



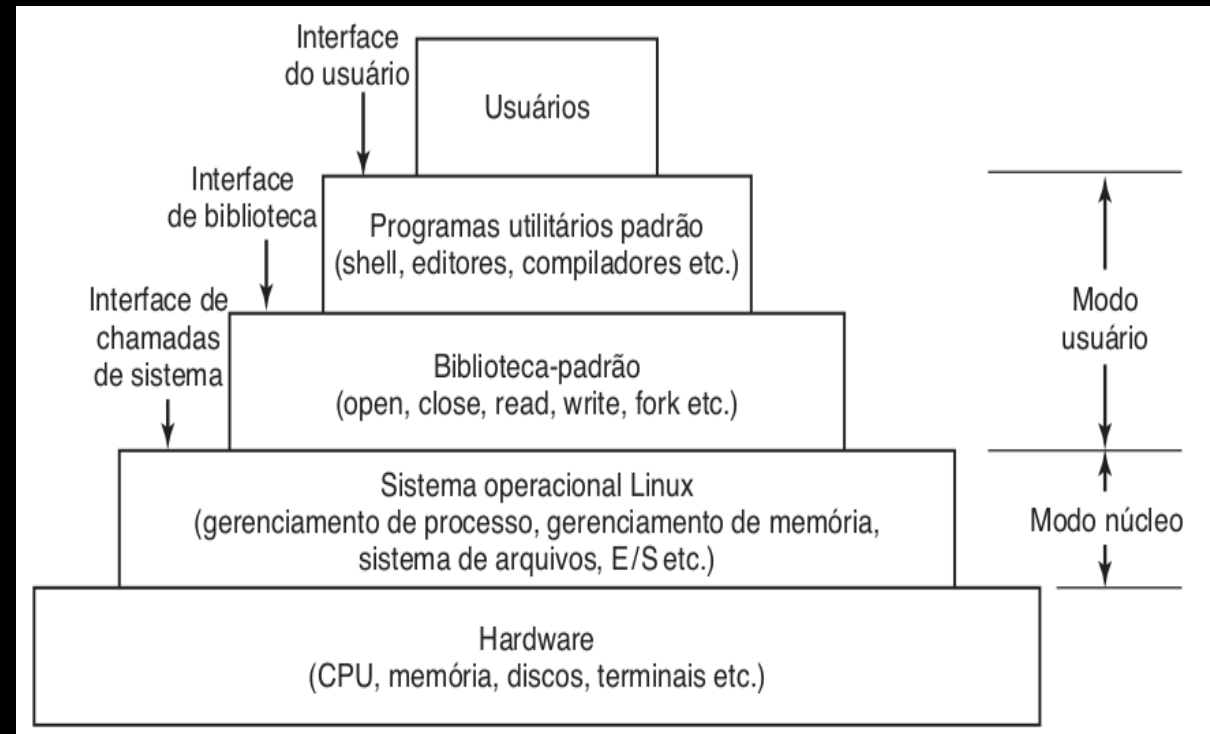
SISTEMAS OPERACIONAIS

IMPLEMENTAÇÃO DO RTAI NO UBUNTU 16.04

KERNEL DO LINUX

FUNCIONAMENTO DO KERNEL

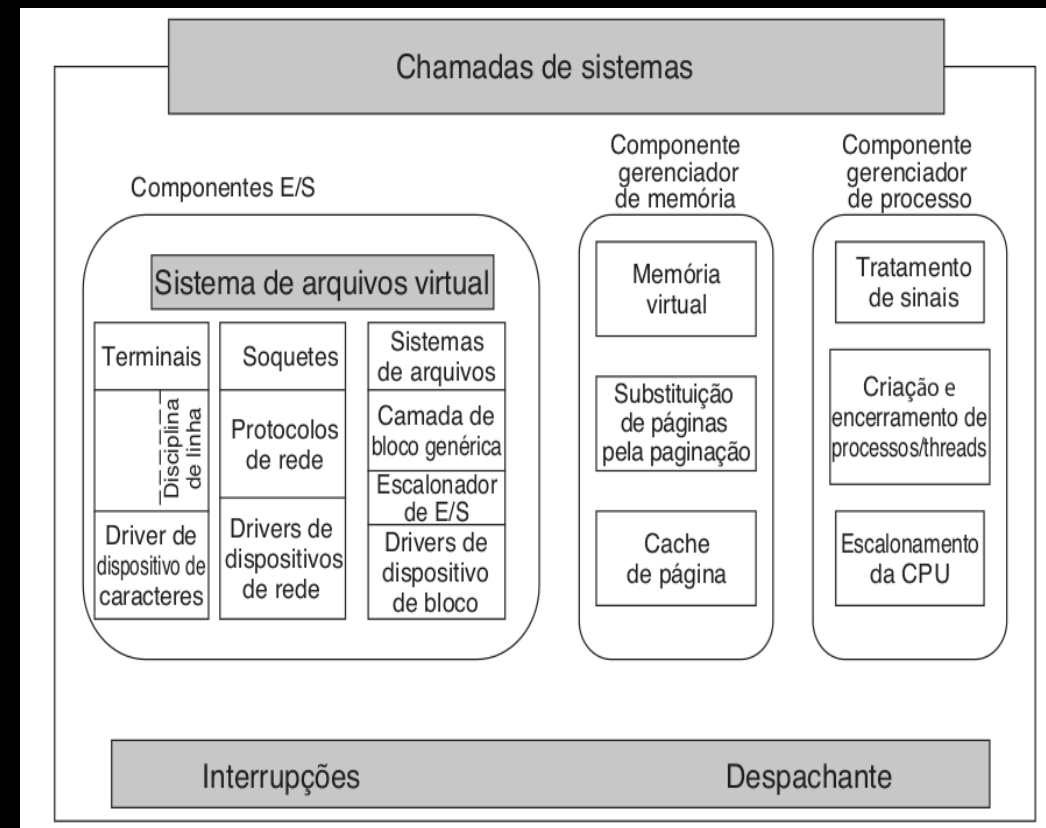
- O KERNEL É CONSIDERADO COMO O NÚCLÉO, ELEMENTO IMPORTANTE DE UM SISTEMA OPERACIONAL
- Hardware: A sua função é controlar o hardware e fornecer uma interface de chamada de sistema para todos os programas.
- chamadas de sistema: usuários criam e gerenciam processos, arquivos e outros recursos.
- Biblioteca: Essas rotinas são escritas em linguagem de montagem, mas podem ser chamadas a partir de C



