.
 - module loader and unloader: là các công cụ ở user mode, làm việc với module-management system để nạp một module vào bộ nhớ.
 - driver-registration system: cho phép các module thông báo với phần còn lại của nhân là có một trình điều khiển mới đã sẵn sàng.
 - conflict-resolution mechanism: cho phép các trình điều khiển khác nhau chiếm lấy tài nguyên máy tính và bảo vệ tài nguyên này khỏi việc truy xuất không phù hợp từ trình điều khiển khác.



Hệ điều hành Linux

9.1.4 Quản lý tiến trình trên Linux



1



9.1.4 Quản lý tiến trình trên Linux

- Hệ thống quản lý tiến trình của UNIX phân chia việc tạo tiến trình và chạy một chương trình mới thành hai thao tác riêng biệt:
 - System call fork() tạo ra một tiến trình mới.
 - Một chương trình mới được chạy sau khi gọi exec().
- Trên UNIX, một tiến trình chứa tất cả thông tin mà hệ điều hành phải lưu trữ để lưu vết ngũ cảnh của một thao tác thực thi của một chương trình đơn.
- Trên Linux, các thuộc tính của tiến trình được chia thành 3 nhóm: định danh của tiến trình, môi trường và ngũ cảnh.



9.1.4 Quản lý tiến trình trên Linux

Giao tiếp liên tiến trình trên Linux

- Tương tự UNIX, Linux thông báo cho các tiến trình có một sự kiện đã xảy ra thông qua các signal.
- Số lượng signal là giới hạn và chúng không chứa thông tin.
- Signal có thể được tạo ra bởi tiến trình hoặc nhân. Tuy nhiên, nhân không dùng signal để giao tiếp với tiến trình đang chạy ở kernel mode, thay vào đó, nó sử dụng các trạng thái định thời và cấu trúc *wait_queue*.



9.1.4 Quản lý tiến trình trên Linux

Chuyển dữ liệu giữa các tiến trình trên Linux

- Pipe: Cho phép tiến trình con kế thừa một kênh giao tiếp từ tiến trình cha của nó. Dữ liệu ghi ở một đầu của pipe có thể đọc ở đầu còn lại.
- Network: UNIX cung cấp một tập các công cụ mạng để gửi dữ liệu đến các tiến trình cục bộ và từ xa.
- Shared memory: Cho phép giao tiếp dữ liệu một cách nhanh chóng. Dữ liệu được ghi bởi một tiến trình vào một vùng nhớ chia sẻ có thể được đọc ngay lập tức bởi một tiến trình khác nếu nó đã ánh xạ vùng nhớ đó vào không gian nhớ của nó.
 - Để đạt được tính đồng bộ, shared memory cần được sử dụng kết hợp với các phương thức giao tiếp khác.



9.1.4 Quản lý tiến trình trên Linux

Định thời trên Linux

- Trên Linux, định thời không chỉ là việc chạy và tạm dừng các tiến trình, mà còn bao gồm việc thực thi nhiều tác vụ trong nhân.
- Các tác vụ trong nhân bao gồm các tác vụ được yêu cầu bởi tiến trình đang chạy và các tác vụ thực thi nội tại trong trình điều khiển thiết bị.
- Phiên bản 2.5 giới thiệu và sử dụng bộ định thời O(1) dựa trên độ ưu tiên và chế độ quyết định trưng dụng.
- Phiên bản 2.6 sử dụng Completely Fair Scheduler (CFS).



Hệ điều hành Linux

9.1.5 Quản lý bộ nhớ trên Linux



1



9.1.5 Quản lý bộ nhớ trên Linux

- Quản lý bộ nhớ trên Linux gồm 2 thành phần:
 - Cấp phát và giải phóng bộ nhớ vật lý – trang, nhóm các trang và các khối RAM.
 - Xử lý bộ nhớ ảo – ánh xạ bộ nhớ vào không gian địa chỉ của các tiến trình đang chạy.



