

Plano de Ensino de Disciplina

Ciência da Computação

2015

Disciplina: SO1 - Sistemas Operacionais I
Código: **Série:** 3º Semestre
Carga Horária Semanal: 4 aulas
Carga Horária Anual: 68 aulas
Coordenador: Carlos Eduardo de Barros Paes
Professores: Carlos Eduardo de Barros Paes

EMENTA

Conceituação de sistemas operacionais. Evolução desses sistemas. Estrutura do Sistema Operacional. Processos: estados, contexto, gerenciamento pelo kernel, escalonamento, sinais e interrupções. Processos Leves (Threads). Escalonamento de CPU. Comunicação e Sincronização entre processos. Princípios e Prática de Programação Concorrente. Deadlock.

DESCRIÇÃO

A disciplina tem por objetivo apresentar os conceitos necessários para o gerenciamento eficiente dos recursos de hardware disponíveis em uma plataforma computacional, apresentar ao estudante conceitos fundamentais de sistemas operacionais para computadores, bem como aspectos relacionados aos conceitos em sistemas operacionais de importância no mercado e capacitar o estudante a compreender o funcionamento dos sistemas operacionais, bem como fornecer embasamento teórico para que possa assimilar novas tecnologias na área.

OBJETIVOS

Gerais

Proporcionar capacidades e habilidades para que o aluno:

- Conheça e compreenda os mecanismos e as políticas para o compartilhamento dos recursos computacionais e as formas de uso desses recursos por meio do Sistema Operacionais.
- Aplicar esses conhecimentos no processo de desenvolvimento de sistemas computacionais..

Específicos

Conduzir gradativamente o aluno, ao longo do curso, a:

- Conhecer as necessidades que incentivaram o desenvolvimento de sistemas básicos para controle e compartilhamento de recursos;
- Compreender os problemas que surgem ao permitir o compartilhamento de recursos limitados.
- Compreender as teorias criadas para controle desse compartilhamento;
- Aplicar essas teorias de compartilhamento de recursos na organização dos sistemas básicos de programas;
- Conhecer as formas de serviços providos por esse sistema básico ao usuário final do computador.

Procedimentos de Ensino

Cada aula consistirá da combinação adequada de:

- Aulas expositivas em sala com participação do aluno em pelo menos 75% das aulas;
- Atividade individual ou em grupo realizadas em sala de aula;
- Série de exercícios para as provas (apenas provas P1 e P2);
- Leitura recomendada (artigos selecionados)
- Experimentos práticos em laboratório
- Trabalhos de Implementação;
- Seminários: Quando tratar-se da apresentação de seminários, o trabalho deverá ser apresentado na forma escrita e posteriormente apresentado para a turma como um seminário.

Instrumentos e Critérios de Avaliação

Para ser aprovado em qualquer disciplina do curso o aluno deverá atingir pelo menos 75% de presença em 34 semanas de aula e média final igual ou superior a 5.0 (cinco).

A fórmula vigente para o cálculo da **Média Final (MF)**, conforme aprovada pelo Departamento de Ciência da Computação e pelo Conselho Departamental da Faculdade de Matemática, Física e Tecnologia (FMFT), é dada por:

$$MF = (N1 + N2) / 2 \cdot (0.8 + 0.04 A) \text{ com } Ni = (a \cdot Pi + b \cdot Ai) / (a + b) \text{ com}$$

- $a, b \in \{1, 2, 3\}$
- Pi : nota da Prova do semestre i ($i: 1, 2$)
- Ai : nota de Atividades do semestre i ($i: 1, 2$)
- A refere-se a atividade extra, que pode ser constituída por seminários, projetos, trabalhos de pesquisa, trabalhos de campo, etc., de acordo com a especificação da disciplina.

Na disciplina em pauta (SO12), a **Média Final (MF)**, é obtida fazendo-se $a = 3$ $b = 1$ e $A =$ formulada a partir de um projeto de implementação desenvolvido em grupo no decorrer do segundo semestre.

Com isso, tem-se:

$$MF = \frac{N1 + N2}{2} \times (0,8 + 0,04 \times A) \text{ em que } N1 = \frac{3 \times P1 + A1}{4}, N2 = \frac{3 \times P2 + A2}{4}$$

Recursos Necessários

- Projetor digital e retro-projetor;
- Laboratório com o ambiente UNIX (Linux) e as seguintes ferramentas para elaboração de experimentos e projetos de implementação: Compilador C ou C++ (ex: gcc), Compilador Java (SDK da SUN última versão) e Editor de texto (ex: pico, mc e etc)
- Quadro de giz;

Conteúdo Programático

Conceituação de sistemas operacionais

Histórico e tipos de sistemas operacionais.

