

VIỆN ĐẠI HỌC MỞ HÀ NỘI

Khoa Công Nghệ Thông Tin

Linux Kernel

Giảng viên: Ths. Nguyễn Thuỳ Linh

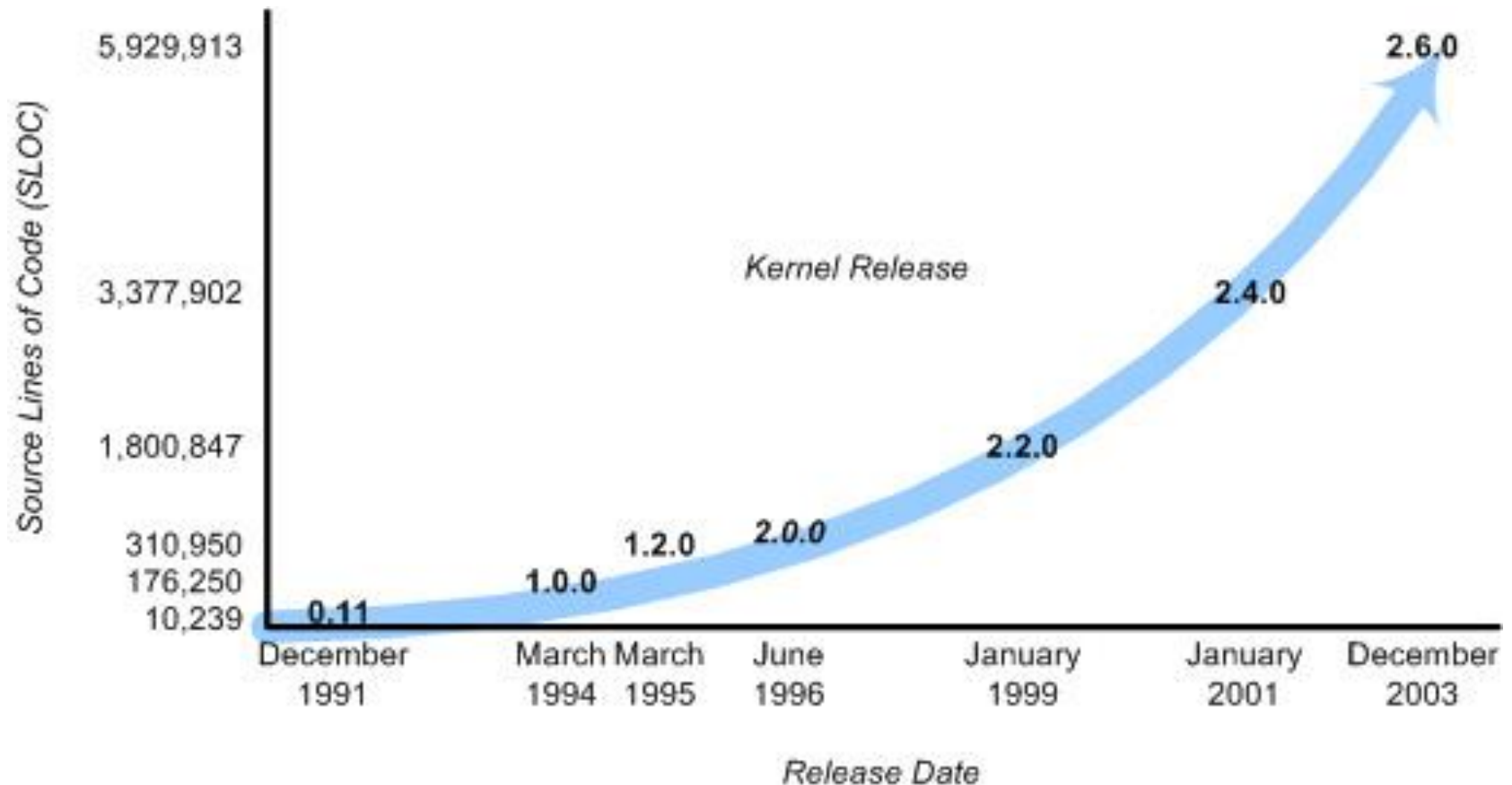
Email: linhnt1601@gmail.com

Linux hay GNU/Linux

- Linux là nhân của hệ điều hành, tương tác trực tiếp với phần cứng và cung cấp các dịch vụ cho chương trình người sử dụng (User Program)
 - Nhiều các ứng dụng khác bổ xung kết hợp với nhân Linux làm thành