

Os 5 diferentes modelos de kernels

/* observação: não estou defendendo qual modelo é melhor ou não. Estou simplesmente demonstrando neste artigo as características de cada um dos modelos de kernel. */



(http://3.bp.blogspot.com/-8gIFIPlmpjE/VUK5t1kjXjI/AAAAAAAAY0/Dft9z9GK_I0/s1600/How%2BLinux%2Bis%2BBuilt.png)

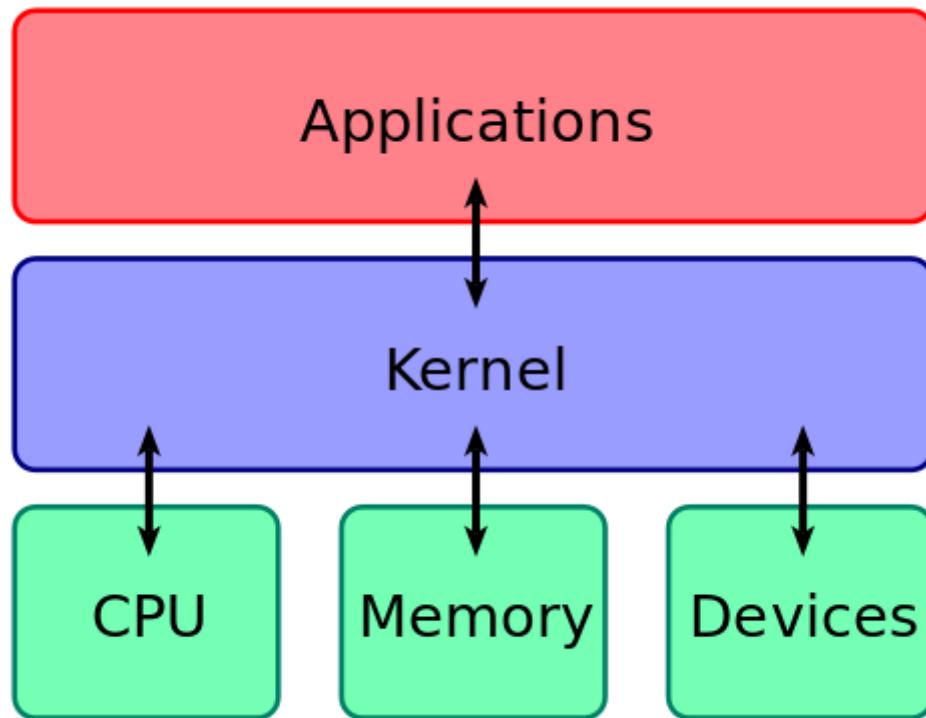
Tratando do assunto sobre o sistema operacional Linux (ou como já mencionei no título do meu canal, não entro em debate caso queira chamá-lo de GNU/Linux ou não; isso é uma coisa individual), precisamos antes analisar um pouco a anatomia de um sistema.

Um sistema operacional é um conjunto de software, cada item construído para um propósito específico. Dividindo o sistema em partes e sem entrar em detalhes, temos o kernel, as bibliotecas do sistema, as ferramentas do sistema e as ferramentas de desenvolvimento. Todas as demais ferramentas que utilizamos no dia a dia são construídas com base nessas.

E como já é notório a todos que Linux é o núcleo (kernel) do sistema, vamos tratar aqui somente desta parte.

O kernel possui quatro responsabilidades básicas:

- Gerenciamento de dispositivos
- Gerenciamento de memória
- Gerenciamento de processos



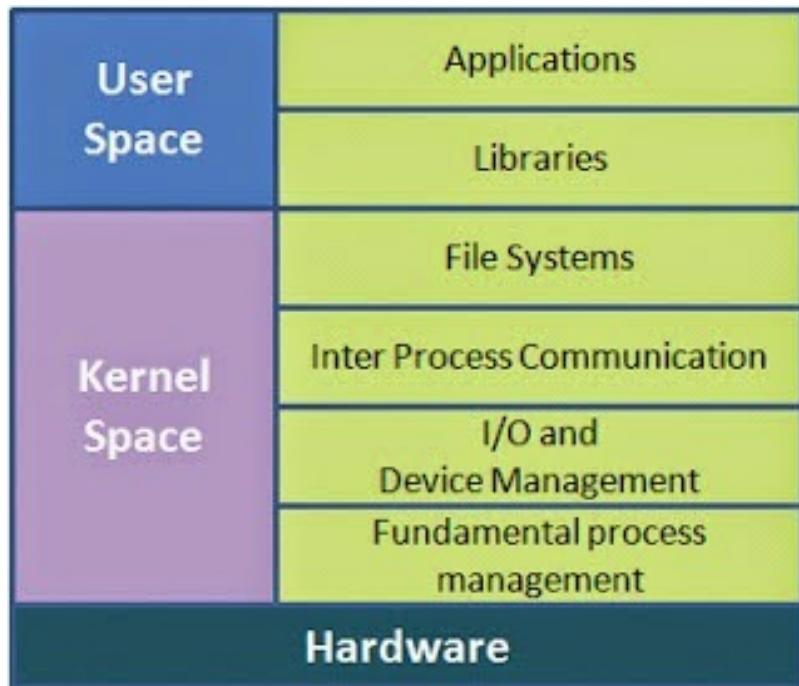
(https://upload.wikimedia.org/wikipedia/commons/thumb/8/8f/Kernel_Layout.svg/500px-Kernel_Layout.svg.png)

É o que o núcleo do sistema operacional é encarregado de realizar; gerenciamento de comunicações entre dispositivos e software, gerencia os recursos do sistema e etc...

Dentro do assunto kernel, temos quatro modelos diferentes entre eles: O monolítico, o microkernel, o nanokernel e o exokernel.

Kernel monolítico:

Também por vezes escutei sendo chamado de monobloco ou estático. É o modelo em que a maioria de seus recursos são executados pelo próprio kernel no kernel space (espaço reservado para carregar o kernel e para que o kernel realize suas funções).



(<http://1.bp.blogspot.com/>-

X1tBSu_gEI8/VQw5tMj4JEI/AAAAAAAOU/w9djIVUpw2E/s1600/main-qimga600677497601b6b1b2fee7224472f16.jpg)

Exemplos de sistemas operacionais kernel monolítico:

- Os UNIX-likes (como os BSDs e seus derivados, os Solaris, AIX, HP-UX, Linux). Não vou entrar em detalhes aqui sobre os Unix. Sugiro a leitura dos meus posts sobre os sistemas:

Definição UNIX-like

(<http://tocadotux.blogspot.com.br/2015/02/definicao-unix-like.html>),

O que é Linux: Uma visão geral do sistema operacional Linux (<http://tocadotux.blogspot.com.br/2014/10/o-que-e-linux-uma-visao-geral-do.html>), POLÊMICA - Linux ou BSD? Qual o melhor? (<http://tocadotux.blogspot.com/2015/02/polemica-linux-ou-bsd-qual-o-melhor.html>)

e

Uma opinião nas diferenças entre BSD e Linux (Dando Crédito aonde o Crédito é Devido)

(<http://tocadotux.blogspot.com.br/2014/12/uma-opiniao-nas-diferencias-entre-bsd-e.html>).

