

Sistemas Operacionais Embarcados

Linux Kernel Modules

Kernel

- O Linux é somente o kernel do sistema operacional
- As distribuições são construídas em cima dele
- <https://github.com/torvalds/linux>
- <https://www.kernel.org/doc/html/latest/>



Kernel

Kernel

- Gerencia o hardware
- Oferece abstrações de alto nível para bibliotecas e programas em modo usuário
 - Dispositivos
 - Acesso de entrada e saída
 - Controle de processos
 - Compartilhamento de memória
 - Sinais
 - Etc.
- O kernel permite que as mesmas aplicações rodem em hardwares diferentes



Kernel

Interfaces de baixo nível

- Controla o hardware
- Muda de acordo com o hardware
- Oferece bibliotecas independentes para as próximas camadas do kernel
- Oferece operações específicas de CPU e memória
- Interfaces básicas para os dispositivos (devices)



Kernel

Abstrações de alto nível

- Processos
- Arquivos
- Sockets
- Sinais
- O código desta camada é praticamente constante, independentemente da arquitetura



Kernel

Sistemas de arquivo

- Abstração lógica para organizar arquivos e diretórios em memória (discos, memória flash etc.)
- Exemplos:
 - FAT16
 - FAT32
 - NTFS
 - EXT3
 - EXT4
 - HFS
 - ExFAT
 - ZFS



Kernel

Protocolos de rede

- Exemplos:
 - TCP/IP (IPv4 e IPv6)
 - IPX/SPX (LAN)
 - AppleTalk
 - WAN Networking
 - ISDN (Modem)
 - PPP (Point-to-Point-Protocol)
 - SLIP (Serial Line IP)
 - PLIP (Parallel Line IP)



Kernel

Bibliotecas

- Um nível de abstração acima do kernel
 - Provém APIs* e serviços abstratos que interagem com o kernel
 - Facilitam o acesso das aplicações ao sistema operacional
- GNU C Library: *glibc* (pode ser substituída por uma menor)
- Bibliotecas podem ser linkadas dinamicamente ou estaticamente, não fazendo parte do binário da aplicação



* Application programming interfaces

Kernel

Daemons

- Aplicações executadas como processos em plano de fundo, em vez de estar sob o controle direto de um usuário interativo
- Geralmente inicializados no boot
 - Requisições de rede
 - Atividade de hardware
 - Configuração de hardware
 - Tarefas agendadas (cron)
 - Etc.
- Geralmente tem um nome de processo terminado com a letra d



Kernel

```
ps -aux | grep "d]"
```

root	2	0.0	0.0	0	0	?	S	nov04	0:00	[kthreadd]
root	6	0.0	0.0	0	0	?	I<	nov04	0:00	[kworker/0:0H-kblockd]
root	11	0.0	0.0	0	0	?	I	nov04	1:27	[rcu_schedd]
root	20	0.0	0.0	0	0	?	I<	nov04	0:00	[kworker/1:0H-kblockd]
root	32	0.0	0.0	0	0	?	I<	nov04	0:00	[kworker/3:0H-kblockd]
root	36	0.0	0.0	0	0	?	S	nov04	0:00	[kauditd]
root	37	0.0	0.0	0	0	?	S	nov04	0:00	[khungtaskd]
root	41	0.0	0.0	0	0	?	SN	nov04	0:00	[ksmd]
root	42	0.0	0.0	0	0	?	SN	nov04	0:00	[khugepagedd]
root	89	0.0	0.0	0	0	?	I<	nov04	0:00	[kintegrityd]
root	90	0.0	0.0	0	0	?	I<	nov04	0:00	[kblockd]
root	96	0.0	0.0	0	0	?	I<	nov04	0:00	[md]
root	101	0.0	0.0	0	0	?	S	nov04	0:00	[watchdogd]
root	106	0.0	0.0	0	0	?	I<	nov04	0:00	[kthrotld]
root	220	0.0	0.0	0	0	?	I<	nov04	0:05	[kworker/2:1H-kblockd]
root	231	0.0	0.0	0	0	?	I<	nov04	0:05	[kworker/0:1H-kblockd]
root	423	0.0	0.0	0	0	?	I<	nov04	0:00	[cryptd]
root	1424	0.0	0.0	314760	10392	?	Sl	nov04	0:00	gdm-session-worker [pam/gdm-passwordd]
root	1496	0.0	0.0	0	0	?	S<	nov04	0:00	[krfcommd]
root	62648	0.0	0.0	0	0	?	I	14:38	0:01	[kworker/u8:1-events_unboundd]
root	63434	0.1	0.0	0	0	?	I	15:01	0:00	[kworker/u8:3-events_unboundd]
diogo	64102	0.0	0.0	9036	668	pts/0	S+	15:10	0:00	grep --color=auto d]

Módulos do kernel*

- Mudanças no kernel devem ser compiladas, e só são reconhecidas depois de um *reboot*
- Módulos do kernel (Linux kernel modules - LKMs) estendem o kernel sem necessitar de um *reboot*
- Para compilar módulos do kernel, é preciso obter o código do kernel Linux para o Raspberry Pi. Execute:

```
$ sudo apt install raspberrypi-kernel-headers
$ sudo wget
https://raw.githubusercontent.com/notro/rpi-source/master/rpi-source
-O /usr/bin/rpi-source && sudo chmod +x /usr/bin/rpi-source &&
/usr/bin/rpi-source -q --tag-update
$ rpi-source
```

* <http://www.tldp.org/LDP/lkmpg/2.6/html/lkmpg.html>

Módulos do kernel

- Se o comando `rpi-source` não completar a instalação, siga as instruções na tela para instalar as dependências que faltam. Por exemplo, se surgir a mensagem:

```
ERROR: bc is NOT installed. Needed by 'make modules_prepare'. On  
Raspbian, run 'sudo apt-get install bc' to install it.
```

Execute o que ele pede:

```
$ sudo apt-get install bc  
$ rpi-source
```

Completando a instalação, execute

```
$ sudo ln -s /usr/src/linux-headers-$(uname -r) /lib/modules/$(uname  
-r)/build
```