9.1.5 Quản lý bộ nhớ trên Linux

- Quản lý bộ nhớ vật lý:

- Tùy thuộc vào đặc điểm phần cứng, Linux chia bộ nhớ vật lý thành 6 vùng: DMA, DMA32, NORMAL, HIGHMEM, MOVABLE, DEVICE.
- Mỗi vùng có page allocator riêng, chịu trách nhiệm cấp phát và giải phóng tất cả các trang vật lý, cũng như cấp phát một dãy các trang vật lý liên tiếp khi được yêu cầu.
- Page allocator sử dụng giải thuật buddy heap để lưu vết các trang vật lý khả dụng.
 - Mỗi vùng nhớ cấp phát có một vùng nhớ liền kề - buddy.
 - Nếu hai vùng nhớ liền kề đều trống, chúng được kết hợp thành một vùng nhớ lớn hơn.
 - Vùng nhớ trống kích thước lớn có thể được chia thành 2 vùng nhớ để đáp ứng các yêu cầu cấp phát nhỏ.



9.1.5 Quản lý bộ nhớ trên Linux

Liên kết động trên Linux

- Linux cài đặt cơ chế liên kết động ở user mode thông qua thư viện liên kết đặc biệt:
 - Mỗi chương trình có sử dụng liên kết động chứa một hàm liên kết tĩnh nhỏ - nó sẽ được gọi khi chương trình bắt đầu chạy.
 - Hàm liên kết tĩnh ánh xạ thư viện liên kết vào bộ nhớ.
 - Thư viện liên kết xác định thư viện động được yêu cầu bởi chương trình cũng như các biến và hàm mà nó cần (từ các thư viện động này).
 - Thư viện liên kết ánh xạ các thư viện động này vào phần giữa của bộ nhớ ảo và giải quyết các tham chiếu đến thư viện động.



9.1.5 Quản lý bộ nhớ trên Linux

Hoán vị và phân trang

- Hệ thống phân trang cần phải tái định vị các trang nhớ của bộ nhớ vật lý được đưa ra xuống đĩa khi cần thêm bộ nhớ cho một tác vụ nào đó.
- Hệ thống phân trang được chia thành 2 phần:
 - Giải thuật pageout-policy: Quyết định trang nào sẽ được ghi ra đĩa và khi nào thì thực hiện.
 - Paging mechanism: Thực hiện việc di chuyển, đồng thời cũng phụ trách việc đem dữ liệu trang trở lại bộ nhớ vật lý.
 - Trang có thể được đưa đến thiết bị khác, phân vùng khác hoặc tập tin.
 - Sử dụng giải thuật next – fit để ghi các trang liên tục.



Hệ điều hành Linux

9.1.6 Bộ nhớ ảo trên Linux



1



9.1.6 Bộ nhớ ảo trên Linux

- Hệ thống bộ nhớ ảo của Linux duy trì không gian địa chỉ cho mỗi tiến trình. Nó tạo các trang ảo theo yêu cầu và quản lý việc nạp các trang ảo từ đĩa cũng như di chuyển chúng trở về đĩa khi được yêu cầu.
- Không gian địa chỉ của một tiến trình có thể chia thành 2 view riêng biệt:
 - Logical view: Mô tả các hướng dẫn liên quan đến việc bố trí không gian địa chỉ. Không gian địa chỉ chứa một tập các vùng nhớ không chồng lấn nhau, mỗi vùng nhớ là một tập các trang nhớ liên tục.
 - Physical view: Được lưu trữ trên bảng trang của tiến trình và được quản lý bởi một tập các routines.



9.1.6 Bộ nhớ ảo trên Linux

- Khi thực thi một chương trình mới, tiến trình được cung cấp một vùng nhớ ảo mới, hoàn toàn trống. Routine chịu trách nhiệm nạp chương trình sẽ nạp đầy (populate) không gian địa chỉ này với các vùng nhớ ảo.
- Quá trình tạo một tiến trình mới với fork() sẽ tạo ra một bản sao đầy đủ của không gian địa chỉ ảo của tiến trình hiện tại.
 - Nhân sẽ sao chép các thông tin cấu trúc bộ nhớ ảo của cha, sau đó tạo ra một tập các bảng trang cho con.
 - Các bảng trang của cha được sao chép trực tiếp cho con.
 - Kết thúc fork(), cha và con cùng chia sẻ cùng số trang nhớ vật lý trong không gian địa chỉ của chúng.