- MS-DOS, Windows 9x (incluindo o

FreeDOS

) (<http://www.freedos.org/>)

```

IPDru 3 encontrado, irq=0x60
C:\> \PROG>rem ipdrv -u
C:\> \PROG>goto Fim
C:\> \PROG>if not ""=="Teste" cd ..
C:\>rem mode co80

C:\...\I>dosfsck -alvUw C:
dosfsck 2.11.DOS3, 8 Aug 2007, FAT32, LFN
Checking we can access the last sector of the filess
Boot sector contents:
System ID "FRDOS5.1"
Media byte 0xf8 (hard disk)
    512 bytes per logical sector
    16384 bytes per cluster
    32 reserved sectors
First FAT starts at byte 16384 (sector 32)
    2 FATs, 32 bit entries
    19529728 bytes per FAT (= 38144 sectors)
Root directory start at cluster 2 (arbitrary size)
Data area starts at byte 39075840 (sector 76320)
    4881813 data clusters (79983624192 bytes)
63 sectors/track, 255 heads
    2048 hidden sectors
156294336 sectors total
Starting check/repair pass.
Reading file allocation table.
-

```

(<http://1.bp.blogspot.com/-gT3wa48KXS8/VQw87PePtPI/AAAAAAAOG/HElWjDIXjc/s1600/freeDOS.png>)
 (FreeDOS)

- Mac OS nas versões abaixo do 8.6 [possuo uma cópia do Mac OS 8.5. Na época, o Macbook preto usava dispositivo drive de disquete removível para conectar o drive de CD (no mesmo local)].

Kernel Monolítico e modular

O Linux possui a característica de ser tanto monolítico e modular. Ambos possuem (digamos) vantagens e desvantagens; o ponto positivo do monolítico é que ele proporciona melhor segurança e melhor desempenho devido seus recursos residirem dentro do próprio kernel (built-in); porém, por esse fato, seus recursos (mesmo que não sejam necessários em algumas circunstâncias) estarão sempre em execução do momento em que ligar o computador até o momento que desligá-lo, consumindo assim recurso do hardware. Isso em um PC hoje em dia, onde pode-se possuir processadores com 8 núcleos, em torno de 8 Gigabytes de RAM, placas mães monstruosas com barramentos de 1333 pode até não parecer incomodo; mas em um servidor isso ainda é uma boa prática a ser mantida.

```

gabriel@tocaDoTux:~$ uptime
 19:19:16 up 2:19, 2 users, load average: 0,43, 0,44, 0,42
gabriel@tocaDoTux:~$ █

```

(http://2.bp.blogspot.com/-aUUZcedZDOo/VQw-an84HPI/AAAAAAAAs/e_KVwZYiydA/s1600/uptime.png)

(verificando a quanto tempo o computador com o comando uptime)

O modular, desde que seus módulos são carregáveis (LKM: Loadable Kernel Modules), proporciona a vantagem de executar somente o que for necessário. Isso mantém um kernel mais enxuto e faz com que mantenha os recursos computacionais o mais livre possível (como uso de memória e de processamento) caso não haja necessidade de utilizá-los. Ele permite ao usuário carregar e descarregar os módulos dinamicamente da memória com o sistema em execução utilizando o comando **modprobe**:

```
root@tocaDoTux:~# 
root@tocaDoTux:~# 
root@tocaDoTux:~# modprobe -r rfcomm bnep bluetooth
root@tocaDoTux:~# █
```

(removendo três módulos que não estão sendo utilizados)

É possível listar os módulos carregados com o comando **lsmod**:

```
root@tocaDoTux:~# lsmod
Module           Size  Used by
ppdev            12763  0
lp                17149  0
rfcomm            33700  0
bnep              17567  2
bluetooth        119455 10 bnep,rfcomm
rfkill             19012  3 bluetooth
binfmt_misc       12957  1
nfsd              216181  2
nfs               308353  0
nfs_acl           12511  2 nfs,nfsd
auth_rpcgss        37143  2 nfs,nfsd
fscache            36739  1 nfs
lockd              67306  2 nfs,nfsd
sunrpc            173730  6 lockd,auth_rpcgss,nfs_acl,nfs,nfsd
xfs                590943  1
loop                22641  0
kvm_intel          122053  0
kvm                291965  1 kvm_intel
snd_hda_codec_hdmi  30824  4
joydev             17266  0
usbhid             36418  0
hid                81372  1 usbhid
uvcvideo            57744  0
snd_hda_codec_via   41160  1
videodev            70889  1 uvcvideo
v4l2_compat_ioctl32 16655  1 videodev
media               18148  2 videodev,uvcvideo
snd_hda_intel        26259  3
nouveau             583385  4
snd_hda_codec        78031  3 snd_hda_intel,snd_hda_codec_via,snd_hda_codec_hdmi
snd_hwdep            13186  1 snd_hda_codec
snd_pcm              68083  3 snd_hda_codec,snd_hda_intel,snd_hda_codec_hdmi
psmouse             69265  0
snd_page_alloc       13003  2 snd_pcm,snd_hda_intel
evdev               17562  14
pcspkr              12579  0
mxm_wmi              12515  1 nouveau
wmi                 13243  2 mxm_wmi,nouveau
video                17683  1 nouveau
snd_seq              45126  0
```

Com o **modinfo** obtém-se informação sobre o módulo em questão:

```
root@tocaDoTux:~# modinfo ppdev
filename:      /lib/modules/3.2.0-4-amd64/kernel/drivers/char/ppdev.ko
alias:         char-major-99-
license:       GPL
depends:      parport
intree:        Y
vermagic:     3.2.0-4-amd64 SMP mod_unload modversions
root@tocaDoTux:#
root@tocaDoTux:~# modinfo snd
filename:      /lib/modules/3.2.0-4-amd64/kernel/sound/core/snd.ko
alias:         char-major-116-
license:       GPL
description:   Advanced Linux Sound Architecture driver for soundcards.
author:        Jaroslav Kysela <perex@perex.cz>
license:       GPL
description:   Jack detection support for ALSA
author:        Mark Brown <bronie@opensource.wolfsonmicro.com>
depends:      soundcore
intree:        Y
vermagic:     3.2.0-4-amd64 SMP mod_unload modversions
parm:          slots:Module names assigned to the slots. (array of charp)
parm:          major:Major # for sound driver. (int)
parm:          cards_limit:Count of auto-loadable soundcards. (int)
root@tocaDoTux:~
root@tocaDoTux:~# modinfo kvm
filename:      /lib/modules/3.2.0-4-amd64/kernel/arch/x86/kvm/kvm.ko
license:       GPL
author:        Qumranet
depends:      Y
intree:        Y
vermagic:     3.2.0-4-amd64 SMP mod_unload modversions
parm:          oos_shadow:bool
parm:          ignore_msrs:bool
parm:          min_timer_period_us:uint
parm:          allow_unsafe_assigned_interrupts:Enable device assignment on platforms without interrupt remapping support. (bool)
root@tocaDoTux:~#
```

