

MC504/MC514 - Sistemas Operacionais

Introdução

Islene Calciolari Garcia

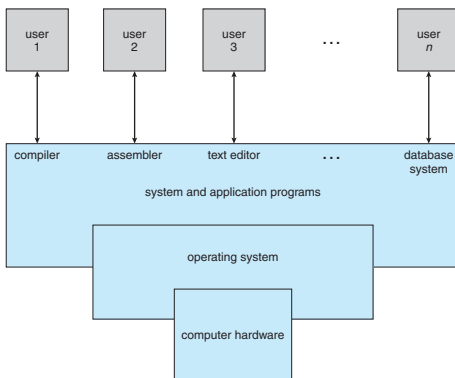
Instituto de Computação - Unicamp

Primeiro Semestre de 2016

Sumário

- 1 Introdução
- 2 Ementa
- 3 Critério de Avaliação
- 4 Espaço de endereçamento

Sistema operacional



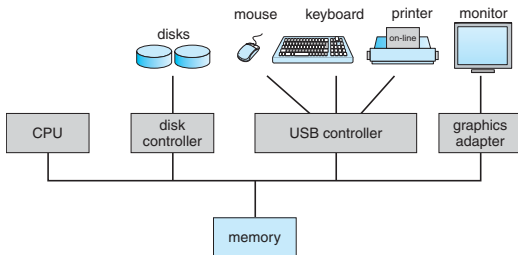
Silberschatz et al.: Figura 1.01

O sistema operacional isola o hardware das camadas superiores em um sistema computacional

Sistema operacional

- *Gerenciador de recursos*

fornece uma alocação controlada de processadores, memória e dispositivos de entrada/saída



Silberrschatz et al.: Figura 1.02

Ementa

Conceito de threads e processos: concorrências, regiões críticas, escalonamento. Conceitos de espaços de endereçamento e de gerenciamento de memória virtual, paginação, segmentação. Sistemas de arquivos: hierarquia, proteção, organização, segurança. Gerenciamento de Entrada/Saída. Estudos de caso.

Critério de Avaliação

Provas teóricas

Serão realizadas três provas escritas P_1 , P_2 e P_3 , sem consulta.

Datas

P_1 : 29 de março

P_2 : 28 de abril

P_3 : 09 de junho

Média

$$M_{prova} = \frac{2 * P_1 + 3 * P_2 + 4 * P_3}{9}$$

Critério de Avaliação

Seminário

Temas possíveis:

- Tópico avançado em Sistemas Operacionais
- Estudos de caso
- Projeto prático com Kernel Linux

Os seminários serão apresentados em grupos de no **máximo quatro** alunos. Todos os integrantes deverão apresentar conteúdo. A avaliação será individual.

MC504: Critério de Avaliação

Média parcial e média final

$$M_{parcial} = 7 * M_{prova} + 3 * M_{semi}$$

$M_{prova} \geq 5.0$	$M_{parcial} \geq 5.0$	$M_{final} = M_{parcial}$
	$M_{parcial} < 5.0$	$M_{final} = (Exame + M_{parcial})/2$
$M_{prova} < 5.0$	$M_{parcial} \geq 7.0$	$M_{final} = M_{parcial}$
	$M_{parcial} < 5.0$	$M_{final} = (Exame + M_{parcial})/2$

Data do exame: 12 de julho.

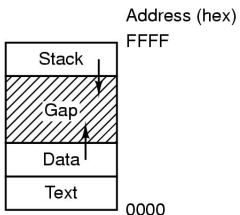
Em caso de fraude em qualquer atividade avaliativa ao longo da disciplina todos os envolvidos ficarão com média final igual a zero.

Principais Referências

- Modern Operating Systems, Andrew Tanenbaum
- A. Silberschatz, P. Galvin, G. Gagne, Operating Systems Concepts
- W. Stallings, Operating Systems: Internals and Design Principles
- Understanding the Linux Kernel, Daniel P. Bovet e Marco Cesati
- The Little Book of Semaphores, Allen B. Downey

Processo

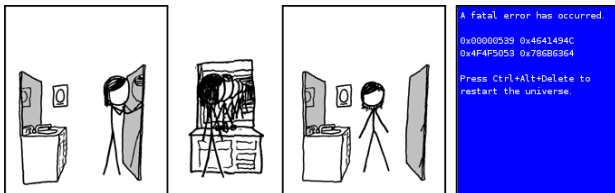
- Programa em execução
- Espaço de endereçamento



Tanenbaum: Figura 1.20

- Veja os códigos: `ender.c` e `ender-malloc.c`
- Endereços reais ou virtuais?

Recursão infinita



<http://techttime.getharvest.com/blog/segmentation-faults/>

```
void f() {  
    printf("Recursão infinita...");  
    f();  
}
```

- Por que o programa gera um erro de execução?
- Veja o código `rec.c`

Borda da área de dados

Códigos desse tipo podem causar erro de execução?

```
void f() {  
    int *v = malloc(4*sizeof(int));  
    v[10] = 5;
```

- Veja o código erro-malloc.c
- Explore a borda da área de dados com o código sbrk.c
- O que acontece quando um bloco grande de dados é alocado?
- O que acontece quando solicitamos muitas alocações? Veja loop-malloc

Pilha de execução

Códigos desse tipo podem causar erro de execução?

```
void f() {  
    int v[5];  
    int i;  
  
    for (i = 0; i < 10; i++)  
        v[i] = i;  
}
```

- Veja o código pilha.c

Vamos examinar a pilha de execução

- Componentes do frame (não necessariamente nesta ordem):
 - valor de retorno
 - endereço de retorno
 - registradores
 - argumentos para a função
 - variáveis locais da função
- Veja os códigos `pilha2.c` e `pilha3.c`
- No gdb execute comandos do tipo:

```
(gdb) p (int[10]) *v
```

```
(gdb) p (int[10]) *(v-2)
```

```
(gdb) set *(v-2) = ...
```

E na próxima aula...

- Programação multithread (vários fluxos de execução)
- Espaço de endereçamento pode ser compartilhado