Hệ điều hành Windows



9.2.1 Lịch sử phát triển của Windows

2



9.2.1 Lịch sử phát triển của Windows

- Từ 1988, Microsoft bắt đầu phát triển một hệ điều hành sử dụng các công nghệ mới (new technology – NT) hỗ trợ các hàm thư viện lập trình (API) của cả OS/2 và POSIX.
- Ban đầu, NT chỉ sử dụng các hàm thư viện OS/2 làm môi trường chủ đạo (native), nhưng dần trong quá trình phát triển, NT được thay đổi để sử dụng Win32 API, theo cùng với sự phổ biến của Windows 3.0.
- Nhiều phiên bản Windows đã được phát hành và sử dụng rộng rãi: 95, 98, Me, 2000, XP, Vista, 7, 8, 8.1, 10, 11.



Hệ điều hành Windows



9.2.2 Nguyên tắc thiết kế của Windows

2



9.2.2 Nguyên tắc thiết kế của Windows

- Bảo mật:
 - Nhiều lớp bảo mật và cơ chế mã hóa: Access control lists (ACLs), Data Execution Prevention (DEP), mã hóa hệ thống tập tin, chữ ký số,...
- Đáng tin cậy: Sử dụng cơ chế bảo vệ phần cứng đối với bộ nhớ ảo và bảo vệ phần mềm đối với các tài nguyên của hệ điều hành.
- Dễ mở rộng: Được xây dựng theo kiến trúc phân tầng
 - Remote procedure calls (RPCs)
 - Advanced local procedure calls (ALPCs)



9.2.2 Nguyên tắc thiết kế của Windows

- Hiệu năng cao:
 - Các thành phần của hệ điều hành có thể giao tiếp với nhau thông qua hệ thống truyền thông điệp hiệu năng cao:
 - Cơ chế trưng dụng áp dụng cho các tiểu trình có độ ưu tiên thấp cho phép hệ thống đáp ứng nhanh với các sự kiện bên ngoài.
 - Được thiết kế theo cơ chế đa xử lý đối xứng.
 - Tương thích cao: Các ứng dụng được xây dựng theo chuẩn IEEE 1003.1 (POSIX) có thể được biên dịch để chạy mà không cần phải thay đổi mã nguồn.
 - Di động (portability): Hệ điều hành Windows có thể chạy được trên nhiều kiến trúc khác nhau mà không cần thay đổi nhiều.



9.2.2 Nguyên tắc thiết kế của Windows

- Hỗ trợ quốc tế: Hỗ trợ nhiều ngôn ngữ khác nhau thông qua national language support (NLS) API.
- Sử dụng năng lượng hiệu quả:
 - Các cơ chế và thành phần giám sát việc sử dụng pin, nhằm tăng thời gian sử dụng: dynamic tick, process lifetime management, desktop activity monitor, connected standby, ...
- Hỗ trợ nhiều thiết bị:
 - Tự động nhận biết loại thiết bị.
 - Cơ chế thêm, xóa và thay thế nóng CPU, RAM.