(verificando)

informações sobre os módulos **ppdev**, **snd**, e **kvm**) (<http://1.bp.blogspot.com/-T1XnW0IMOpA/VQxJRis0lzl/AAAAAAAAPk/TOHUPYZ7HU/s1600/modinfo.png>)

O lado considerado desvantajoso do modular é que ele pode apresentar desempenho menor se comparado ao monolítico (mesmo não de forma drástica) devido o kernel ter que requisitar os módulos externamente (estando dentro do kernel, a ação é mais rápida). O segundo ponto considerado desvantajoso é a parte de segurança, o modular pode apresentar vulnerabilidade desde que ataques podem (talvez) ser explorados e realizados através de seus módulos. Os módulos podem se tornar alvos passíveis de ataques uma vez estando fora do kernel.

Algo parecido com isso em um sistema são as bibliotecas. Bibliotecas podem ser divididas em três tipos: Bibliotecas estáticas, compartilhadas e dinamicamente carregáveis (ou simplesmente dinâmicas).

```
gabriel@tocaDoTux:~$ ldd /bin/ls
    linux-vdso.so.1 => (0x00007ffffa8ba2000)
    libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007fa805f04000)
    libert.so.1 => /lib/x86_64-linux-gnu/libert.so.1 (0x00007fa805fcfc000)
    libacl.so.1 => /lib/x86_64-linux-gnu/libacl.so.1 (0x00007fa805af2000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fa805767000)
    libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fa805563000)
    /lib64/ld-linux-x86-64.so.2 (0x00007fa80613d000)
    libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007fa805346000)
    libattr.so.1 => /lib/x86_64-linux-gnu/libattr.so.1 (0x00007fa805141000)
gabriel@tocaDoTux:~$
```

(verificando)

as dependências de bibliotecas do comando **ls** com o comando **ldd**) (<http://1.bp.blogspot.com/-CWRyduoO-88/VQxLPz32xGI/AAAAAAAAP0/YRTtZg8dVq0/s1600/ldd.png>)

```
gabriel@tocaDoTux:~/Downloads/speedt/src$ file speedt
speedt: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs)
, for GNU/Linux 2.6.26, BuildID[sha1]=0x705dd52ac6d3139c657d5f939f0a562a58e34c8e, not stripped
gabriel@tocaDoTux:~/Downloads/speedt/src$
gabriel@tocaDoTux:~/Downloads/speedt/src$ ldd speedt
    linux-vdso.so.1 => (0x00007fff0c1c6000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fd02f725000)
    /lib64/ld-linux-x86-64.so.2 (0x00007fd02fac9000)
gabriel@tocaDoTux:~/Downloads/speedt/src$
```

(Exibindo as informações e dependências do speedt)

```
gabriel@tocaDoTux:~/Downloads$ file papyrus-1.0.0-linux-x64-installer.run
papyrus-1.0.0-linux-x64-installer.run: ELF 64-bit LSB executable, x86-64, version 1
(GNU/Linux), statically linked, stripped
gabriel@tocaDoTux:~/Downloads$ ldd papyrus-1.0.0-linux-x64-installer.run
    not a dynamic executable
gabriel@tocaDoTux:~/Downloads$
```

(<http://2.bp.blogspot.com/>-

UyMHq7awfLA/VRL7yU_I8hI/AAAAAAAASI/txyMyL80iOM/s1600/staticallyLinked.png)

[Exibindo as informações do programa papyrus, onde através dos comandos file e ldd (o mesmo comando usado acima para verificar as dependências dos comandos ls e speedt) é possível notar que é um programa em que suas bibliotecas são estáticas (bibliotecas residem dentro do próprio programa)]

Somente para ficar fixo na mente, vou usar então as bibliotecas estáticas para compará-las com o kernel monolítico e as dinâmicas para compará-las com o kernel modular. As bibliotecas estáticas são bibliotecas estáticas que são instaladas nos programas antes que possam ser executados, ou seja, residem dentro do programa. Já as bibliotecas dinâmicas são carregadas e utilizadas a qualquer momento enquanto o programa estiver em execução.

A escolha em que se determina se quais recursos serão inclusos no kernel Linux ou modularizados é feita durante o processo de configuração para compilá-lo e instalá-lo.

.config - Linux/x86 3.19.1 Kernel Configuration

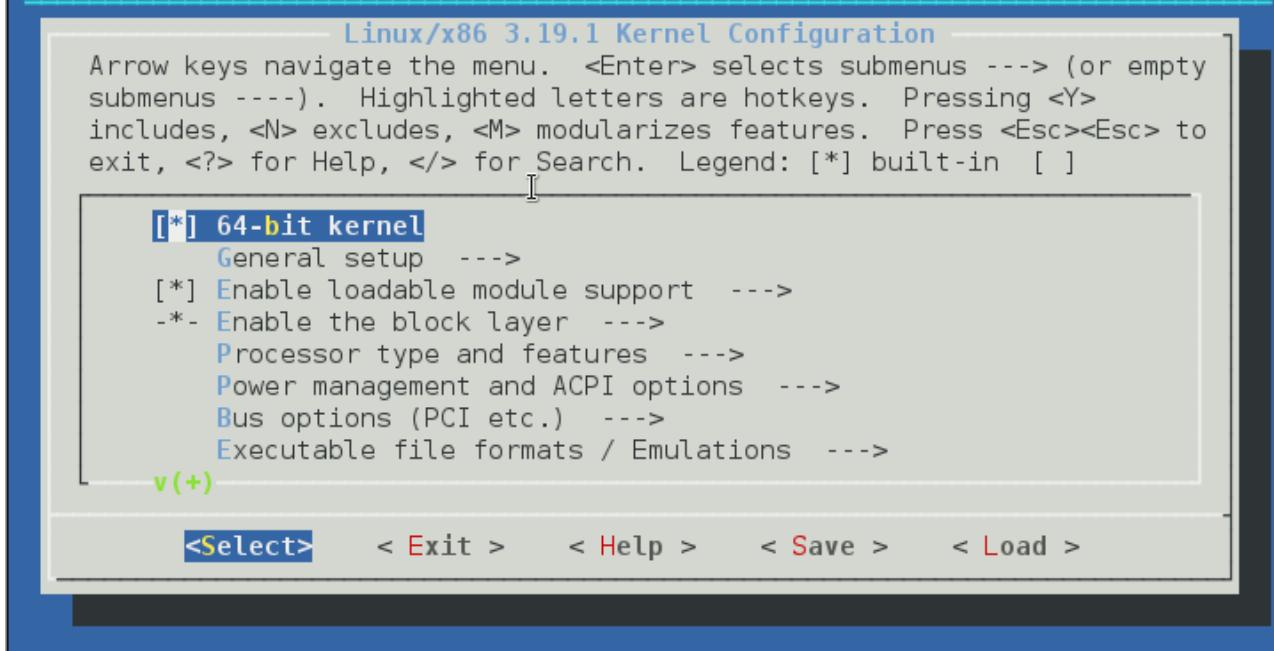
Linux/x86 3.19.1 Kernel Configuration

Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in []