một hệ điều hành sử dụng được, các ứng dụng đó phần lớn là phần mềm GNU
 - GNU là dự án của tổ chức phần mềm tự do FSF, xây dựng các phần mềm giống Unix, nhưng là phần mềm tự do và không chứa mã Unix
- => Hệ điều hành được gọi là GNU/Linux

Lịch sử phiên bản nhân Linux

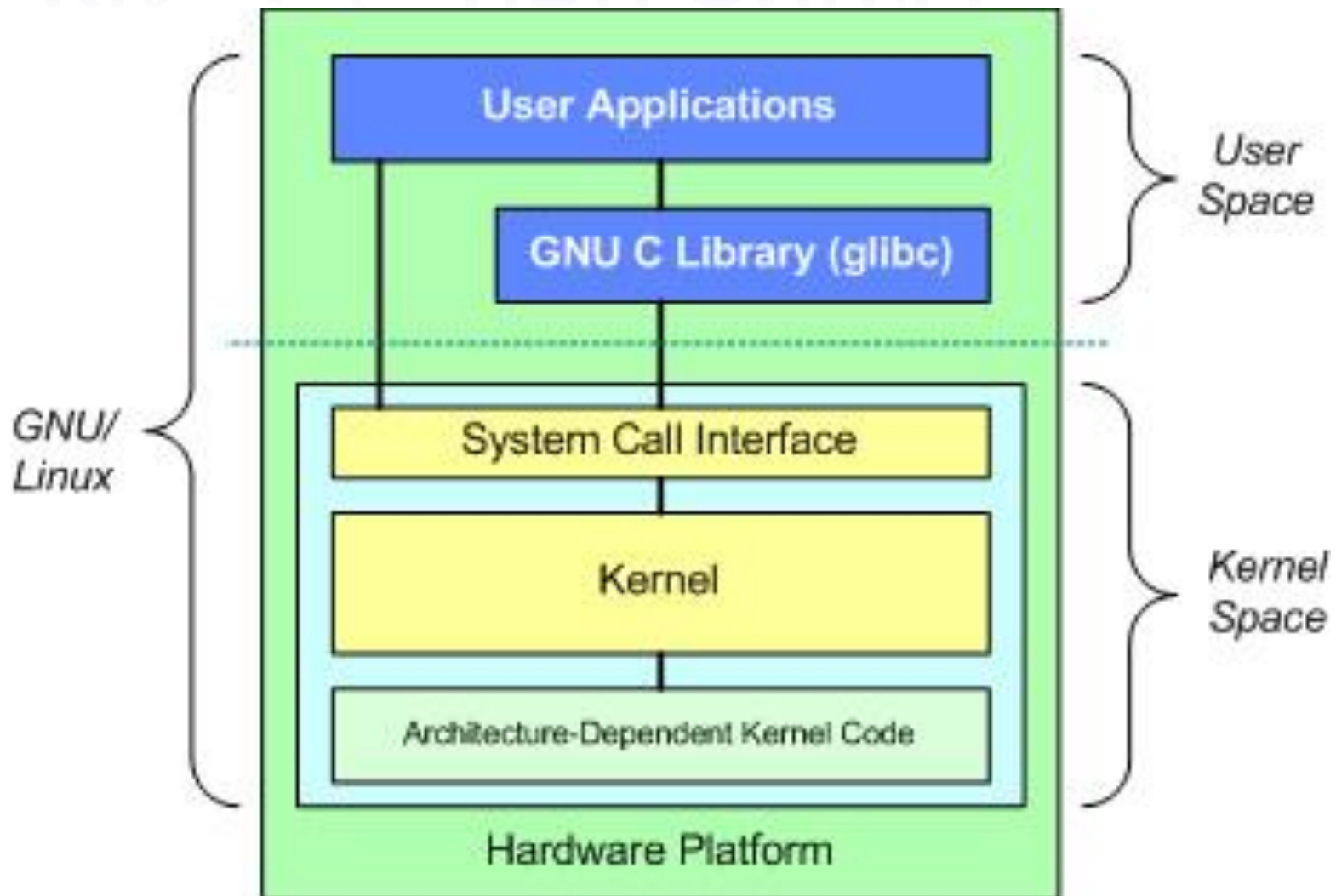
- Năm 1991, Linus Torvalds đưa ra phiên bản đầu tiên 0.11, sử dụng giấy phép GNU GPL