Hệ điều hành Windows



9.2.3 Kiến trúc của Windows 10

2

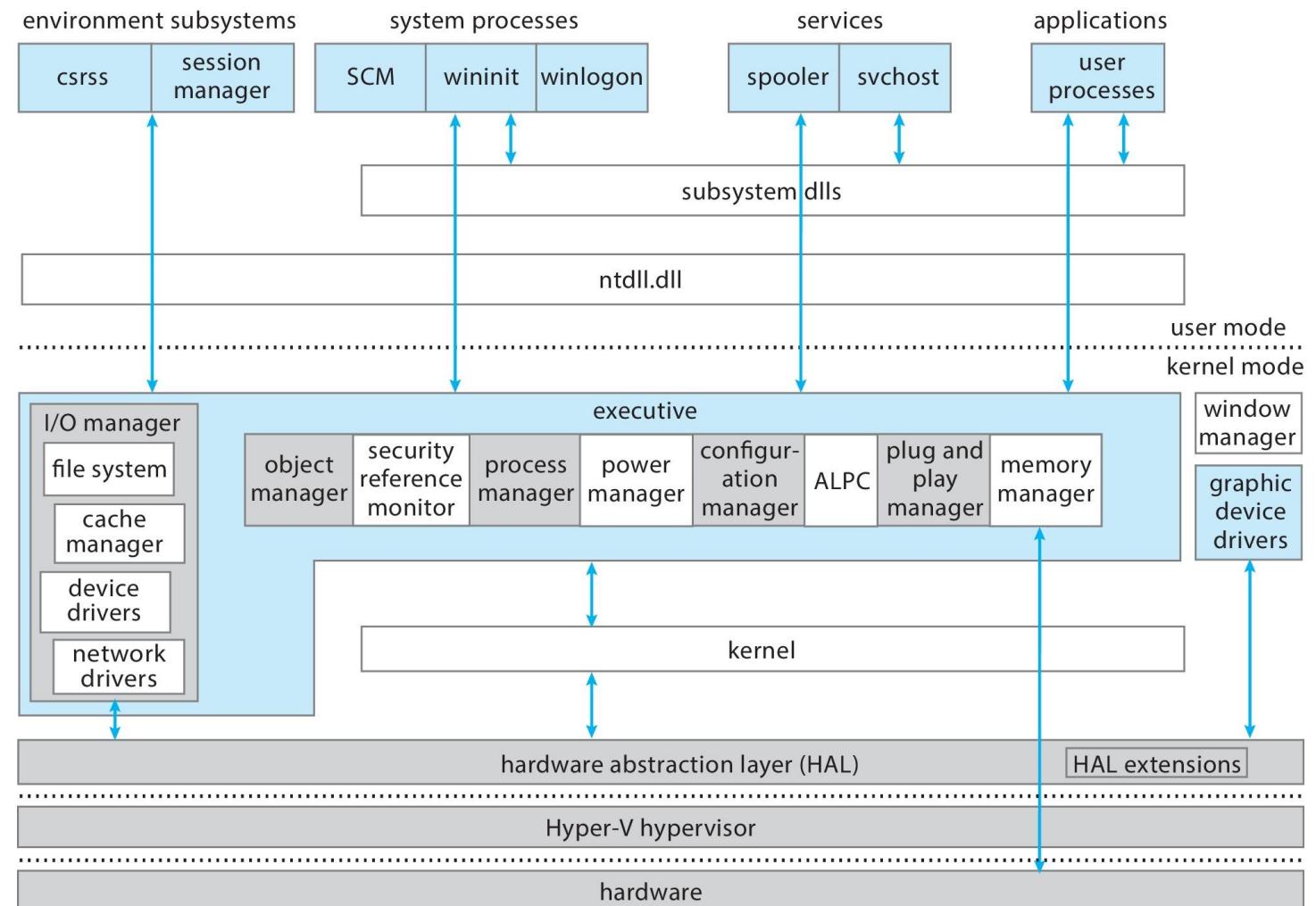


9.2.3 Kiến trúc của Windows 10

- Hệ thống phân lớp gồm nhiều modules hoạt động ở nhiều cấp độ phân quyền khác nhau:
 - Kernel mode: hardware abstraction layer (HAL), nhân, các tác vụ điều hành (executive).
 - User mode: Tập hợp của các hệ thống con (subsystem)
 - Các hệ thống con môi trường (environmental subsystems) mô phỏng các hệ điều hành khác nhau.
 - Các hệ thống con bảo vệ (protection subsystems) cung cấp các chức năng an ninh.



9.2.3 Kiến trúc của Windows 10





9.2.3 Kiến trúc của Windows 10

Các thành phần hệ thống trong Windows 10

- Nhân
- Hyper-V Hypervisor
- HAL
- Nhân bảo mật (secure kernel)
- Các tác vụ điều hành (executive)



9.2.3 Kiến trúc của Windows 10

Các thành phần hệ thống trong Windows 10

- Nhân
 - Chịu trách nhiệm quản lý: Định thời tiểu trình, chuyển ngữ cảnh, đồng bộ tiểu trình, xử lý ngắt và ngoại lệ, chuyển đổi giữa user mode và kernel mode (through qua các system call).
 - Phần lớn được cài đặt bằng C, một số ít sử dụng hợp ngữ.
- Hyper-V Hypervisor: Cung cấp các chức năng ảo hóa
- HAL:
 - Che dấu sự khác biệt về phần cứng.
 - Cung cấp giao diện phần cứng ảo để phục vụ cho bộ định thời trong nhân, các tác vụ điều hành và trình điều khiển thiết bị.



