

# Laboratório de Sistemas Operacionais

## Módulos e Processos no Linux

Gustavo M. D. Vieira

Campus Sorocaba – UFSCar  
`gdvieira@ufscar.br`

2010/2

# Módulos no Linux

## Módulos

- Módulos são um mecanismo que aumentam a flexibilidade de núcleos monolíticos, permitindo a carga e descarga em tempo de execução de partes do núcleo.
- O Linux permite que praticamente toda a sua funcionalidade esteja organizada em módulos dinâmicos.

## Ferramentas de Gerência de Módulos

- `lsmod`: Lista os módulos atualmente carregados.
- `insmod`: Carrega (insere) um módulo (básico).
- `rmmod`: Remove um módulo (básico).
- `modinfo`: Exibe informações sobre um módulo.

# Módulos no Linux

## Estrutura de um Módulo

- Um módulo é objeto C com uma estrutura pré-definida (convenção).
- Exemplo:

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

int init_module(void) {
    ...
    return 0;
}

void cleanup_module(void) {
    ...
}

MODULE_LICENSE("GPL");
```

# Módulos no Linux

## Compilação de um Módulo

- Os módulos são mantidos na árvore do Linux. Não são componentes externos.
- Porém, é possível compilar módulos fora do núcleo.
- Para isto precisamos ter os fontes do núcleo sendo usado disponível. Por exemplo: `/src/linux-2.6`.
- Basta então criar um makefile listando os módulos a serem compilados:

```
obj-m += hello.o
```

e chamá-lo com:

```
$ make -C /src/linux-2.6 SUBDIRS=$PWD modules
```

# Módulos no Linux

## Comunicação com Processos

- Um módulo executa como um código no núcleo. Logo um módulo, ao ser carregado, **é o núcleo**.
- Como programas de aplicação falam com o núcleo?

# Módulos no Linux

## Comunicação com Processos

- Um módulo executa como um código no núcleo. Logo um módulo, ao ser carregado, **é o núcleo**.
- Como programas de aplicação falam com o núcleo?
- A forma padrão são chamadas de sistema, mas elas são complicadas de se implementar.
- Vários sistemas Unix usam um meio alternativo: sistemas de arquivos “virtuais”.

`/proc`: Fornece informações sobre processos e outras partes do sistema.

`/sys`: Forma moderna de acessar partes do sistema.

# Módulos no Linux

## Sistemas de Arquivos Virtuais

- Posso ler e escrever nestes arquivos como se fossem reais.
- Posso usar qualquer ferramenta.
- Exemplo:

```
$ cat /proc/cpuinfo
```

ou

```
$ echo 1 > /proc/sys/net/ipv4/ip_forward
```

## Módulo Exemplo

- Ver módulo exemplo: `hello.c`.
- Outros detalhes: `printk`.

# Processos

## task\_struct

- Os dados de um processo ficam no registro `task_struct`.
- Executando no núcleo, é possível conseguir um ponteiro para esta estrutura com a macro `current`.
- A partir daí podemos navegar na estrutura. Exemplo:  
`current->parent` // Processo pai
- Todos os processos estão organizados em uma lista circular, `current->tasks`.
- Para navegar por esta lista, use as macros:  
`next_task(current)` // Próximo processo na lista  
`for_each_process(p)` // Varre todos os processos
- Mais informações sobre como navegar esta lista em <http://linuxgazette.net/133/saha.html>



# Processos

## `task_struct->cred`

- Dentro de `task_struct` podemos encontrar a estrutura `cred` que guarda as credenciais de um processo.
- Entre outras coisas, o id do usuário e do grupo que criou o processo e define o que ele pode fazer.
- O id 0 é o root e pode fazer tudo!
- Mais informações sobre credenciais em “`man credentials`”.
- A estrutura `cred` é protegida. Para acessá-la, deve-se fazer:

```
cred = get_task_cred(current->parent);  
// Faz alteração  
put_cred(cred);
```