Cách đặt tên phiên bản nhân

- Đánh số dạng : **X.Y.Z**
- Trong đó X là số hiệu phiên bản
- Y là số hiệu phụ của phiên bản, cũng mang ý nghĩa chỉ tình trạng của phiên bản, trong đó, Y chẵn thể hiện phiên bản đã ổn định, Y lẻ là bản chưa ổn định, đang trong quá trình phát triển
- Z là chỉ định cho số hiệu phát hành của một phiên bản nhân Linux. Một phiên bản ổn định của nhân Linux có thể có nhiều số hiệu phát hành khác nhau
- VD : 2.4.28 2.6.28

Kiến trúc HĐH GNU/Linux



Kiến trúc HĐH GNU/Linux

- **User Applications** : Lớp các ứng dụng người dùng
- **GNU C Library (glibc)** : Lớp thư viện phục vụ cho giao diện các lời gọi hệ thống tạo liên kết giữa các ứng dụng và nhân Linux.
- **System call interface** : giao diện các lời gọi hệ thống thực hiện các chức năng cơ bản như đọc, ghi
- **Kernel** : phần mã nhân hệ điều hành hoặc chính xác hơn là mã nhân độc lập với kiến trúc vi xử lý (processor). Các mã lệnh trong lớp này dùng chung cho mọi loại processor mà Linux hỗ trợ
- **Architecture-dependent code** : các mã lệnh phụ thuộc vào kiến trúc từng loại processor

Nhân là gì ?

- Nhân Linux thực ra là bộ quản lý các tài nguyên. Các tài nguyên gồm:
 - *các tiến trình (process)*
 - *bộ nhớ (memory)*
 - *Các thiết bị phần cứng (hardware devices).*
- Nhân Linux quản lý các tài nguyên đó và là người điều hành việc truy cập tài nguyên đồng thời của nhiều user. (*User được hiểu theo nghĩa rộng bao gồm tất cả những gì có nhu cầu sử dụng tài nguyên hệ thống: các tiến trình, các phần mềm, v.v....*)

Nhân Linux

- LinuxKernel thực thi một giao diện máy ảo đối với các tiến trình người sử dụng.
- Linux Kernel tổng hợp tất cả các phần cứng vào một giao diện ảo phù hợp.
- Linux hỗ trợ đa nhiệm theo một kiểu mà người sử dụng dễ hiểu: Từng tiến trình một có thể thực hiện, mặc dù nó là tiến trình duy nhất trên máy, loại trừ việc sử dụng bộ nhớ chính và các tài nguyên phần cứng khác. Kernel thực tế chạy nhiều tiến trình đồng thời và chịu trách nhiệm việc dàn xếp truy cập tới các tài nguyên phần cứng để cho từng tiến trình truy cập thuận lợi trong khi việc bảo mật trong các tiến trình vẫn được duy trì.