(<http://3.bp.blogspot.com/>-

lhkAPH6__VM/VQxbFl3pjDI/AAAAAAAQQo/GYgrq_ZgdQQ/s1600/kernelMenu.png)

(informações de escolha do kernel na parte superior do painel de configuração do kernel)

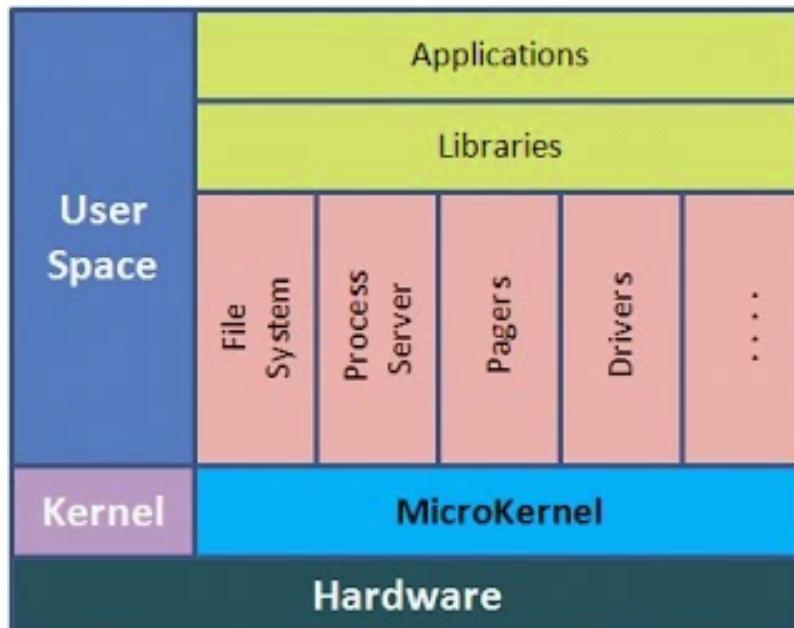


(<http://4.bp.blogspot.com/-Gx-dkwjn95Y/VQxbunYf1bI/AAAAAAAQw/2WIHXF6NMPA/s1600/kernelMenu2.png>)

Obs.: Não confundir com sistema operacional modular (**microkernel**).

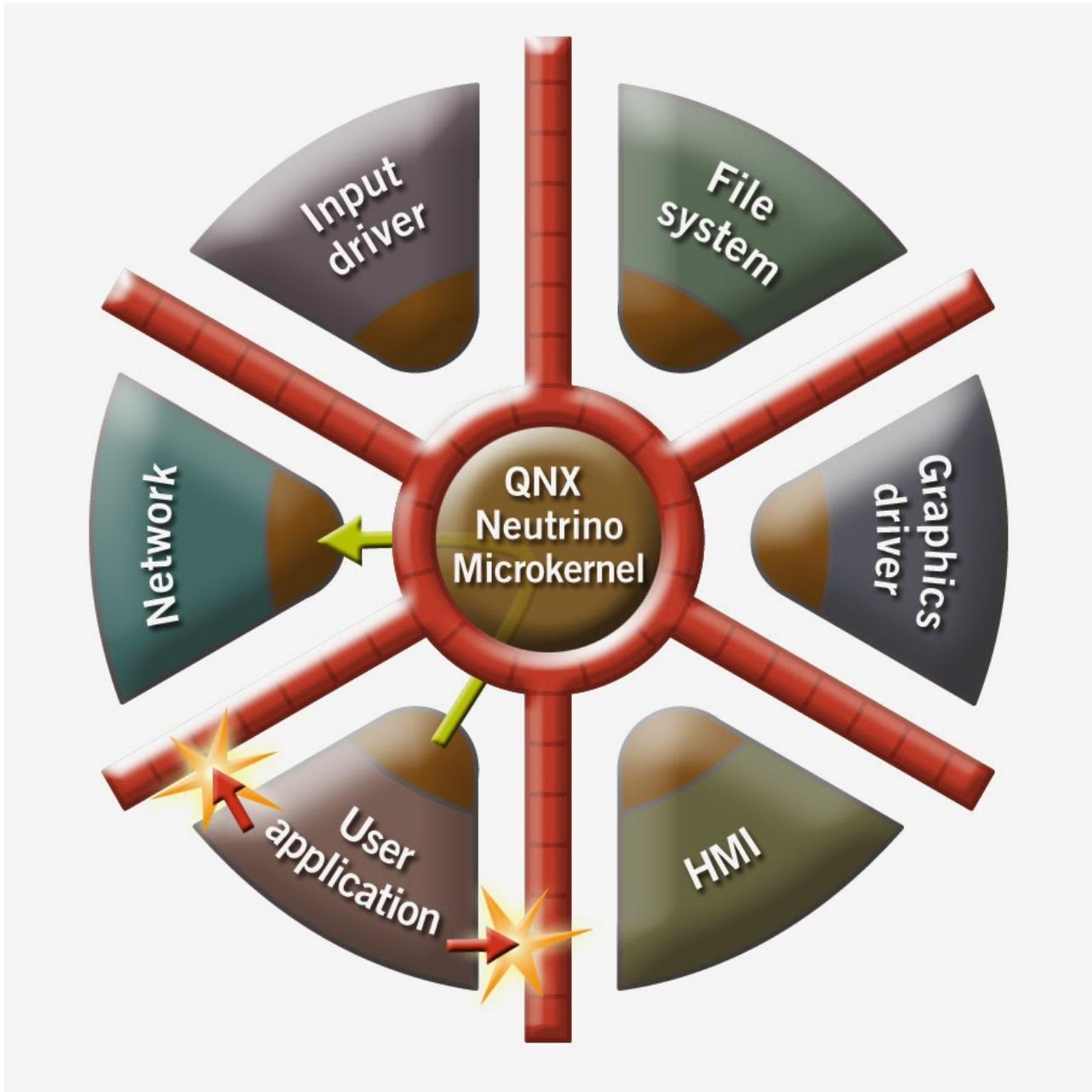
Microkernel (micro-núcleo em português) ou Client-Server:

É um núcleo minuscúlo que trabalha somente com o mínimo de processos possíveis, essenciais para manter o sistema em funcionamento, executando-os no kernel space. Todos os demais processos são executados por daemons conhecidas como servidores de forma isolada e protegidos no user space.