KERNEL DO LINUX

OS SUBSISTEMAS

- O núcleo encontra-se diretamente sobre o hardware
- Tratadores de interrupção, (a principal maneira de interagir com E/S), e o mecanismo de despacho
- O despacho inicializa o driver apropriado
- as operações de E/S são todas integradas sob a camada VFS
- Todos os drivers Linux são classificados como drivers de dispositivos de caracteres ou drivers de dispositivos de blocos.
- Tarefas de gerenciamento de memória incluem manter os mapeamentos memória virtual para física.
- gerenciamento de processo é a criação e término de processos.



KERNEL DO LINUX

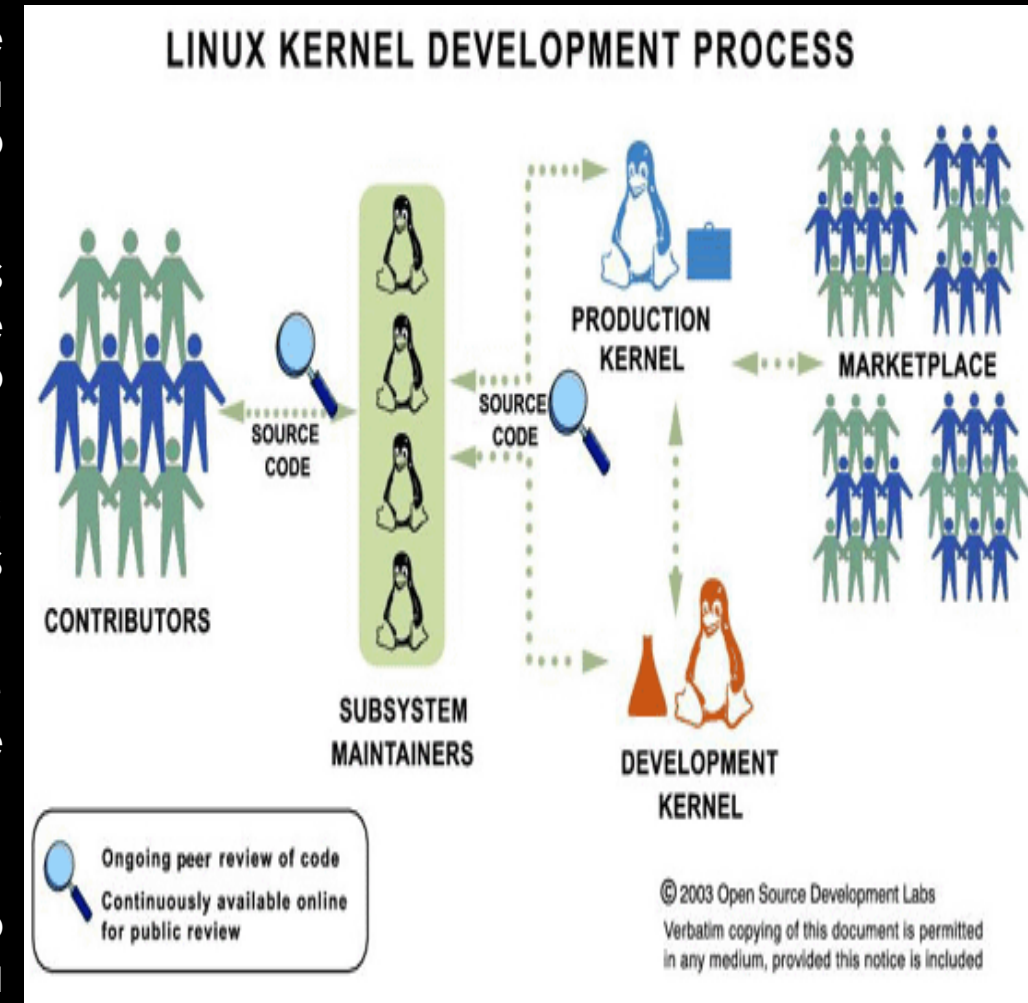
PROCESSO DE DESENVOLVIMENTO

- Patch - arquivo contendo diferenças textuais entre arquivos. Utilizado para descrever diferenças em código-fonte de linguagens de programação.
- Árvore (Source Tree) - repositório do código-fonte de um sistema de controle de versão;
- branch (ramo) - ramificação de um código fonte onde geralmente são implementadas novas funcionalidades;
- Kernel mainline - Kernel Linux "vanilla" ou principal, originado da árvore principal de desenvolvimento, que é gerenciada por Linus Torvalds;
- merge (mesclar) - é a junção de arquivos de código fonte.
- Problemas de regressão (regressions) - bugs em funcionalidades que antes estavam corretas e são causados por um determinado evento.

KERNEL DO LINUX

PROCESSO DE DESENVOLVIMENTO

- Processo de merge de patches:código que é considerado suficientemente estável e aceito pela comunidade de desenvolvimento é incluído no kernel mainline
- Processo de correção do mainline:Durante as próximas 6 até 10 semanas, somente patches que corrigem problemas devem ser submetidos para o mainline
- Processo de criação de Versões:À medida que as correções de bug são feitas no mainline, o número de patches enviados vai cair com o tempo
- Processo de estabilização:Durante o processo de estabilização, o objetivo dos desenvolvedores é resolver o máximo número de problemas (regressions) antes que a versão estável seja lançada
- Processo de Manutenção:Uma vez que o lançamento da versão estável é feito, a manutenção dela é passada para o “stable team”.



KERNEL DO LINUX

OS MODULOS

- Os módulos são blocos de códigos que podem ser carregados no núcleo enquanto o sistema está executando. Mais comumente esses são drivers de dispositivos de bloco ou caracteres, mas eles também podem ser sistemas de arquivos inteiros, protocolos de rede, ferramentas de monitoramento de desempenho, ou qualquer coisa desejada.