- Sistemas em lote (batch)
- Sistemas de tempo compartilhado
- Sistemas de computadores pessoais
- Sistemas paralelos
- Sistemas de tempo real
- Sistemas distribuídos

Estrutura e conceitos básicos de sistemas operacionais

- Componentes do sistema
- Serviços de sistemas operacionais
- Chamada ao Sistema (System Calls)
- Programas de sistema
- Estrutura do sistema
- Máquinas virtuais

Gerência de Processos

- Processos (heavyweight process)
- Thread (lightweight process)
- Escalonamento de processos
- Sincronização de processos
- Deadlock e Starvations
- Problemas clássicos de intercomunicação de processos.

Laboratório de Sistemas Operacionais

- Introdução ao Sistema Operacional Linux
- Comandos básicos do Shell
- Linguagem C (ambiente de desenvolvimento)
- Gerência de Processos
- Sinais no Unix
- Comunicação entre Processos e Sincronização: Pipes
- Comunicação entre Processos e Sincronização: FIFOs e Memória Compartilhada
- Comunicação entre Processos e Sincronização: Semáforos e Queues
- Threads (Java e Pthreads)

INSERÇÃO DA DISCIPLINA NO PROJETO DO CURSO

Esta disciplina dará ao aluno um conhecimento sobre os principais conceitos sobre sistemas operacionais, tais como processos, algoritmos de escalonamento, comunicação entre processo e programação concorrente. Além disso, a disciplina apresenta e discute a arquitetura dos principais sistemas operacionais.

Cronograma

Semana	Tópicos	Ref. Bibl.	Aulas
1	Apresentação do curso de Sistemas Operacionais <ul style="list-style-type: none">– Conteúdo programático do curso, sistema de avaliação e bibliografia de referência.– Motivações do estudo de Sistemas Operacionais Laboratório de Sistemas Operacionais <ul style="list-style-type: none">– Introdução ao Linux– Comandos Básicos de Shell– Exercícios: utilização dos comandos básicos de Shell	Plano de Ensino da Disciplina	4
2	Introdução aos Sistemas Operacionais <ul style="list-style-type: none">– Conceituação de Sistemas Operacionais.– Histórico e tipos de sistemas: Sistemas em Lote, Sistemas de Tempo Compartilhado, Computadores Pessoais, Sistemas Paralelos, Sistemas de Tempo Real e Sistemas Distribuídos. Laboratório de Sistemas Operacionais <ul style="list-style-type: none">– Ambiente de desenvolvimento em C– Uso do gcc para compilação de programa em C– Criação de bibliotecas estáticas– Link de bibliotecas em C– Exercícios: implementação de programa simples em C.	Capítulo 1	4
3	Estruturas de Sistemas Operacionais <ul style="list-style-type: none">– Componentes do sistema– Serviços de Sistemas Operacionais– Chamada de sistemas (System Calls)– Programas de sistema– Estrutura do sistema– Máquinas virtuais Laboratório de Sistemas Operacionais <ul style="list-style-type: none">– Linguagem C– Alocação dinâmica em C– Semântica de Ponteiros em C– Exercícios: implementação de alguma estrutura de dados dinâmica em C (Lista, árvore e etc..)	Capítulo 3	4
4	Gerenciamento de Processos <ul style="list-style-type: none">– Conceito de processos.– Escalonamento de processos– Operações nos processos Laboratório de Sistemas Operacionais <ul style="list-style-type: none">– Chamadas ao sistema Linux– Gerenciamento de Arquivos– Exercícios: implementação de leitura e gravação em arquivos texto e binário	Capítulo 4	4

5	Gerenciamento de Processos <ul style="list-style-type: none"> – Operações nos processos Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Chamadas ao sistema Linux – Gerenciamento de Processos (uso de fork, execlp, wait e etc...) – Exercícios: implementação de criação de processos usando fork 	Capítulo 4	4
6	Threads <ul style="list-style-type: none"> – Visão geral – Benefícios – Threads de usuário e de kernel Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Threads em Java – Coordenação de threads (join,start,stop,resume e etc..) – Sincronização em Java (método/bloco sincronizado) 	Capítulo 5	4
7	Threads <ul style="list-style-type: none"> – Modelos de Multithreading – Estudos de Caso: Threads no sistema Solaris e na Linguagem Java Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Locks – ReadWriteLocks – Serviço de execução de tarefas em Java – Exercícios: Implementação de um escalonador de tarefas em Java usando alguns política de escalonamento 	Capítulo 5	4
8	Escalonamento de Processos <ul style="list-style-type: none"> – Conceitos básicos – Critérios de escalonamento – Algoritmos de escalonamento Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Simulador de Sistemas Operacionais SOSim 	Capítulo 6	4
9	Escalonamento de Processos <ul style="list-style-type: none"> – Escalonamento com múltiplos processadores – Escalonamento de tempo real – Escalonamento de threads – Avaliação de algoritmos Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Simulador de Sistemas Operacionais SOSim 	Capítulo 6	4
10	Prova P1		4
11	Comunicação e Sincronização de Processos <ul style="list-style-type: none"> – Processos Cooperativos Comunicação entre Processos – Programação Concorrente – Fundamentos de Sincronização – Problema da região crítica – Soluções para duas tarefas – Hardware de sincronização Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Comunicação de processos – Pipes – Exercícios: implementação de comunicação entre processos usando pipes 	Capítulo 4 e 7	4