(http://4.bp.blogspot.com/-eUWHsd8Zlms/VQxAxDX-41I/AAAAAAAQ8/rr_oDh-aXzQ/s1600/main-qimg-0ad32e0008b2176e7e1089bca2804620.jpg)

O sistema fica dividido mais ou menos assim: Servidor I/O , servidor de memória, servidor de gerenciamento de processos, servidor de sistema de arquivos, servidor de device drivers e etc. Esses servidores se comunicam com o microkernel; o sistema monitora continuamente cada um destes processos e se uma falha for detectada, ele substitui automaticamente este processo defeituoso sem reiniciar a máquina (reboot), sem perturbar os outros processos em execução e, principalmente, sem que o usuário perceba.



(http://1.bp.blogspot.com/-4wl98rkjToU/VQxA4XBbFII/AAAAAAAAPk/kaLpnsS2XGE/s1600/qnx_microkernel_wheel.jpg)

(visão geral de um microkernel)

A ideia surgiu na década de 80 onde visavam trabalhar em um kernel que fosse mais seguro, mais estável, mais funcional do que o modelo monolítico e considerando o modelo monolítico obsoleto. Em que sentido? Explicam que o modelo monolítico possui a desvantagem de que uma pane no kernel pode causar uma pane no sistema inteiro, enquanto no modelo microkernel, se uma pane ocorrer, ainda é possível evitar uma pane no sistema inteiro bastando reiniciar o serviço que causou tal pane.

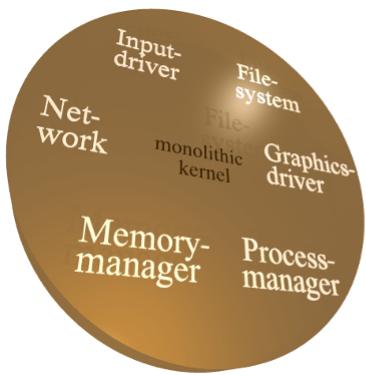


Ilustração de como funciona o kernel monolítico



E de como ficaria caso ocorra alguma pane



Ilustração de como é funciona o micrkernel



E de como ficaria caso ocorra algum pane

(http://4.bp.blogspot.com/-HIUFhj9VkbM/VQxj_z9HDeI/AAAAAAAARc/2R28443FCGM/s1600/compara%C3%A7%C3%A3o%2Bde%2Bmodelos%2Bde%2Bkernels%2Bmonol%C3%ADtico%2Be%2Bmicro.png)

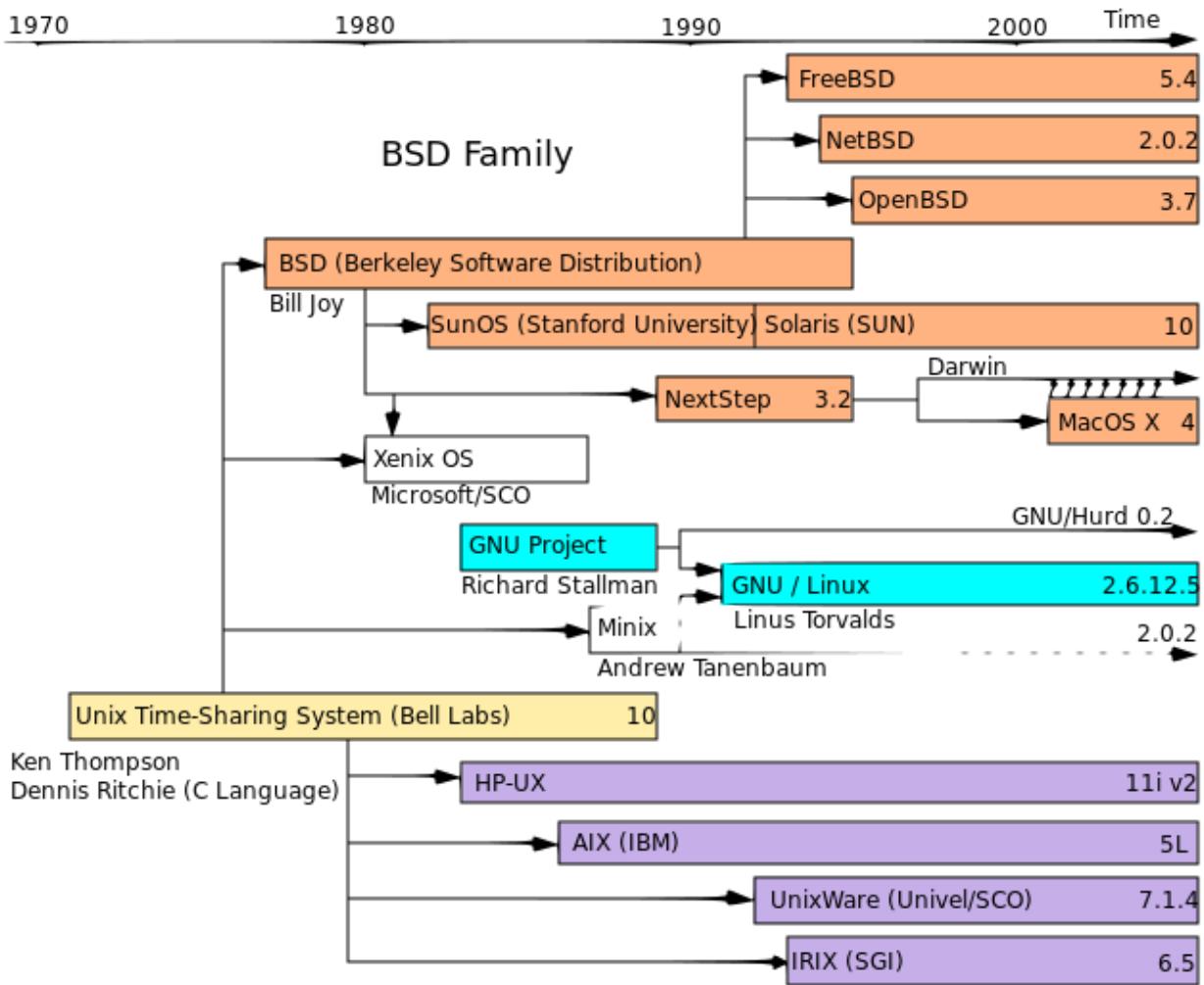
Exemplo de sistemas operacionais microkernel:

- **Minix**

(<http://www.minix3.org/>) quase todos nós que conhecemos Linux conhece esse sistema. Ele é voltado a didática, para quem quer conhecer como um sistema operacional funciona realmente. Pode-se dessecá-lo e analisá-lo. Não tenho críticas ao sistema desde que essa é a intenção do sistema. Linus mesmo afirma na página 87 do Livro “Só por prazer” que Andrew o truncou de propósito, de forma ruim para que estudantes pudessem descobrir os erros por conta própria.

