

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Sistemas Operacionais
Última alteração: 10 de junho de 2013

- **Professor:** Dorgival Olavo Guedes Neto (Sala 4022)
- Local: ICEx, sala a definir
- Horário: segundas e quartas, 17:00 às 19:00
- **Estagiário em docência:** a definir
- **Dúvidas:** Utilize o Moodle.
Não envie mail diretamente para o professor, exceto para o caso de querer discutir código!

Informações gerais

Objetivo

Abordar os princípios básicos da área de Sistemas Operacionais com uma ênfase na noção de sistemas de computação, usando uma visão voltada para algoritmos e interfaces.

Ao final do curso o aluno deve ser capaz de identificar os principais elementos de um Sistema Operacional, analisar compromissos no desenvolvimento de soluções para problemas como contenção de recursos, escalonamento, gerência de memória e uso de armazenamento secundário, bem como descrever os princípios de operação de Sistemas Operacionais com relação a esses fatores.

Os conceitos da disciplina serão exercitados através de exercícios práticos voltados para a verificação de características específicas de sistemas operacionais e do desenvolvimento de versões a nível de usuário de alguns dos sistemas discutidos.

Ementa

Conceitos de sistemas operacionais, redes de computadores, protocolos e serviços de comunicação. Arquitetura de redes de computadores. Camadas inferiores do modelo de referência de ISO: física, enlace, redes e transporte. Redes locais, interligação de redes; especificação de protocolos.

Programa

O curso cobrirá a maior parte do livro texto adotado, com ênfase maior nos princípios básicos da área. Questões mais avançadas e de interesse mais recente, como segurança e aplicações multimídia, serão tratadas de forma mais superficial. Os alunos são encorajados a ler no livro as seções que não serão discutidas em sala de aula.

- Processos:
 - comunicação,
 - sincronização,
 - escalonamento;
- Gerência de memória:
 - *swapping*,
 - paginação,
 - segmentação,
 - memória virtual;
- Sistema de arquivos;
- Gerência de entrada/saída (E/S).

Calendário

(Calendário tentativo, sujeito a alterações)

Aula	Data	Assunto	Obs.
1	04/03	Introdução. Motivação, histórico.	
2	06/03	Conceitos e funcionalidades.	
3	11/03	Funcionalidades, estrutura, princípios básicos	
4	13/03	Geração e boot. Chamadas de sistema.	TP1->
5	18/03	Processos: introdução. Processos no Unix	
6	20/03	Processos: cooperação.	
7	25/03	<i>Threads</i> . Escalonamento de processos.	
—	27/03	Aula cancelada	
8	01/04	Escalonamento.	
9	03/04	Escalonamento. Sincronização entre processos.	
10	08/04	Primeira prova: capítulos 1 a 5.	
11	10/04	Algoritmos e primitivas de sincronização.	<-TP1
12	15/04	Problemas de sincronização	
13	17/04	Problemas clássicos de sincronização	
14	22/04	Problemas de sincronização. <i>Deadlocks</i> .	
15	24/04	<i>Deadlocks</i>	
16	29/04	Gerência de memória. Revisão: sincronização.	
—	01/05	Feriado: Dia do Trabalho	TP2->
17	06/05	Segunda prova: capítulos 5, 6 e 7.	(SBRC)
—	08/05	Aula cancelada	(SBRC)
18	13/05	Gerenciamento de memória. Segmentação	
19	15/05	Gerenciamento de memória. Paginação.	TP3->
20	20/05	Memória virtual: princípios	
21	22/05	Memória virtual: funcionamento	
22	27/05	Memória virtual: aspectos de implementação	<-TP2
23	29/05	Interface do Sistema de Arquivos	
24	03/06	Terceira prova: capítulos 8 e 9.	
25	05/06	Implementação do Sistema de Arquivos	
26	10/06	Implementação do Sistema de Arquivos	
27	12/06	Estrutura do armazenamento secundário	
—	17/06	Copa das Confederações no Mineirão	
—	19/06	Copa das Confederações: jogo do Brasil	
28	24/06	Proteção e segurança	
—	26/06	Copa das Confederações no Mineirão	<-TP3
29	01/07	Proteção e segurança. Observações finais	
30	03/07	Quarta prova: capítulos 10, 11, 12, 14 e 15	

Avaliação

Distribuição dos pontos

- **Provas:** quatro, *aproximadamente* 51 % da nota final.
Não há prova suplementar, exceto para casos de ausência justificada.
- **Trabalhos Práticos:** três, *aproximadamente* 45 % da nota final

- **Participação em sala e no moodle:** aproximadamente 4% da nota final.

Provas

Teremos quatro provas, com datas e matéria a serem definidas ao longo do curso. (Consulte o calendário para as datas corretas.)

