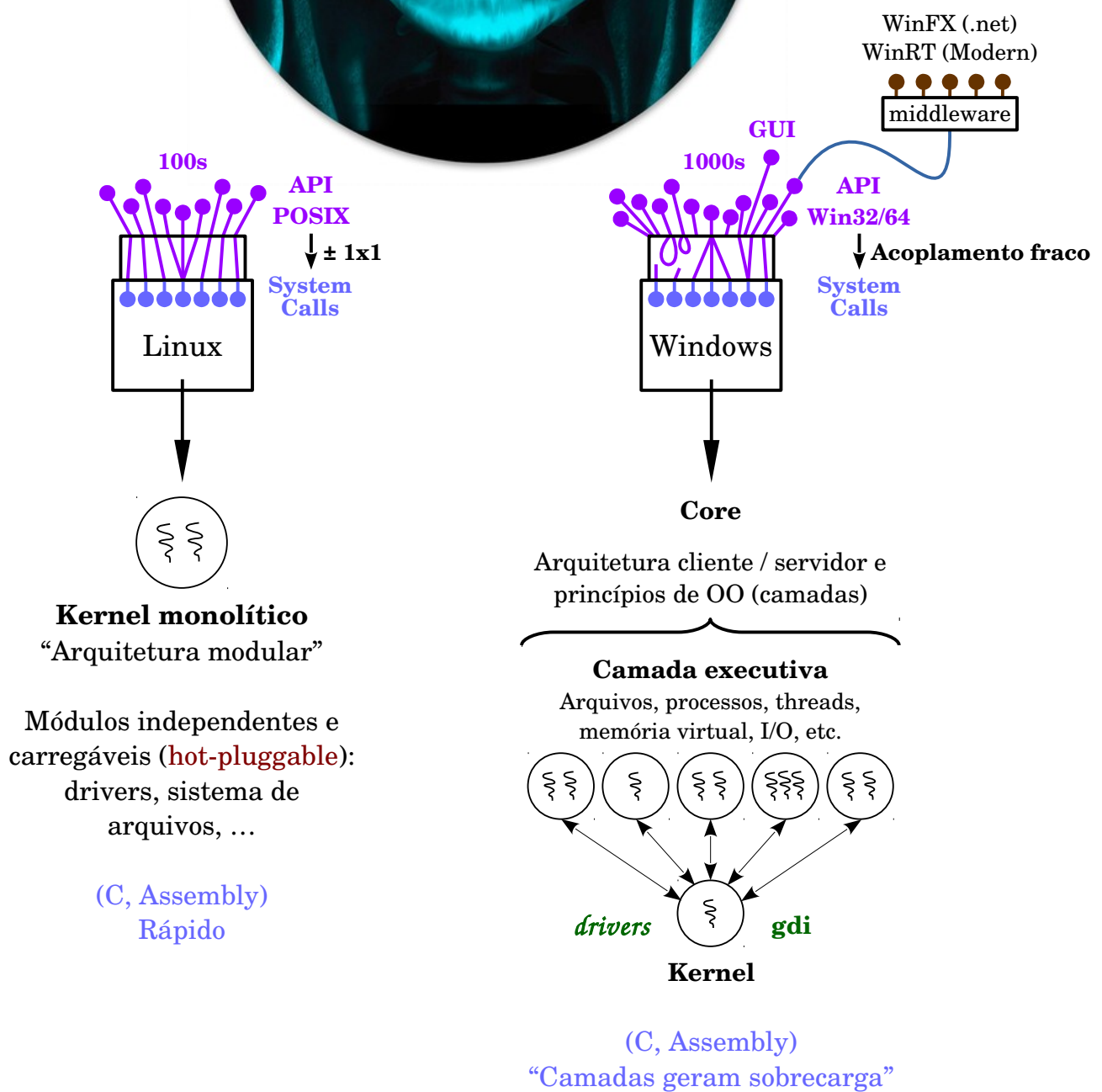


Linux / Windows – Raio X



POSIX

Portable Operating System Interface (uniXes)

1. UNIX → BSDs (FreeBSD, NetBSD, OpenBSD, Dragonfly BSD), Solaris, macOS, iOS
2. Linux → Android

Kernel

- Monolítico
- Camadas (OOD – Object Oriented Design)
- Modular
- Microkernel
- Exokernel

Curiosidade

Why Windows 10?