- **Hurd**

(<https://www.gnu.org/software/hurd/hurd.html>) é o microkernel do sistema GNU que vem sido desenvolvido desde 1.990. Ainda não está pronto e nem há previsão para a sua versão estável. É... sim, GNU é um sistema operacional desenvolvido pelo Free Software Foundation. Já vi discussões em que afirmam que GNU não é um sistema e sim um projeto. Tudo por causa da imagem abaixo.



(https://upload.wikimedia.org/wikipedia/en/thumb/8/85/Timeline_of_Unix_families.svg/655px-Timeline_of_Unix_families.svg.png)

Mas veja o texto abaixo sobre o GNU extraído do próprio website do projeto em <https://www.gnu.org/#More-GNU> (<https://www.gnu.org/#More-GNU>):

More about GNU

GNU is a Unix-like operating system. That means it is a collection of many programs: applications, libraries, developer tools, even games. The development of GNU, started in January 1984, is known as the GNU Project. Many of the programs in GNU are released under the auspices of the GNU Project; those we call GNU packages.

The name “GNU” is a recursive acronym for “GNU's Not Unix.” “GNU” is pronounced g'noo, as one syllable, like saying “grew” but replacing the r with n.

The program in a Unix-like system that allocates machine resources and talk to the hardware is called the “kernel”. GNU is typically used with a kernel called Linux. This combination is the GNU/Linux operating system. GNU/Linux is used by millions, though many call it “Linux” by mistake.

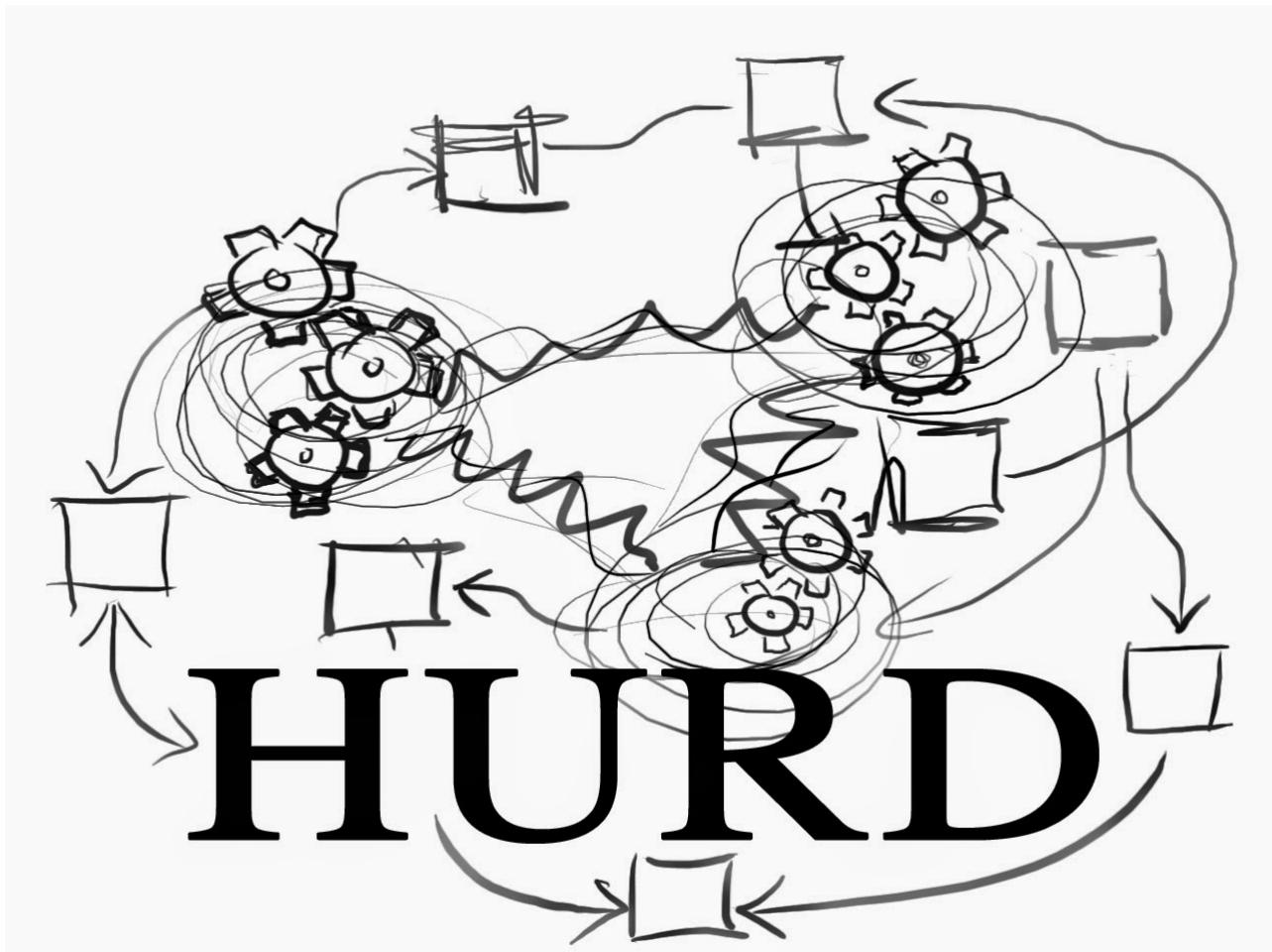
GNU's own kernel, The Hurd, was started in 1990 (before Linux was started). Volunteers continue developing the Hurd because it is an interesting technical project.

Veja

também

(<https://www.gnu.org/philosophy/philosophy.pt-br.html>)

<https://www.gnu.org/philosophy/philosophy.pt-br.html>



(<http://3.bp.blogspot.com/-AXfoiJEBjSQ/VSgnatDTITI/AAAAAAAAYY/AlyrdqJKIDs/s1600/HURDjpgzIO9wZJ7dT.jpg>)

- **QNX**

(<http://crackberry.com/history-qnx-and-it%E2%80%99s-implementation-blackberry-10>)

- **HelenOS**

(<http://www.helenos.org/>) dentre o tais, esse é o que eu mais gosto. Surgiu na universidade da Charles, localizada em Praga, capital da república Tcheca (na mesma cidade onde Jan Hus, evangelista que foi queimado vivo pela igreja católica). Baseado no microkernel SPARTAN, escrito por Jakub Jermář, é um sistema operacional microkernel POSIX-similar projetado do zero, de código aberto, multiservidores (daemons que separam e isolam tarefas no user space como:Naming service, VFS, file system drivers, Location service, device drivers, network layers, graphics stack layers, etc); feito para propósito geral. É utilizado como uma plataforma para aprendizado escolar no curso de sistemas operacionais, como hobby, mas também visam mais duas coisas, que seriam criar um sistema operacional totalmente utilizável em algumas tarefas do dia a dia (como servidor, PDA ou desktop) e a outra seria se divertir. Ele não é um UNIX-like como pode ser lido nessa link

[http://trac.helenos.org/wiki/DiffFromUnix.](http://trac.helenos.org/wiki/DiffFromUnix)

(<http://trac.helenos.org/wiki/DiffFromUnix>.) Eu gosto da ideia da comunidade não ter desavença com a comunidade Linux.

