

SISTEMAS OPERACIONAIS I 2015

Prof. Carlos Eduardo de B. Paes

Departamento de Computação
Pontifícia Universidade Católica de São Paulo

Objetivos do Curso

- Apresentar conceitos necessários para o gerenciamento eficiente dos recursos de hardware disponíveis em uma plataforma computacional.
- Apresentar ao estudante conceitos fundamentais de Sistemas Operacionais
- Fornecer o embasamento teórico para que o aluno possa assimilar novas tecnologias na área de computação
- Promover a teoria e prática de programação concorrente

Conteúdo Programático (resumo)

- Conceituação de Sistemas Operacionais
- Evolução Histórica dos Sistemas Operacionais
- Classificação (Taxonomia) dos Sistemas Operacionais
- Arquitetura dos SOs
- Conceito de Processo
- Threads
- Escalonamento de CPU
- Mecanismos de Comunicação e Sincronização (Inter-Process Communication – IPC)
- Deadlock

Avaliação

- Provas P1, P2 e PS
- Notas A1 e A2 → Atividades bimestrais
 - Exercícios em sala de aula (Teoria)
 - Experimentos em laboratório (SO1 Prática)
- Nota A:
 - Projeto de Implementação (SO1 Prática)
 - Monografia e Seminário

Avaliação

- Cálculo da média final:
 - $MF = [(N1 + N2) / 2] \times (0,8 + 0,04 \times A)$
 - Onde temos:
 - $N1 = (3 \times P1 + A1) / 4$ e $N2 = (3 \times P2 + A2) / 4$
 - P1 e P2: provas semestrais
 - A1 e A2: atividades semestrais (em sala de aula ou em laboratório)
 - A: Projeto de implementação e/ou seminário
- Datas das provas:
 - P1: 03/04
 - P2: 29/05
 - PS: 05/06

Pré-Requisitos

- Conhecimento da linguagem de programação C/C++, algoritmos e estrutura de dados (DA+IC+LP+ED)
- Conhecimento de organização de computadores (OC) e Arquitetura de Computadores (ARQ)
- Conhecimento de linguagem de montagem de algum processador moderno (OC)
- Conhecimento de mecanismos de interrupção (OC)

Procedimentos de Ensino

- Aulas expositivas em sala de aula com participação do aluno em pelo menos 75% das aulas;
- Atividades individuais ou em grupo realizadas em sala ou laboratório
- Listas de exercícios para as provas
- Leitura recomendada de livros, revistas, artigos e etc..
- Trabalhos de implementação

Laboratório

- Programação na linguagem C
- Sistema Operacional Linux (*Unix like*)
- Utilização de chamadas ao Sistema Operacionais
- Prática de programação concorrente
- Solução de problemas relacionados a sincronização e comunicação
- IPC e Threads (Java e pthreads)

Ambiente de Apoio

- *PUC-SP Moodle*
- *Chave de acesso:* **ausm28**
- *Material da disciplina*
- *Fórum de discussão*
- *Notícias*
- *Etc...*

Conduta Acadêmica

- Durante as aulas exige-se:
 - Silêncio;
 - Concentração;
 - Respeito aos colegas e ao professor;
 - **Telefones celulares desligados;**
 - Chegar no horário!!
- Evitar saídas constantes da sala de aula!!

Conduta Acadêmica

- **Presença**: A presença em sala de aula será verificada através de chamada no início das aulas. Informe o professor (se possível antecipadamente) no caso de ausências inevitáveis.
- **Grupos de discussão**: São permitidos (e mesmo incentivados) o estudo e a discussão de exercícios em grupos, porém cada aluno deve apresentar suas soluções individualmente para qualquer material a receber nota, exceto para exercícios desenvolvidos em sala

Conduta Acadêmica

- **Trabalhos**: Não é permitido o compartilhamento de trechos de texto ou qualquer material relativo ao desenvolvimento dos trabalhos entre os alunos, ou o reaproveitamento de material desenvolvido em anos anteriores.

Conduta Acadêmica

- **Código de integridade**: Cada aluno deve conduzir seu trabalho acadêmico com honestidade e integridade. Exemplos de conduta imprópria incluem:
 - **Plágio**: Uso de idéias e informações de outra pessoa sem reconhecimento da fonte.
 - **Uso de trabalho de outros**: Submeter um trabalho ou prova preparado por outra pessoa, ou preparar e oferecer um trabalho ou prova para uso por outros.
 - **Entrega de material usado anteriormente**: Submeter, sem consentimento prévio do professor, trabalho (ou parte) que já tenha sido submetido anteriormente em qualquer instituição.
 - **Alteração de notas**: Falsificar, alterar ou acessar sem prévia autorização notas e outros dados acadêmicos de qualquer aluno.

Recomendações

- Recomenda-se:
 - Obter o material de apoio antes das aulas
 - Fazer anotações;
 - Complementar os estudos fora do horário de aula
 - Participar em sala de aula
 - Esclarecer dúvidas com o professor durante ou fora do horário de aula

Bibliografia de Referência

- SILBERSCHATZ, A. et al., *Applied Operating System Concepts*. First Edition, John Wiley & Sons, Inc, 2000.
- TANENBAUM, D and WOODHULL, A., *Operating System: Design and Implementation*. Second Edition, Prentice Hall, 1997.

Bibliografia Complementar

- GALVIN, S., *Operating System Concepts*. Fifth Edition, Addison-Wesley, 1998.
- NUTT, G., *Operating System: A Modern Perspective*. Addison-Wesley, 1997.
- STEVES, W. R., *Unix Networking Programming: Interprocess Communication*. Second Edition, Prentice Hall, 1998.

Bibliografia Complementar

- MILENKOVIC, M., Operating Systems: Concepts and Designs, McGraw Hill, 1987.
- MADNICK, DOHOUAN, Operating Systems, McGraw Hill Ed., 1974.
- KELLER, L. S., Operating Systems: Communicating and Controlling the Computer, Pretince-Hall Inc, 1988.

Bibliografia Complementar

- MAEKAWA, OLDEHOEFT, Operating Systems: Advanced Concepts, Benjamin Cummings-Co., 1987.
- MACHADO, F.B., MAIA, L. P., Introdução a Arquitetura de Sistemas Operacionais, LTC - Livros Técnicos e Científicos Ed. Ltda, 1992.
- DEITEL, H. M.: An Introduction to Operating Systems, Addison Wesley, 1984.

Bibliografia Complementar

- KIRNER, C., MENDES, S.B.T., Sistemas Operacionais Distribuídos, Editor Campus, 1988.
- BACH, M. J., The Design Of The Unix Operating System, Prentice Hall, Inc, 1990.

Bibliografia Complementar Linguagem C

- KERNIGHAN, B. W., RITCHIE, D. C a Linguagem de Programação ANSI. Editora Campus. 1990.

Motivações

- Conceitos de SOs são relevantes em toda ciência da computação.
- Um bom entendimento dos conceitos de SOs fornece uma excelente base para a construção de sistemas de software complexos

Motivações

- Entender o SO é fundamental para um entendimento profundo de um sistema de computação (mesmo que você nunca vá escrever um SO).