

MC504 - Sistemas Operacionais

Introdução

Islene Calciolari Garcia

Instituto de Computação - Unicamp

Primeiro Semestre de 2014

Sumário

- 1 Introdução
- 2 Ementa
- 3 Critério de Avaliação
- 4 Processo

Sistema operacional

Aplicações		
Shell	Compiladores	Editores
<i>Sistema operacional</i>		
Hardware		

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
- *Máquina estendida*
oferece uma máquina virtual mais simples de programar do que o hardware
- *Precisamos de muito mais do que o kernel*
(exemplo: *gnu coreutils*)

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 casos.

Critério de Avaliação

Provas teóricas

Serão realizadas duas provas escritas P_1 e P_2 , ambas sem consulta.

Datas

P_1 : 15 de abril
 P_2 : 10 de junho

Média

$$M_{prova} = \frac{P_1 + P_2}{2}$$

Critério de Avaliação

Projetos práticos

Principais temas:

- Programação multithread
- Kernel Linux

Os projetos práticos poderão ser desenvolvidos individualmente ou em grupos de no **máximo três** alunos. A avaliação do código fonte projetos poderá levar em conta os seguintes itens: (i) correção, (ii) clareza e (iii) eficiência: tempo e espaço.

Critério de Avaliação

Projetos práticos

Cada aluno ou grupo deverá criar um repositório no github ou bitbucket, onde deverá ficar disponível para *download* o código desenvolvido e a documentação associada. Os projetos deverão ser executados, explicados e apresentados em sala de aula.

Seja criativo! Deixe seu projeto como contribuição para os alunos dos próximos semestres.

Critério de Avaliação

Média das provas e projetos (média parcial):

$$M_{prova} \geq 5 \text{ e } M_{proj} \geq 3$$

$$M_{parcial} = \frac{3 * M_{prova} + M_{proj}}{4}$$

$$M_{prova} < 5 \text{ ou } M_{proj} < 3$$

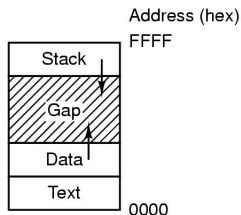
$$M_{parcial} = \min(M_{prova}, M_{proj})$$

Principais Referências

- Modern Operating Systems, Andrew Tanenbaum
- 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

```
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`
- Explore a borda da área de dados com o código `sbrk.c`

E na próxima aula...

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