1. Because 7, 8, 9, ...
2. Windows 9 → Conflito de identificação: Windows 9x (95, 98, 9)

APIs

POSIX

pid = **fork()**

pid = **waitpid**(*pid*, *status*, *options*)
s = **execve**(*filename*, *argv*, *envp*)
 → **POSIX**: **execl**, **execv**, **execle**, **execve**
 → **syscall**: **exec**

exit(*status*)

fd = **open**(*pathname*, *flags*, ...)

s = **close**(*fd*)

n = **read**(*fd*, *buf*, *count*)

n = **write**(*fd*, *buf*, *count*)

position = **lseek**(*fd*, *offset*, *whence*)

s = **stat**(*pathname*, *buf*)

s = **mkdir**(*pathname*, *mode*)

s = **rmdir**(*pathname*)

s = **link**(*oldpath*, *newpath*)

s = **unlink**(*pathname*)

s = **mount**(*source*, *target*, ...)

s = **umount**(*target*)

s = **chdir**(*path*)

s = **chmod**(*pathname*, *mode*)

s = **kill**(*pid*, *sig*)

seconds = **time**(*tloc*)

Win32/64

CreateProcess ^{*10}

LPCSTR	lpApplicationName
LPSTR	lpCommandLine
LPSECURITY_ATTRIBUTES	lpProcessAttributes
LPSECURITY_ATTRIBUTES	lpThreadAttributes
BOOL	bInheritHandles
DWORD	dwCreationFlags
LPVOID	lpEnvironment
LPCSTR	lpCurrentDirectory
LPSTARTUPINFOA	lpStartupInfo
LPPROCESS_INFORMATION	lpProcessInformation

WaitForSingleObject ^{*2}

ExitProcess ^{*1}

CreateFile ^{*7}

CloseHandle ^{*1}

ReadFile ^{*5}

WriteFile ^{*5}

SetFilePointer ^{*4}

GetFileAttributesEx ^{*3}

CreateDirectory ^{*2}

RemoveDirectory ^{*1}

DeleteFile ^{*1}

SetCurrentDirectory ^{*1}

SetFileSecurity ^{*3}

TerminateProcess ^{*2}

GetLocalTime ^{*1}

Características

Linux

Projetado por programadores, para programadores.

Todo programa do sistema deve fazer uma única coisa e fazer bem.

Flexibilidade: elementos básicos que podem ser combinados das mais variadas formas.

cp, rm

X Window System (Servidor X ou X11) executa em modo usuário. O SO pode entrar em ação sem uma GUI.

case-preserving; case-sensitive

Chamadas do sistema simples, com poucos parâmetros e pouca redundância.

Windows

Interface gráfica é fácil para iniciantes e oculta do usuário os detalhes de funcionamento do sistema, mas oferece pouca flexibilidade.

copy, delete

Gerenciamento da interface gráfica está incorporada ao core.

case-preserving; case-insensitive

Chamadas do sistema complexas, com muitos parâmetros e funcionalidades redundantes.

Linux

Processos obedecem a uma hierarquia pai-filhos

Primitivas essenciais apenas.

Arquivos de configuração independentes e acesso restrito.

Uso: PCs, notebooks, tablets, smartphones, servidores, cluster (nuvem), mainframes, super computadores, Internet (roteadores).

Apropriado para máquinas com <16 MB e 100's GB RAM

Processadores físicos: 1000's

Sistema de arquivos virtual integra sistemas de arquivos diferentes em uma única hierarquia.

Windows

Primitivas essenciais e funcionalidades de alto nível, ex.: CopyFile.

Registry (Registro do Windows): normalmente as aplicações tem acesso irrestrito ao registro → **Caos!**

Uso: PCs, notebooks, tablets, smartphones, servidores, cluster (nuvem), Xbox.