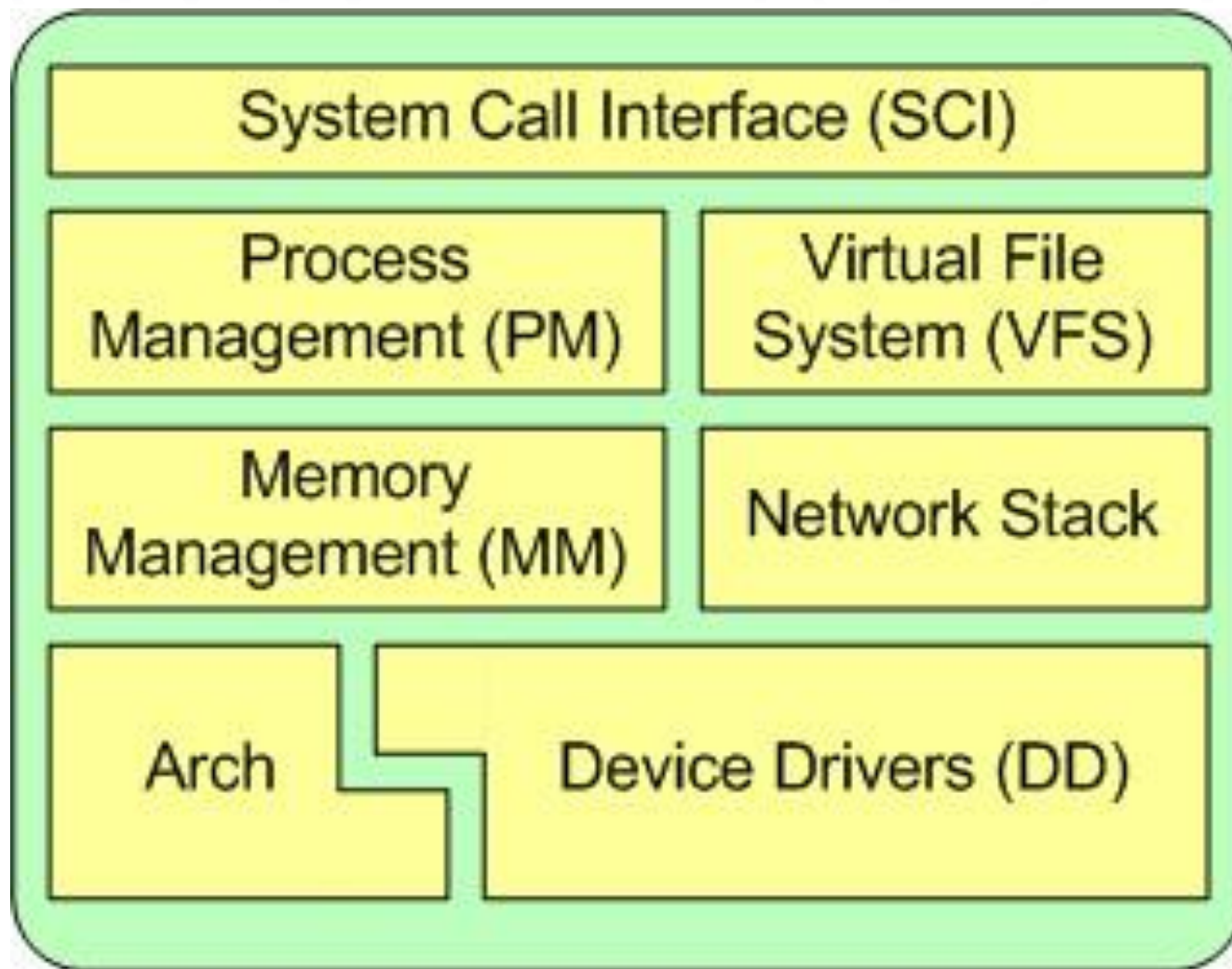
Chức năng của nhân Linux

- *Quản lý thời gian*: Gợi thời gian hệ thống, xác định thời gian CPU, ngắt tiến trình nếu thời gian thực hiện tiến trình đó là quá lâu.
- *Quản lý tài nguyên*: Như là bộ nhớ đĩa cứng...
- *Quản lý hệ thống file*: Có thể là các file, thư mục cục bộ hay từ xa...
- *Quản lý các tiến trình thường trú*.
- *Quản lý bộ nhớ ảo*: Để thực thi nhiều tiến trình đồng thời trong khi số lượng bộ nhớ có hạn, UNIX tổ chức bộ nhớ trên đĩa như là một vùng bộ nhớ (bộ nhớ ảo). Kernel phải "swap" các tiến trình giữa bộ nhớ và bộ nhớ ảo.