- Insere módulo e os demais módulos necessários:
- modprobe [options] module
- insmod [options] module
- Remove módulo: rmmod [options] module
- Mostar as listas de Módulos instalados: lsmod
- Insere, remove, lista módulos carregados e informações sobre um módulo:
- modinfo [options] module}
- Gera dependência entre módulos:
- Cria /lib/modules/*/modules.dep}
- depmod [options] module1.o module2.o...}

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <asm/io.h>
#include <asm/segment.h>

static int lpt=0x378;
static char data=0;

module_param(data,byte,0);
MODULE_PARM_DESC(data,"dado p/porta");

int init_module(void)
{
    printk(KERN_INFO "init\n");
    outb(data,lpt);
    return 0;
}

void cleanup_module(void)
{
    printk(KERN_INFO "cleanup\n");
}

module_init(init_module);
module_exit(cleanup_module);
```

KERNEL DO LINUX

OS MODULOS - EXEMPLOS

```
#include <linux/module.h>
```

```
MODULE_LICENSE("Dual BSD/GPL");
```

```
static int alo_inicio(void) {  
    printk("Alo, Mundo!\n");  
    return 0;  
}
```

```
static void alo_fim(void) {  
    printk("Adeus, Mundo Cruel!\n");  
}
```

```
module_init(alo_inicio);  
module_exit(alo_fim);
```

```
obj-m := alomundo.o
```

```
all:  
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
```

```
clean:  
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

SISTEMA EM TEMPO REAL

- *"sistema capaz de garantir os requisitos de tempo dos processos sob seu controle"*
- Sistemas Hard Real-Time
- Sistemas Soft Real-Time

RTAI

- *Real Time Application Interface (Interface de Aplicação em Tempo Real)*
- fornecendo a capacidade de torná-lo totalmente preemptível
- *Módulos instalados do RTAI:*

```
root@ibk-VirtualBox:/lib/modules/4.4.43-rtai/build/lib# lsmod
Module                Size  Used by
rtai_shm               24576  0
rtai_netrpc            36864  0
rtai_msg               40960  0
rtai_mbx               24576  1 rtai_netrpc
rtai_sem               40960  2 rtai_mbx,rtai_netrpc
rtai_fifos             49152  0
rtai_sched             122880 6 rtai_mbx,rtai_msg,rtai_sem,rtai_shm,rtai_fifos,rtai_netrpc
rtai_hal               491520 7 rtai_mbx,rtai_msg,rtai_sem,rtai_shm,rtai_fifos,rtai_sched,rtai_netrpc
crct10dif_pclmul       16384  0
```