

Kernel

Tiago Heinrich

UniSociesc Joinville

16/04/2020

Introdução

- 7 Milhões de linhas (sendo o sistema mais flexível já criado)
- Disponibilizar o Linux em outra arquiteturas
- Sendo aplicado em um vasto conjunto de aplicações TV, radio, smartphones, ... Super computadores (Grid 5000)
- Kernel para seu ambiente específico (10Mb...109Mb)
- (GPL) aberto e disponível para qualquer um estudar

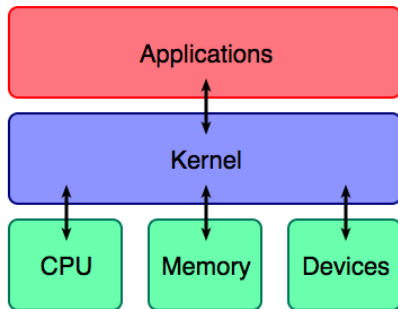
Motivos?

- Linux é gratuito
- O Linux é totalmente personalizável em todos os seus componentes
- O Linux roda em plataformas de hardware baratas e de baixo custo
- Linux é poderoso
- O kernel do Linux pode ser muito pequeno e compacto
- O Linux é altamente compatível com muitos sistemas operacionais comuns.
- Linux é bem suportado

Função

- É carregado na RAM quando o sistema inicializa e contém procedimentos críticos necessários para o sistema para operar
- Interage com os componentes de hardware, atendendo a todos os níveis programáveis de baixo nível elementos incluídos na plataforma de hardware
- Fornece um ambiente de execução para os aplicativos executados no computador sistema
- O sistema operacional semelhante ao Unix oculta todos os detalhes de baixo nível relativos à organização física do computador de aplicativos executados pelo usuário

Função



- O código crítico do kernel geralmente é carregado em uma área separada da memória
- O kernel executa tarefa como:
 - Executar processos
 - Gerenciar dispositivos de hardware como o disco rígido
 - Manipular interrupções neste espaço protegido do kernel (Kernel space).

O sistema operacional segrega memória virtual em dois grupos:

- *user space* e *kernel space*
- Principalmente, essa separação serve para fornecer proteção de memória e hardware
- O espaço do kernel é estritamente reservado para a execução de privilégios de kernel do sistema operacional, extensões do kernel e a drivers de dispositivo

- Componentes de sistema de baixo nível
 - Applications (Bash, Firefox, LibreOffice, nmap ...)
 - System daemons (systemd, runit, logind, networkd, PulseAudio)
 - Windowing system (X11)
 - Libraries (GTK, Qt...)
 - Graphics (Mesa)
- *C standard library* (glibc e POSIX)
 - `open()` , `exec()` , `socket()` , `fopen()` ,
`calloc()` ...

Kernel space

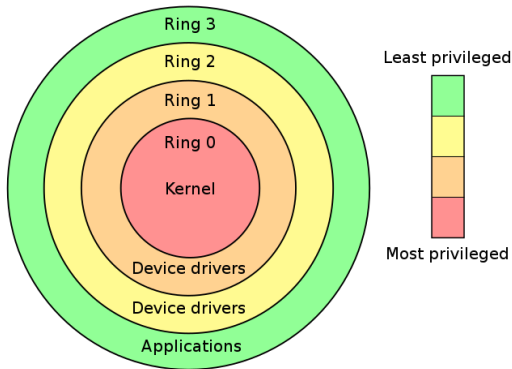
- O Linux kernel System Call Interface (stat, read, open, write, mmap, close, exit ... 390 chamadas)
- Agendamento de processo
- Inter-process communications (IPC)
- Network
- Virtualização
- LVM, Netfilter, SELinux, AppArmor...

System call

- Responsável por solicitar um serviço para o kernel
 - Criação e execução de novos processos
 - Comunicação com serviços do kernel (programação de processos)
- As chamadas do sistema fornecem uma interface essencial entre um processo e o sistema operacional
- Security model

Protection ring

- Mecanismos para proteger dados e funcionalidade de falhas



Source

- Até a versão 2.4, duas *Tree*: “development” e “stable”
- Atualmente só existe a “stable”
- Um ciclo dura entre dois e três meses.

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

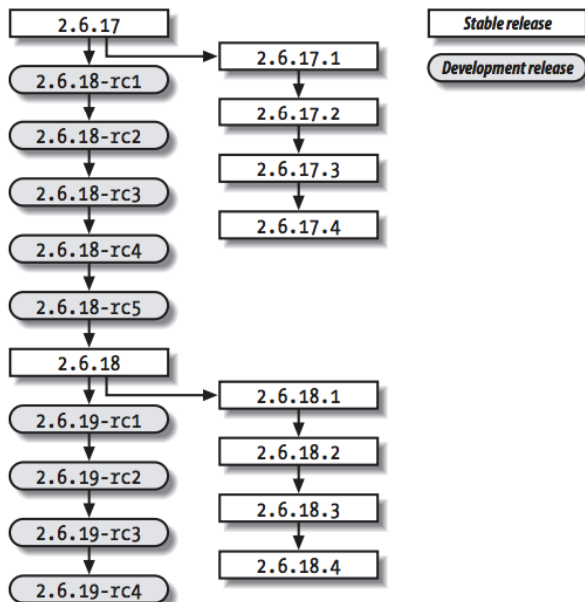
Latest Stable Kernel:



5.6.4

mainline:	5.7-rc1	2020-04-12	[tarball]	[patch]	[view diff]	[browse]	
stable:	5.6.4	2020-04-13	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
stable:	5.5.17	2020-04-13	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	5.4.32	2020-04-13	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	4.19.115	2020-04-13	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	4.14.176	2020-04-13	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	4.9.219	2020-04-13	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	4.4.219	2020-04-13	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.16.82	2020-02-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
linux-next:	next-20200413	2020-04-13					[browse]

Source



Configuração

```
# Download Kernel
```

```
wget https://cdn.kernel.org/pub/linux/  
kernel/v5.x/linux-5.6.4.tar.xz
```

```
# Create Folder
```

```
mkdir linux  
mv linux-5.6.4.tar.xz linux/; cd linux/  
tar -xf linux-5.6.4.tar.xz
```

Configuração

- Para verificar TODAS as configurações **make config** (mais de 2k opções) (Linux From Scratch)
- Utilizar configurações default **make defconfig**
- Alterar configurações **make menuconfig**
- Por fim: **make -j4**

Instalação & Upgrade

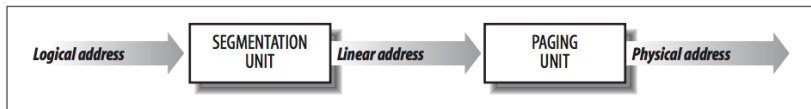
- 1 Obtenha o novo código fonte.
- 2 Aplique as alterações na árvore de origem antiga para trazê-la para o nível mais recente.
 - Aplicar os patch
- 3 Reconfigure o kernel com base na configuração anterior do kernel.
- 4 Crie o novo kernel.
- 5 Instale o novo kernel.

Conceitos básicos do Kernel

Os programadores casualmente se referem a um endereço de memória como a maneira de acessar o conteúdo de uma célula de memória.

- **Logical address**
- **Linear address** (Virtual)
- **Physical address**

Memory Addresses



- Nos sistemas multiprocessadores, todas as CPUs geralmente compartilham a mesma memória
- O Memory Management Unit (MMU) é responsável por converter endereços Logical para Linear
- A paginação é responsável por converter o endereço Linear para Physical.
- **Memory Management**

- Um processo é geralmente definido como uma instância de um programa em execução (*tasks* ou *threads*)
- Os processos são como seres humanos: são gerados, têm uma vida mais ou menos significativa, opcionalmente geram um ou mais processos filhos e, eventualmente, eles morrem. Cada processo tem apenas um pai. (Daniel P.)

Interrupts e Exceptions

- Uma interrupção é geralmente definida como um evento que altera a sequência de instruções executado por um processador
 - Interrupções síncronas são produzidas pela unidade de controle da CPU durante a execução
 - Interrupções assíncronas são geradas por outros dispositivos de hardware
- Fornecer uma maneira de desviar o processador para o código fora do fluxo normal de controle.

Interrupts e Exceptions

- O tratamento de interrupção é uma das tarefas mais sensíveis executadas pelo kernel
- Quando um sinal de interrupção chega, a CPU deve interromper o que está fazendo e mudar para uma nova atividade
- As interrupções podem ocorrer a qualquer momento (o kernel pode estar lidando com um deles enquanto outro chega)
- Regiões críticas existem dentro do código do kernel, onde as interrupções devem ser desabilitado

Process Scheduling

- Agendamento (*scheduling*), diz respeito a quando mudar e qual processo escolher
 - Execução de múltiplos processos/usuários no sistema
- Os sistemas operacionais Unix devem atender a vários objetivos conflitantes
 - Tempo de resposta rápido do processo
 - Bom rendimento para segundo plano empregos
 - Evitar a inanição de processos (*starvation*)
 - Reconciliação das necessidades de processos de baixa e alta prioridade
 - ...

Process Scheduling

O agendamento do Linux é baseado na técnica de compartilhamento de tempo:

- O tempo da CPU é dividido em fatias, uma para cada processo executável
- Caso o processo em execução no momento não for finalizado quando seu intervalo de tempo ou o tempo expirar, uma troca de processo poderá ocorrer
- Existe a dependência de interrupções neste momento
- O agendamento também se baseia nos processos de classificação de acordo com sua prioridade (**nice** ou **htop**)

Execução de programas

- Diferentes formatos executáveis
- Bibliotecas compartilhadas
 - Arquivos executáveis não contêm todo o código necessário para executar o programa, mas esperam que o kernel carregue funções de uma biblioteca em tempo de execução

Execução de programas

- Para a execução de **ls -l (/bin/ls)**
 - 1 O shell de comando faz um *fork* de um novo processo
 - 2 Este novo processo chama uma *system call* como *execve()*
 - 3 Este conjunto de *system calls* localiza o binário em */bin/ls*
 - 4 Modificações são realizadas para que a função seja correspondente
 - 5 Quando a *system calls* termina, o processo começa a executar o código armazenado no arquivo executável, que executa a listagem do diretório