Chức năng của nhân Linux

- *Quản lý tiến trình:* UNIX là một hệ điều hành đa nhiệm, do đó việc quản lý tiến trình đồng thời rất phức tạp. Hệ điều hành phải quản lý việc khởi tạo và kết thúc các tiến trình cũng như các tranh chấp có thể xảy ra.
- *Quản lý các bộ điều khiển thiết bị.*
- *Quản lý mạng:* Bao gồm nhiều thiết bị phần cứng khác nhau và các thủ tục khác.
- *Quản lý việc khởi động và dừng máy.*

Các hệ thống con chính của nhân



Giao diện lời gọi hệ thống

- Giao diện lời gọi hệ thống (System Call Interface – SCI) thực hiện các lời gọi hệ thống từ vùng ứng dụng vào nhân Linux.
- Giao diện này độc lập với kiến trúc bộ vi xử lý ngay cả trong cùng một họ vi xử lý.
- SCI có thể thực hiện các dịch vụ gọi hàm đồn kênh và tách kênh.
- Các gói liên quan được cài trong thư mục ẩn `./linux/kernel` và phần độc lập với kiến trúc vi xử lý nằm trong `./linux/arch`.

Quản lý các tiến trình

- Quản lý tiến trình (Process Management) đảm bảo việc thực hiện các tiến trình. Trong vùng nhân Linux, mỗi tiến trình được gọi là một mạch lệnh (thread) và được thể hiện thành một vi xử lý ảo (gồm mã lệnh, dữ liệu, các ngăn xếp và các thanh ghi của CPU). Trong vùng ứng dụng thì chỉ dùng từ tiến trình mặc dù Linux không phân biệt hai khái niệm này (threads và processes). Nhân cung cấp một giao diện lập trình ứng dụng (API) để: tạo tiến trình mới (fork, exec hoặc các hàm POSIX), ngừng tiến trình (kill, exit) và thông tin, đồng bộ giữa các tiến trình (signal hoặc các cơ cấu POSIX).

Quản lý các tiến trình

- Quản lý tiến trình còn dùng để chia sẻ CPU giữa các mạch lệnh đang hoạt động. Nhân thực hiện một thuật toán lập lịch cố định bất kể đến các mạch lệnh đang tranh chấp quyền sử dụng CPU. Lịch này cũng hỗ trợ cả chế độ đa xử lý đối xứng (Symmetric MultiProcessing – SMP). Các gói liên quan được cài trong thư mục ẩn `./linux/kernel` và phần độc lập với kiến trúc vi xử lý nằm trong `./linux/arch`.

Quản lý tiến trình

- Module luật định thời (scheduling policy): chịu trách nhiệm phân xử xem process nào được quyền truy xuất CPU. Hệ thống hoạt động có thông suốt hay không nhờ vào bộ luật này, tránh trường hợp 1 process lợi dụng sơ hở của điều luật mà chiếm thời gian hệ thống qua nhiều lần các process khác bị đóng băng
- Module phụ thuộc kiến trúc (architecture-specific): Được thiết kế với một giao diện chung để tóm tắt chi tiết của bất kỳ kiến trúc máy tính cụ thể nào. Những module này gồm các mã assembly phụ thuộc vào mỗi loại CPU dùng để dừng hay tiến hành một tiến trình.