12	Comunicação e Sincronização de Processos <ul style="list-style-type: none"> – Semáforos – Monitores Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Comunicação de processos – FIFO e Shared Memory – Exercícios: implementação de comunicação entre processos usando FIFO e memória compartilhada 	Capítulo 4 e 7	4
13	Comunicação e Sincronização de Processos <ul style="list-style-type: none"> – Problemas Clássicos de IPC Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Sincronização de processos – Semáforos e Mutex (padrão System V) – Exercícios: implementação de semáforos para sincronizar processos 	Capítulo 4 e 7	4
14	Deadlock <ul style="list-style-type: none"> – Modelo de sistema – Caracterização de deadlocks – Métodos para tratar de deadlock Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Problemas clássicos de IPC – Produtor e Consumidor – Exercício: implementação do problema do produtor e consumidor usando semáforos e memória compartilhada 	Capítulo 8	4
15	Deadlock <ul style="list-style-type: none"> – Prevenção de deadlock – Impedimento de deadlock – Detecção de deadlock – Recuperação de um deadlock Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Problemas clássicos de IPC – Leitores e Escritores – Exercício: implementação do problema dos leitores e escritores usando semáforos e memória compartilhada 	Capítulo 8	4
16	Prova P2 Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Problemas clássicos de IPC – Barbeiro Dorminhoco – Exercício: implementação do problema do barbeiro dorminhoco usando semáforos e memória compartilhada 	Capítulo 8	4
17	Correção e vista da Prova P2 Problema dos Filósofos Glutões Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Exercício: implementação do problema dos filósofos glutões usando semáforos. 		
18	Prova PS Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Exercício: implementação do problema dos filósofos glutões usando semáforos. 		4
19	Correção e vista da Prova PS Laboratório de Sistemas Operacionais <ul style="list-style-type: none"> – Exercício: implementação do problema dos filósofos glutões usando semáforos. 		

PRÉ-REQUISITOS - TÓPICOS

Essa disciplina possui como pré-requisito os seguintes tópicos:

- Conhecimento de programação em alguma linguagem de alto nível, técnicas de programação e estrutura de dados (LP2+DA)
- Conhecimento básico de organização de computadores (OC)
- Conhecimento de linguagem de montagem de algum processador moderno (OC)
- Conhecimento de mecanismos de interrupção (OC)

PRÉ-REQUISITOS - DISCIPLINAS

Essa disciplina possui como pré-requisito as seguintes disciplinas:

- Laboratório de Programação 1 e 2
- Organização de Computadores
- Desenvolvimento de Algoritmos

BIBLIOGRAFIA

Básica

- SILBERSCHATZ, A. et al., *Operating Systems Concepts with Java*. Eighth Edition, John Wiley & Sons, Inc, 2009.
- TANENBAUM, A., *Modern Operating Systems*. Third Edition, Prentice Hall, 2010.
- MACHADO, F.B., MAIA, L. P., *Introdução a Arquitetura de Sistemas Operacionais*, LTC - Livros Técnicos e Científicos Ed. Ltda, 3ª Edição, 2007.

Complementar

- GALVIN, S., *Operating System Concepts*. Fifth Edition, Addison-Wesley, 1998.
- TANENBAUM, D and WOODHULL, A., *Operating System: Design and Implementation*. Second Edition, Prentice Hall, 1997
- NUTT, G., *Operating System: A Modern Perspective*. Addison-Wesley, 1997.
- STEVES, W. R., *Unix Networking Programming: Interprocess Communication*. Second Edition, Prentice Hall, 1998.
- MILENKOVIC, M., *Operating Systems: Concepts and Designs*, McGraw Hill, 1987, ISBN: 0-07-041920-5.
- MADNICK, DOHOUAN, *Operating Systems*, McGraw Hill Ed., 1974, ISBN: 0-07-039455-5.
- KELLER, L. S., *Operating Systems: Communicating and Controlling the Computer*, Prentice-Hall Inc, 1988, ISBN: 0-13-638040-9.
- MAEKAWA, OLDEHOEFT, *Operating Systems: Advanced Concepts*, Benjamin Cummings-Co., 1987, ISBN: 0-8053-7121-4.
- DEITEL, H. M.: *An Introduction to Operating Systems*, Addison Wesley, 1984.
- 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.