9.2.3 Kiến trúc của Windows 10

Các thành phần hệ thống trong Windows 10

- Nhân bảo mật (secure kernel):
 - Hoạt động như môi trường kernel mode cho các ứng dụng Trustlet (các ứng dụng cài đặt theo mô hình bảo mật Windows).
 - Hỗ trợ truy xuất đến các khóa bí mật phần cứng, trusted platform module (TPM), ...
- Các tác vụ điều hành:
 - Tập hợp các dịch vụ mà các hệ thống con môi trường sẽ sử dụng: Quản lý bộ nhớ ảo, quản lý tiến trình, quản lý I/O, quản lý năng lượng, cơ chế plug-and-play, registry, local procedure call facility, ...
 - Được tổ chức theo nguyên tắc thiết kế hướng đối tượng.



Hệ điều hành Windows

9.2.4 Quản lý tiến trình/tiểu trình trên Windows



2



9.2.4 Quản lý tiến trình/tiểu trình trên Windows

- Mỗi tiến trình có một không gian địa chỉ ảo riêng, các thông tin về hoạt động của tiến trình và một affinity với một hoặc nhiều bộ xử lý.
- Tiểu trình là các đơn vị thực thi được định thời bởi nhân.
- Mỗi tiểu trình có trạng thái riêng, bao gồm độ ưu tiên, processor affinity và các thông tin dành cho việc kế toán.
- Có 8 trạng thái tiểu trình: *initializing*, *ready*, *deferred-ready*, *standby*, *running*, *waiting*, *transition* và *terminated*.
- Mỗi tiểu trình có hai chế độ thực thi: user-mode thread (UT) và kernel-mode thread (KT).
 - Mỗi chế độ sử dụng một stack riêng.
 - Việc chuyển đổi giữa các stack và thay đổi chế độ CPU được thực hiện bởi nhân.



9.2.4 Quản lý tiến trình/tiểu trình trên Windows

Định thời trên Windows 10

- Bộ định thời sử dụng 32 độ ưu tiên để xác định thứ tự thực thi của các tiểu trình.
 - Độ ưu tiên được chia thành 2 lớp:
 - Lớp real-time có độ ưu tiên từ 16 đến 31.
 - Lớp variable có độ ưu tiên từ 0 đến 15.
- Định thời theo độ ưu tiên trên Windows 10:
 - Có xu hướng đem lại thời gian đáp ứng nhanh cho các tiểu trình interactive đang sử dụng chuột và các cửa sổ.
 - Cho phép các tiến trình hướng I/O được giữ các thiết bị I/O đang bận.



9.2.4 Quản lý tiến trình/tiểu trình trên Windows

Định thời trên Windows 10

- Định thời có thể diễn ra khi tiểu trình chuyển sang trạng thái ready hoặc waiting, khi tiểu trình kết thúc hoặc khi ứng dụng thay đổi độ ưu tiên hoặc processor affinity.
- Các tiểu trình real-time được ưu tiên sử dụng CPU, nhưng Windows 10 không đảm bảo chắc chắn một tiểu trình real-time sẽ luôn được thực thi trong bất cứ một khoảng thời gian giới hạn cho trước (soft-realtime).