Quản lý tiến trình

- Module độc lập kiến trúc (architecture-independent): Module này giao tiếp với module policy để quyết định tiến trình nào sẽ thực hiện tiếp. Sau đó module architecture-specific để khôi phục tiến trình thích hợp. Thêm vào đó module này gọi memory manager để đảm bảo rằng bộ nhớ được lưu trữ chính xác cho tiến trình được khôi phục.
- Module hàm gọi hệ thống(system call): cho phép tiến trình người sử dụng truy nhập chỉ tới tài nguyên được đưa ra bởi kernel. Nó gồm các hàm mà user có thể dùng để tương tác với bộ quản lý tiến trình.

Quản lý bộ nhớ - MM

- Để sử dụng bộ nhớ hiệu quả theo cách mà phần cứng quản lý bộ nhớ ảo, bộ nhớ được chia thành các trang (mỗi trang là 4KB đối với phần lớn loại vi xử lý). Linux có các cơ cấu quản lý lượng bộ nhớ khả dụng và các cơ cấu phần cứng để mapping giữa bộ nhớ vật lý và bộ nhớ ảo.
- Việc quản lý bộ nhớ còn làm nhiều hơn là chỉ quản lý các trang 4KB. Linux dùng một sơ đồ định vị lát (slab allocator) lên trên mỗi trang. Sơ đồ này dùng trang 4KB làm cơ sở nhưng tạo một cấu trúc bên trong, theo dõi trang nào đầy, trang nào mới dùng một phần, trang nào còn trống.

Quản lý bộ nhớ - MM

- Khi nhiều user sử dụng bộ nhớ, dung lượng có thể không đủ. Khi đó các trang nhớ được chuyển sang ổ cứng. Quá trình này được gọi là trao đổi (swapping) giữa bộ nhớ và ổ cứng. Các gói phần mềm liên quan đến quản lý bộ nhớ đặt trong thư mục `./linux/mm`.
- Module phụ thuộc kiến trúc: code gọi các lệnh của hệ thống quản lý bộ nhớ của CPU