Conteúdo das provas (**a confirmar**):

- Primeira prova: introdução, estrutura do S.O., processos e threads, escalonamento.
- Segunda prova: escalonamento, sincronização de processos e *deadlocks*.
- Terceira prova: gerência de memória e memória virtual.
- Quarta prova: sistemas de arquivos (interface e implementação), discos, proteção e segurança.

Obviamente, cada prova pode ser composta sobre os conceitos abordados anteriormente em outras provas.

Exercícios

Serão disponibilizadas (no moodle) listas de exercícios sobre o assunto de cada capítulo, com soluções. Apesar de não valerem pontos, essas listas visam exercitar os principais conceitos associados e servem como preparação para as provas.

Trabalhos

- **TP1:** Um interpretador de comandos do usuário (*shell*)
(**Trabalho individual ou em dupla**). 15 pontos.
- **TP2:** Um problema simples de sincronização usando threads
(**Trabalho individual ou em dupla**). 12 pontos.
- **TP3:** Um simulador de memória virtual
(**Trabalho individual ou em dupla**). 18 pontos.

As práticas devem ser desenvolvidas no ambiente Linux usando as linguagens ANSI C/ANSI C++, sem bibliotecas adicionais além das bibliotecas padrão de cada linguagem (e do Linux, obviamente). Uma boa opção, para aqueles que têm acesso a um computador em casa, é instalar uma versão de Linux. Isso pode ser feito com uma instalação como a Ubuntu a partir de um CD, ou você pode instalar um ambiente virtual como o Virtual Box e executar uma máquina virtual Linux. No caso do Ubuntu, certifique-se que os seguintes pacotes sejam instalados: `gcc`, `gcc-doc`, `glibc-doc`, `manpages`, `manpages-dev` e `ddd`. Isso pode ser feito com o comando:

```
sudo apt-get install gcc gcc-doc glibc-doc manpages manpages-dev ddd
```

0.1 Sobre prazos e formato de entrega:

Todos os trabalhos devem ser entregues eletronicamente, como um arquivo `.zip` ou `.tar.gz`, contendo todo o material relacionado ao trabalho em questão — não inclua arquivos objeto (`.o`) nem arquivos executáveis; um relatório deve fazer parte da entrega, detalhando aspectos principais do trabalho executado. **O relatório e todos os arquivos fonte devem conter o(s) nome(s) dos autores. O relatório deve conter uma seção descrevendo qual foi a participação de cada aluno no desenvolvimento do trabalho.**

A entrega eletrônica deve ser feita até o **meio-dia** do dia de entrega do trabalho.

Trabalhos entregues depois do prazo serão penalizados em 10 % ao dia, começando com trabalhos entregues no dia indicado, mas após o prazo. Trabalhos com mais de 4 dias de atraso (50% de penalização) não serão aceitos.

Para evitar problemas, comece a fazer cada trabalho assim que ele for definido. O prazo planejado para cada entrega prevê trabalho contínuo durante o período; deixar para fazer tudo nos últimos dias **não vai dar certo**, acreditem.

Parte da avaliação dos trabalhos poderá ser feita na forma de questões escritas nas provas teóricas e/ou por entrevistas.

Confira as datas dos trabalhos no calendário do curso

Referências e links úteis

Livro-Texto

Fundamentos de Sistemas Operacionais, oitava edição, Silberschatz, Galvin e Gagne, Ed. LTC.

Atenção: há uma outra versão semelhante, porém baseada no uso de Java como ambiente de aplicação. Apesar do conteúdo ser bastante semelhante, a diferença de enfoque pode deixar de fora alguns aspectos importantes.

Bibliografia suplementar

Existem outros bons livros sobre Sistemas Operacionais por aí, mas eles não trazem o material com a profundidade e organização desejadas para este curso, portanto não são recomendados como substituto para o livro texto adotado. Entretanto, alunos em busca de uma outra visão sobre o tema podem querer consultar esses livros eventualmente:

- Sistemas Operacionais Modernos, Tanenbaum
A outra referência mais importante na área.
- Projeto e Implementação de Sistemas Operacionais, Tanenbaum
O primeiro livro com enfoque mais moderno na área. Foca toda a discussão sobre o código fonte de um S.O. desenvolvido para o livro, o Minix.
- Stevens, W.R., *Advanced Unix Programming*, uma boa para se conferir todos os detalhes das interfaces de programação de processos, E/S e memória do Unix/Linux.
É basicamente uma versão estendida do livro *Unix Network Programming, 2nd ed.* do mesmo autor, pela Prentice Hall (não sei se há tradução). Esse livro é tão famoso entre os programadores Unix que já apareceu até no cinema!

Links úteis

Assuntos variados

Alguns links úteis que trazem mais detalhes sobre tópicos específicos discutidos durante o curso.

- **C:** o tutorial “Programming in C” tem muitas informações importantes, como o uso de sinais, etc.
- **Codificação:** detalhes de diferentes técnicas de codificação.

Artigos

- **Hints for Computer System Design** (ACM SOSP 1983)