Hệ điều hành Windows



9.2.5 Quản lý bộ nhớ ảo trên Windows 10

2

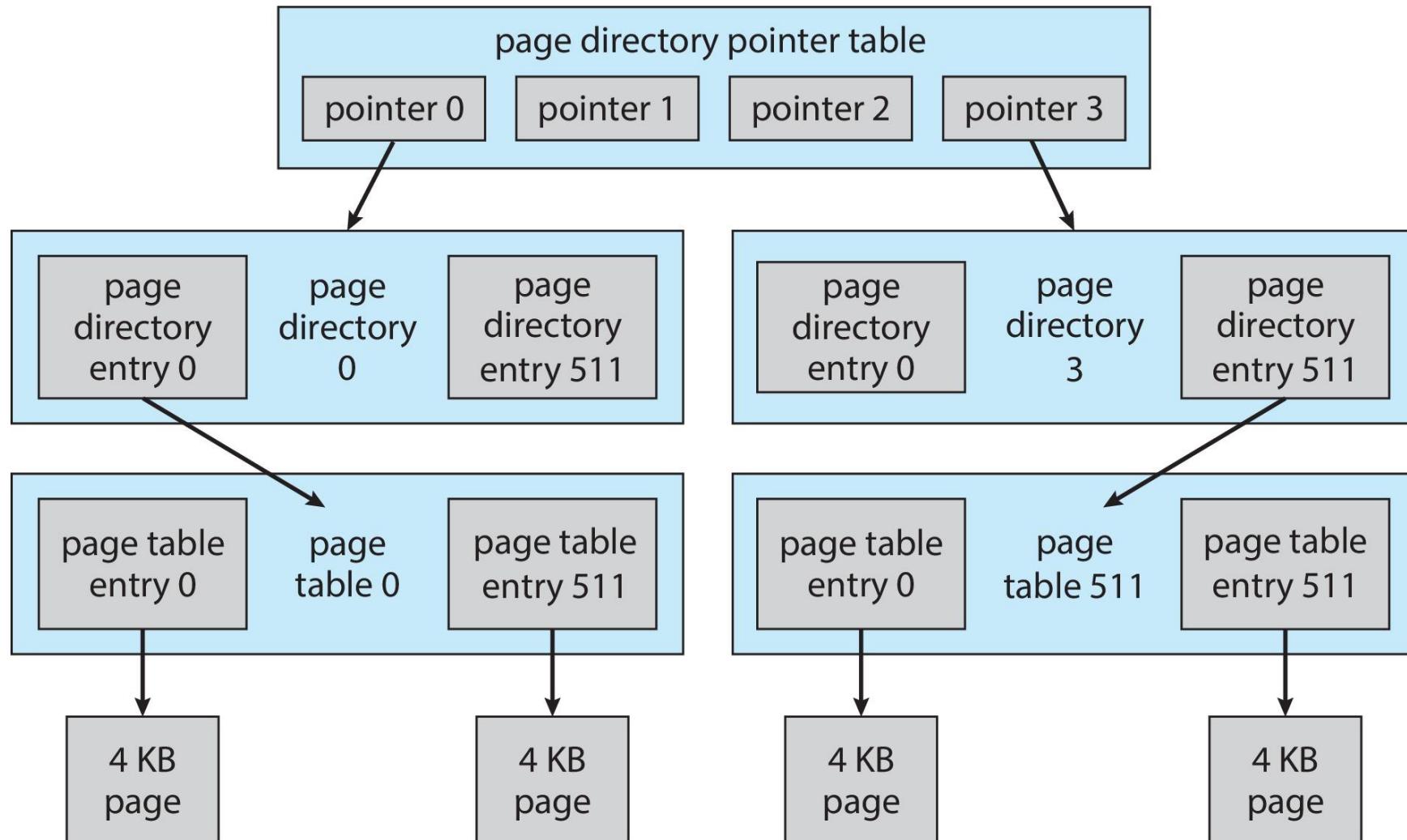


9.2.5 Quản lý bộ nhớ ảo trên Windows 10

- Thành phần quản lý bộ nhớ ảo trên Windows 10 có cơ chế quản lý trang với kích thước trang bất kỳ, miễn là phần cứng có hỗ trợ kích thước đó.
- Thành phần quản lý bộ nhớ ảo trên Windows 10 sử dụng quy trình 2 bước để cấp phát vùng nhớ:
 - Bước 1: Dự trữ (để riêng) một phần không gian địa chỉ của tiến trình.
 - Bước 2: Thực hiện cấp phát vùng nhớ bằng cách gán không gian địa chỉ ảo (trong bộ nhớ vật lý hoặc trong các tập tin paging).
- Mỗi trang có thể có một trong sáu trạng thái sau: valid, zeroed, free, standby, modified và bad.