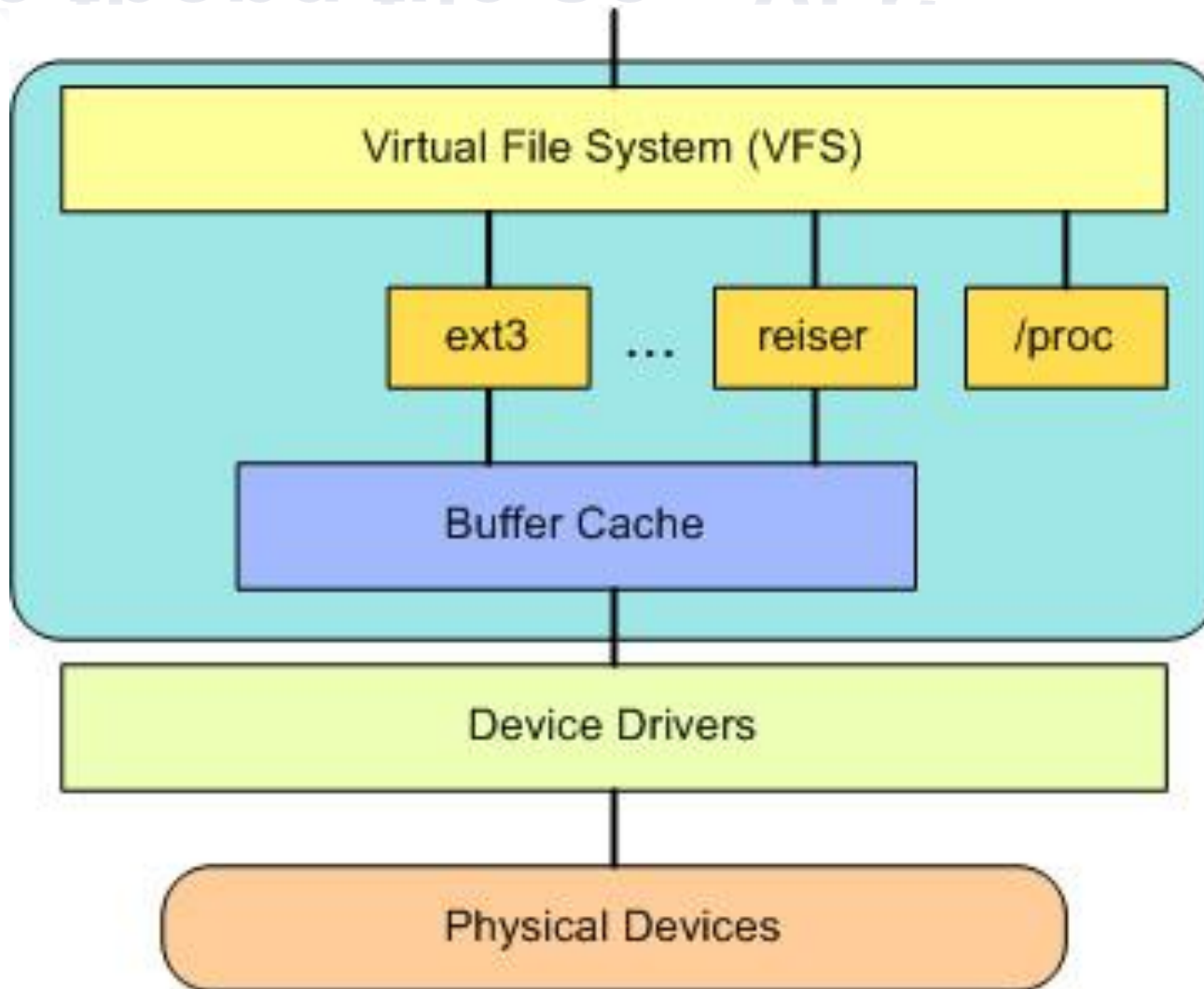
Quản lý bộ nhớ - MM

- Module độc lập kiến trúc: ánh xạ cho từng process và swap bộ nhớ ảo. Nó cũng quyết định xem phải loại bỏ trang nào , mở trang nào. Các lập trình viên Linux không thiết kế 1 module policy riêng vì policy cho MM sẽ không thay đổi
- System call cho phép các tiến trình tác động lên MM bao gồm xin cấp phát vùng nhớ

Hệ thống file ảo

- Hệ thống file ảo (Virtual File System - VFS) là một khía cạnh của nhân Linux, cung cấp một giao diện trừu tượng hoá chung cho hệ thống file.
- VFS tạo nên một lớp chuyển đổi giữa System Call Interface và các hệ thống file của Linux.
- VFS hỗ trợ các loại định dạng file khác nhau để hệ thống có thể làm việc được
- VFS giúp cho các thiết bị vật lý có thể làm việc với hệ thống đơn giản hơn

Hệ thống file ảo - VFS



Hệ thống file ảo

- Nằm trên cùng của VFS là lớp các API các chức năng như mở, đóng, đọc, viết file. Dưới cùng của VFS là lớp trừu tượng hệ thống file xác định các chức năng lớp trên thực hiện như thế nào. Đó là các plug-in đối với một hệ thống file cho trước (có trên 50 plug-in như vậy). Các phần mềm liên quan đến VFS nằm trong thư mục `./linux/fs`.

Hệ thống file ảo

- Bên dưới lớp file hệ thống là bộ đệm cache (buffer cache) gồm các chức năng chung cho mọi hệ thống file (không phụ thuộc vào một kiểu hệ thống file riêng biệt nào). Lớp cache này tối ưu hoá việc truy cập vào các thiết bị vật lý bằng cách giữ dữ liệu trong một thời gian ngắn (hoặc đọc trước sao cho dữ liệu luôn có khi cần). Dưới bộ đệm cache là các driver thiết bị là giao diện của các thiết bị vật lý cụ thể.