~~HelenOS vs. Linux~~

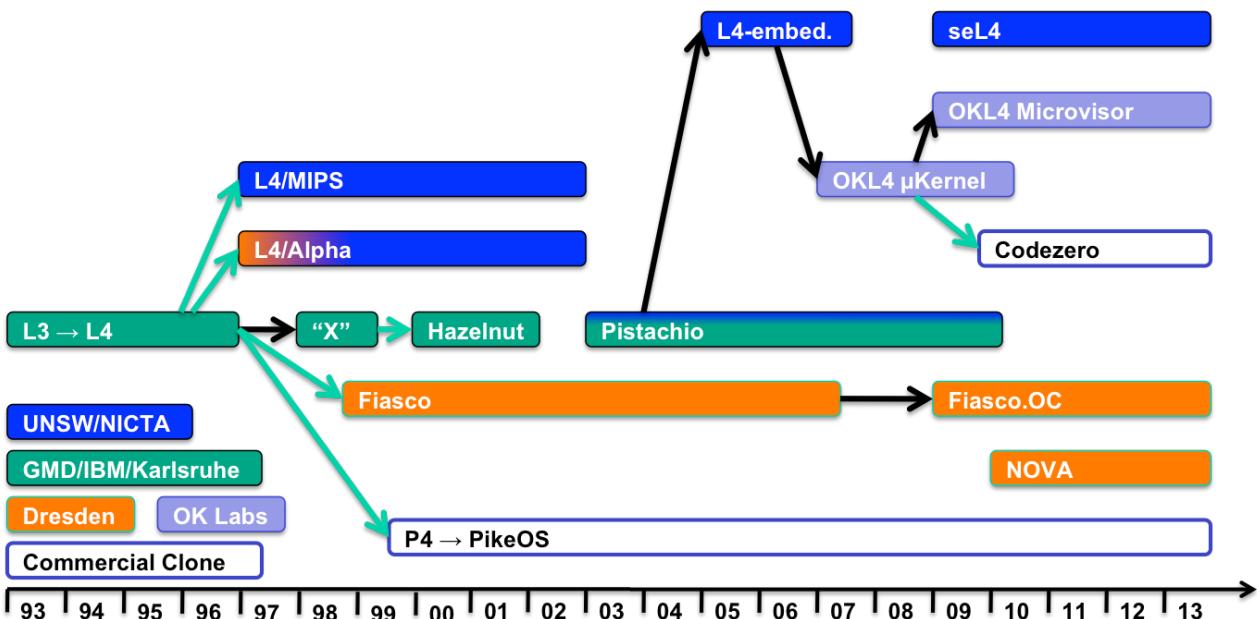
HelenOS and Linux



(<http://3.bp.blogspot.com/-KqV6WfFfrFA/VQxeG-2aW7I/AAAAAAAARM/Ayk3xtOM1a4/s1600/heleOSandLinux.png>)

■ L4

(<http://l4hq.org/projects/kernel/>) É um microkernel escrito do zero, foi criado visando a melhoria de desempenho dos sistemas microkernel. Existem alguns sistemas baseados nesse microkernel; até mesmo um port do Linux para a API do L4 μ-kernel, chamado L4Linux (<http://l4linux.org/>). Existem também algumas implementações desse microkernel, o Code Zero (<https://github.com/jserv/codezero>) foi o primeiro que eu conheci dessa família.



(<http://l4hq.org/projects/kernel/familytree.png>)

■ F9

(<https://github.com/f9micro/f9-kernel>) inspirado na família L4.

■ MonaOS

(<http://monaos.org/>) é um sistema operacional livre que possui micro kernel pequeno (kernel do tamando de 132KB), novo e rápido; escrito em C++. O sistema não é nem POSIX nem um clone Windows. Está sob a licença MIT e roda em processadores Intel IA32.

- **mikro-SINA**

(<http://os.inf.tu-dresden.de/mikrosina/>) desenvolvido na Alemanha e descontinuado em 2004.

- **TUDOS**

(http://demo.tudos.org/intro_tutorial.html)

Quer saber mais sobre Microkernel? Siga o link: <http://microkernel.info/> (<http://microkernel.info/>)

Kernel monolítico X microkernel

Muita se tem discutido sobre o assunto quando o assunto é kernel monolítico ou microkernel. Do lado do microkernel alegam que o modelo monolítico é obsoleto, possui desvantagens de ser vulnerável, não ser portável, seu código fonte pode se tornar enorme e ter menor desempenho. Mas essas afirmações ainda ficam como teoria. Na prática, é difícil afirmar que isso é fato.

Houve até mesmo uma discussão travada Linus e Andrew (<http://www.oreilly.com/openbook/opensources/book/appa.html>) sobre o assunto. Até o Ken Thompson participou do debate afirmando que é mais fácil implementação do modelo monolítico. Linus concorda que o microkernel é melhor, e que de um ponto de vista teórico e estético o Linux perde, porém afirma também que o fato de ser microkernel não é o único critério a se analisar um bom sistema. Discute até mesmo na questão da portabilidade.

Sistemas operacionais que possuem kernel híbrido:

Existe ainda os sistemas entre meio termo, os sistemas que possuem kernel híbrido. A ideia é que se obtenha o desempenho do kernel monolítico e a estabilidade e segurança do microkernel. Linus Torvalds tem dito que a questão de kernel hidrido, como a coisa toda, é apenas markting.

Dentre os sistemas de kernel híbrido, cito os de meu conhecimento:

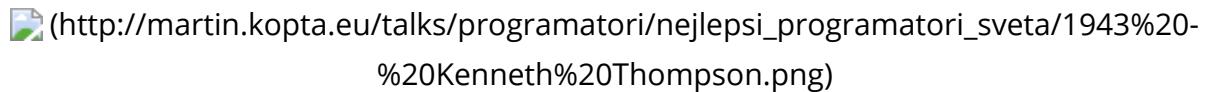
- Mac OSX: Esse foi o primeiro sistema kernel híbrido que conheci. Seu kernel Darwin, é uma hibridação do microkernel Mach e do kernel monolítico FreeBSD. Antigamente seu kernel era chamado de XUN (uma hibridação do 4.3BSD com o Mach 2.5) quando a Next, empresa que Steve Jobs fundou ao ser demitido da Apple possuía o sistema NextStep. Depois, quando a Apple estava chegando à falência, decidiu inovar em uma ultima tentativa. Compraram a Next (para obter o NextStep, por que o Mac OS já estava uma m... Era mais fácil comprar um do que escrever outro do zero) aí uniram o Next com a interface do Mac OS.

