

Arquitetura do Sistema Operativo

1. Organização do Sistema Operativo

1.1. Organização Típica

1.2. Base de segurança do sistema

1.3. Syscalls

1.4. Kernel Monolítico

1.5. Kernel em Camadas

1.6. Micro Kernel

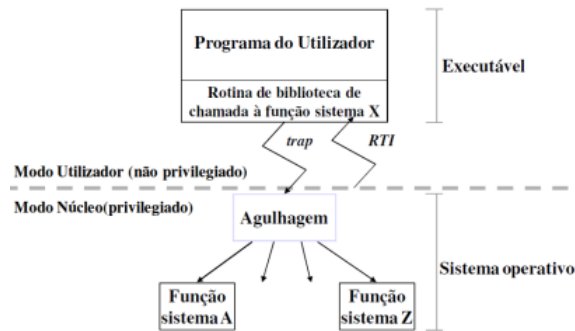
1. Organização do Sistema Operativo

1.1. Organização Típica

- Um SO divide-se em:
 - *kernel* (dividido em módulos);
 - *syscalls* (usadas pelas aplicações);
 - processos de sistema.

1.2. Base de segurança do sistema

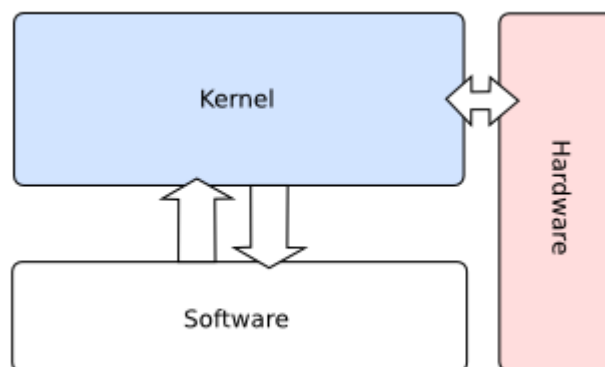
- Durante o seu tempo de vida, um processo (*aplicação*) executa-se no modo kernel e no modo de utilizador (**kernel mode** e **user mode**, respetivamente);
- No **user mode**, o acesso a certos recursos é limitado;
- No **kernel mode**, o processo tem acesso não restrito a recursos do *hardware*.
- Um processo a executar-se no **user mode** pode executar código do **kernel mode** ao executar *syscalls*;



1.3. Syscalls

1.4. Kernel Monolítico

- Todo o SO opera no *kernel mode*;
- *syscalls* implementam todos os serviços do SO:
 - gestor de procesos;
 - concorrência;
 - gestão de memória.
- *Drivers* de periféricos são adicionados ao *kernel* como módulos.
- A quantidade de código executado no espaço do *kernel* torna-o mais propício a erros fatais.

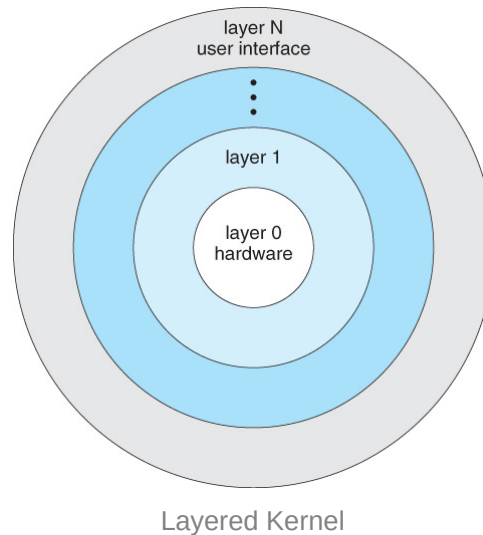


Monolithic Kernel

1.5. Kernel em Camadas

- Cada camada usa serviços da camada anterior;
 - Simplifica depuração;

- Menos eficiente que outros tipos de arquitetura:
 - quando um programa executa uma operação de I/O, é executada um *syscall* que é apanhada (*trapped*) na camada I/O, o que gera uma cadeia de chamadas a funções presentes em camadas precedentes:
 - **maior latência**, pois cada chamada adiciona *overhead* à *syscall*.



1.6. Micro Kernel

- Contém apenas o essencial do SO:
 - Gestão de fluxos de execução (*threads*);
 - Gestão de memória;
 - Comunicação entre processos;
 - Gestão de interrupções.
- Servidores:
 - executam processos independentes para as restantes funcionalidades:
 - gestão de processos;
 - memória virtual;
 - sistema de ficheiros.