Bộ quản lý mạng

- Bộ quản lý mạng được thiết kế theo một kiến trúc lớp mô phỏng theo đúng kiến trúc lớp của các giao thức. Nhắc lại rằng IP là giao thức lớp mạng lõi nằm bên dưới giao thức vận chuyển (thường là TCP). Bên trên TCP là lớp socket được gọi đến qua SCI.
- Lớp socket là API chuẩn của hệ thống con network, tạo nên một giao diện cho các giao thức mạng khác nhau. Lớp socket quy định một cách quản lý kết nối và di chuyển dữ liệu chuẩn hoá giữa các điểm đầu cuối. Tài nguyên mạng nằm ở thư mục `./linux/net`.

Bộ quản lý mạng

- Module Driver cho thiết bị mạng: mỗi module cho mỗi loại thiết bị
- Module độc lập thiết bị: phân chia tất cả các thiết bị mạng
- Module giao thức mạng: mỗi module cho mỗi loại giao thức truyền
- Module độc lập giao thức mạng: phân chia cho tất cả các module giao thức mạng và các driver.
- System call : Hạn chế đưa ra những thủ tục mà tiến trình người sử dụng có thể truy nhập.

Các driver thiết bị

- Phần lớn mã nguồn của nhân Linux là các driver để điều khiển các thiết bị phần cứng.
- Cây thư mục của mã nguồn nhân Linux có một thư mục con driver trong đó có các thư mục con ứng với các thiết bị khác nhau.
- Mã nguồn driver nằm ở `./linux/drivers`.

Mã lệnh phụ thuộc kiến trúc vi xử lý

- Mã lệnh phụ thuộc kiến trúc vi xử lý (Architecture-dependent code) chứa các phần mã lệnh tùy theo các loại vi xử lý khác nhau (x86, x64, amd...)
- Phần lớn Linux độc lập với kiến trúc vi xử lý, nhưng cũng có những bộ phận cần phải theo đúng từng kiến trúc cụ thể để hoạt động được và hiệu quả.
- Thư mục con `./linux/arch` chứa các mã nguồn phụ thuộc kiến trúc đó. Ví dụ với một máy trạm tiêu biểu, thư mục đó là `i386`,

Một số đặc điểm đáng chú ý

- Nhân Linux đã sử dụng bộ nhớ và CPU ngày càng hiệu quả hơn và đặc biệt ổn định.
- Tính khả chuyển (portability) : Linux có thể dịch để chạy trên rất nhiều loại processor và nền tảng phần cứng khác nhau đáp ứng các ràng buộc kiến trúc và nhu cầu khác nhau

Một số đặc điểm đáng chú ý

- Linux, với tư cách là một hệ điều hành nguồn mở, là nơi dùng để test các giao thức mới và những đặc điểm tiên phong của các giao thức đó. Linux hỗ trợ một số lớn giao thức mạng từ TCP/IP cho đến các giao thức mạng cao tốc (trên 1 Gigabit Ethernet – 1GbE – cho đến 10GbE). Linux cũng hỗ trợ các giao thức như SCTP (Stream Control Transmission Protocol) có nhiều đặc tính tiên tiến hơn TCP.

Một số đặc điểm đáng chú ý

- Linux cũng là một kernel động, cho phép thêm bớt các thành phần phần mềm trong khi đang chạy. Các thành phần đó gọi là các module tải động được (dinamically loadable kernel modules) có thể được tải vào bộ nhớ khi boot (khi tìm thấy một phần cứng mới cần đến nó) hoặc tải và rút ra khỏi bộ nhớ bất kỳ lúc nào bởi user.

Một số đặc điểm đáng chú ý

- Một đặc tính mới của Linux gần đây là nó có thể dùng làm hệ điều hành cho các hệ điều hành khác (hypervisor). Kernel Linux gần đây đã được bổ xung tính năng máy ảo từ nhân (Kernel-based Virtual Machine – KVM). Tính năng này tạo ra một giao diện mới cho vùng người dùng, cho phép các hệ điều hành khác (Linux hoặc Windows) chạy trên KVM. Yêu cầu duy nhất là processor phải hỗ trợ tập lệnh ảo hoá mới.