- **Haiku**

(<https://www.haiku-os.org/>) Esse eu sei que é desenvolvido por antigos desenvolvedores do BeOS, mas a informação real de que ele é um sistema de kernel híbrido foi difícil de encontrar. So vi treta no site do sistema e desencanei de pesquisar.

- **Plan9**

(<http://plan9.bell-labs.com/plan9/>) Esse é um sistema voltados a pesquisa. Foi desenvolvido por Ken Thompson, criador do Unix e da linguagem C (na época B), Rob Pike, Dave Presotto, and Phil Winterbottom. É um sistema em que cada processo é executado em seu próprio name space mutável.



■ *DragonflyBSD*

(<http://www.dragonflybsd.org/>) Esse foi considerado o primeiro sistema operacional non-Mach que se possui o kernel híbrido no livro da Wikipedia.

A screenshot of a presentation slide. On the left, there is a sidebar with a list of tools: mintty (Cygwin terminal), UnxUtils, and UWIN. Below this is a section titled "5.2.6 References" with a note: "[1] Updated: Cygwin 1.7.33-1". The main content area has a heading "5.3.1 System design" and a sub-section "Kernel". It discusses DragonFly's hybrid kernel, noting it contains features of both monolithic and microkernels, such as the message passing capability of microkernels enabling larger portions of the OS to benefit from protected memory, as well as retaining the speed of monolithic kernels for certain critical tasks. The messaging subsystem being developed is similar to those found in microkernels such as Mach. There is also a note about shared resources protection.

150

CHAPTER 5. UNIX-LIKE OPERATING ENVIRONMENTS

passing capability of microkernels enabling larger portions of the OS to benefit from protected memory, as well as retaining the speed of monolithic kernels for certain critical tasks. The messaging subsystem being developed is similar to those found in microkernels such as Mach,

cessor), reducing competition by removing the need to share certain resources among various kernel tasks.^[4]

Shared resources protection In order to run safely on



Mas em um post no grupo oficial do sistema em uma rede social, eu encontrei a informação de que ele é na verdade monolítico. O que me fez pesquisar melhor sobre o sistema.



[REDACTED]

DragonFly BSD

March 25, 2013 ·

▼

Is DragonFly BSD suitable as a kernel for smartphones?

Reading a lot of that OS, it crossed my mind that phones would be greatly benefited from those microkernel-like features.

[Like](#) · [Comment](#) · [Share](#)



[REDACTED] DragonFly is by and large monolithic, just like Linux and the other BSD's.

September 13, 2013 at 10:32am · [Like](#)



Write a comment...



(<http://4.bp.blogspot.com/-10q6qt5HBdc/VQxS5sqavGI/AAAAAAAQQ/8OQjbD8nKU0/s1600/dragonfly mono.png>)

foi então onde entrei em contato com Matt Dillon e ele me afirmou o Dragonfly é tão monolítico quanto o kernel Linux e o kernel FreeBSD:



Matthew Dillon

para mim ▾

15:21 (Há 4 horas) ★



▼



inglês ▾

português ▾

[Traduzir mensagem](#)

[Desativar para: inglês](#) ×

Loadable kernel modules have kinda blurred the lines. DragonFly is just as monolithic as the linux or FreeBSD kernel is.

-Matt



Clique aqui para [Responder](#) ou [Encaminhar](#)

(http://2.bp.blogspot.com/-AdPZHbrT05c/VQxTtY_FF4I/AAAAAAAQQY/hSEbLa0MeGw/s1600/matt.png)

O que o DragonflyBSD faz é gerar o kernel real gera um kernel virtual e trata o processo dentro de uma vmspace. Quando o vkernel termina o processo, o kernel real destrói o vmspace com o processo e o vkernel.

O NT kernel (utilizado no Windows NT, 2000, XP, Vista e Windows 7) é um kernel híbrido. Francamente? Achei muito difícil conseguir informações deste tipo sobre o Windows. Por fim só consegui alguma informação sobre o kernel NT no web site do ReactOS [se alguém tiver alguma informação, posta aí :)]:

<https://www.reactos.org/wiki/Kernel> (<https://www.reactos.org/wiki/Kernel>)

Bom, existe muitos e muitos sistemas no mundo por aí, mas isso é uma questão de pesquisas a serem realizadas. Incentivo a todos a fazerem isso para aprofundarem seus conhecimentos.

Estou deixando alguns links para poderem acompanhar as comparações entre os sistemas:

- [\(https://en.wikipedia.org/wiki/Comparison_of_operating_system_kernels\)](https://en.wikipedia.org/wiki/Comparison_of_operating_system_kernels)
- [\(https://en.wikipedia.org/wiki/Comparison_of_open-source_operating_systems\)](https://en.wikipedia.org/wiki/Comparison_of_open-source_operating_systems)
- [\(http://www.brendangregg.com/blog/2015-03-17/linux-performance-analysis-perf-tools.html\)](http://www.brendangregg.com/blog/2015-03-17/linux-performance-analysis-perf-tools.html)
- critica do Linus sobre kernel híbrido (<http://www.realworldtech.com/forum/?threadid=65915&curpostid=65936>)

Não deixem de se inscrever no canal para continuar acompanhando. Dê seu like e compartilhe se gostou das informações. Tem algo para conhecimento a acrescentar? Compartilhe aqui nos comentários. Vale lembrar que inscritos no blog tem direito a 5% de desconto na LyTux (<https://plus.google.com/+LytuxBr/videos>) (tanto para cursos quanto para consultoria em serviços). Alias, vou dar uma breve sumida, por que estou fazendo o meu curso de Asterisk na LyTux. Beleza galera? Um abraço e até a próxima.



(<http://creativecommons.org/licenses/by-sa/4.0/>)

Diferentes modelos de kernels: Monolítico (e modular), microkernel, kernel híbrido (entre os dois primeiros), nanokernel e hexokernel. De Gabriel da Silveira Costa (Toca do Tux) (<http://tocadotux.blogspot.com.br/>) está licenciado com uma Licença Creative Commons - Atribuição-Compartilhamento 4.0 Internacional (<http://creativecommons.org/licenses/by-sa/4.0/>). Baseado no trabalho disponível em <http://tocadotux.blogspot.com/2015/03/diferentes-modelos-de-kernels.html>. Podem estar disponíveis autorizações adicionais às concedidas no âmbito desta licença em <http://tocadotux.blogspot.com/2015/03/diferentes-modelos-de-kernels.html> (<http://tocadotux.blogspot.com/2015/03/diferentes-modelos-de-kernels.html>).

Inscreva-se em nossa Newsletter

Email...

Enviar

Compre na imago brinquedos



(<http://www.imagobrinquedos.com.br/>)

Utilize o cupom de desconto TOCADOTUX e economize 5% na sua compra