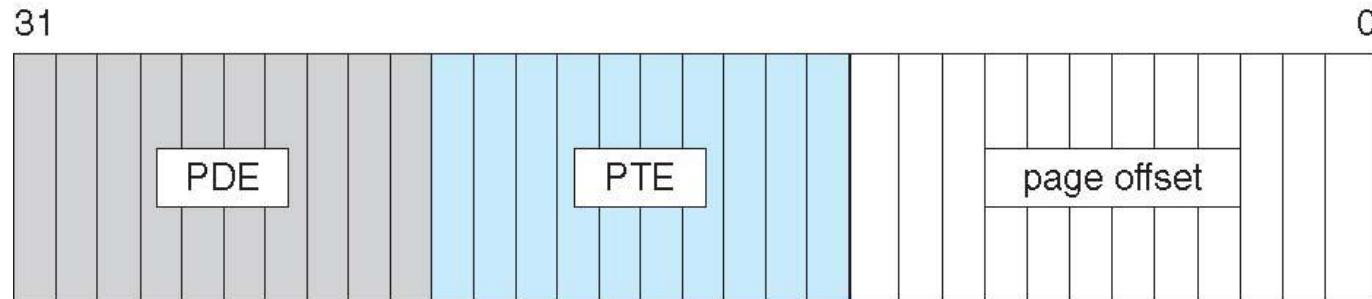
9.2.5 Quản lý bộ nhớ ảo trên Windows 10





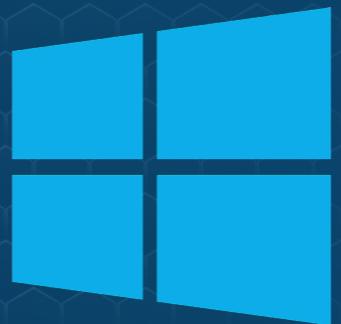
9.2.5 Quản lý bộ nhớ ảo trên Windows 10

- Việc chuyển đổi địa chỉ ảo trên Windows 10 sử dụng nhiều cấu trúc dữ liệu:
 - Mỗi tiến trình có một page directory chứa 1024 mục (*page directory entries*) với kích thước mỗi mục là 4 byte.
 - Mỗi mục trong page directory trỏ đến một bảng trang, mỗi bảng trang này chứa 1024 mục (*page table entries* - PTEs) với kích thước mỗi mục là 4 byte.
 - Mỗi PTEs trỏ đến một khung trang kích thước 4KB trong bộ nhớ vật lý.
- Chuyển đổi địa chỉ ảo thành địa chỉ vật lý





Hệ điều hành Windows



9.2.6 Các tác vụ điều hành

2



9.2.6 Các tác vụ điều hành

Quản lý Plug-and-Play

- Công cụ quản lý Plug-and-Play (PnP) được sử dụng để ghi nhận và đáp ứng khi có các thay đổi về phần cứng.
- Khi các thiết bị mới được thêm vào hệ thống (PCI, USB, ...) công cụ quản lý PnP nạp các trình điều khiển tương ứng với thiết bị đó.
- Công cụ quản lý này cũng sẽ lưu vết quá trình các tài nguyên được sử dụng bởi từng thiết bị.



9.2.6 Các tác vụ điều hành

Local Procedure Call Facility

- Phương thức giao tiếp giữa hai tiến trình trong cùng một máy tính.
- Xây dựng theo mô hình client – server.
- Khi một kênh LPC được tạo ra, một trong số 3 loại kỹ thuật chuyển thông điệp phải được khai báo:
 - Loại 1: Dành cho các thông điệp ngắn, tối đa 256 byte. Port sử dụng cho hàng đợi thông điệp được sử dụng như bộ nhớ lưu trữ trung gian và các thông điệp được sao chép từ tiến trình này đến tiến trình khác.
 - Loại 2: Sử dụng một vùng nhớ chia sẻ, không cần sao chép các thông điệp lớn mà chỉ cần trỏ đến vùng nhớ đó.
 - Loại 3: Gọi là *quick* LPC, được sử dụng bởi thành phần giao diện đồ họa của hệ thống con Win32.



9.2.6 Các tác vụ điều hành

Registry

- Các thông tin cấu hình (configuration) được lưu trong các khu vực riêng gọi là hives. Chúng được quản lý bởi công cụ gọi là registry.
- Các thông tin hệ thống, thiết lập của người dùng, thông tin của phần mềm, các tùy chọn bảo mật và khởi động được lưu giữ trong các hive riêng biệt.
- Hệ điều hành Windows tạo ra các điểm phục hồi hệ thống (system restore point) trước khi thực hiện các thay đổi trên registry, tạo điều kiện phục hồi lại phiên bản registry trước đó nếu có sự cố xảy ra.



Tóm tắt lại nội dung buổi học

- Hệ điều hành Linux và hệ điều hành Windows
 - Lịch sử phát triển
 - Nguyên tắc thiết kế
 - Các thành phần chính
 - Quản lý tiến trình
 - Định thời
 - Giao tiếp liên tiến trình
 - Quản lý bộ nhớ



THẢO LUẬN