Windows 7: 256 CPUs

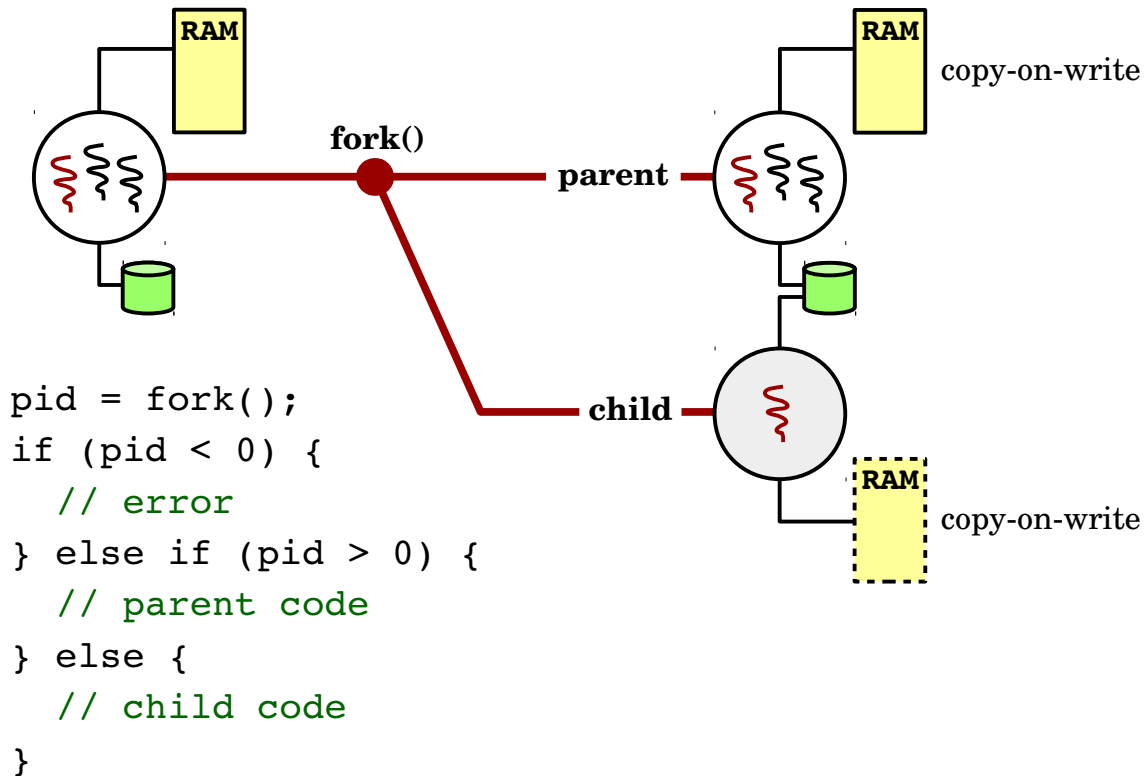
Windows 10: 640 CPUs

Windows Server 2016: 512 cores

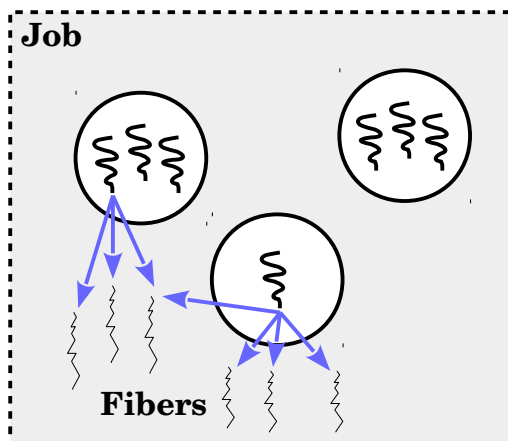
Sistemas de arquivos diferentes (NTFS, FAT-16, FAT-32) são identificados por unidades distintas (C:, D:).

Processos e Threads

Linux



Windows



Jobs e Fibers: Opcional

Job: Controle/restrições sobre um grupo

Fiber = Lightweight thread

Fibers devem ser escalonadas cooperativamente pela aplicação (user-mode): SwitchToFiber()

Uma thread executa uma única fiber por vez

