

**Disciplina:** SISTEMAS HARDWARE-SOFTWARE

**Carga Horária Total:** 80

**Período Letivo:** 2019 / 62

**Matriz Curricular:** 5º período

**Ementa:** Representação de informação; Representação de programas em código de máquina; Arquitetura de processadores e otimização de programas; Hierarquia de memória; Exceções e controle de fluxo em sistemas operacionais; Memória virtual; Gerenciamento de E/S; Programação concorrente.

**Objetivos:** Ao final da disciplina o estudante será capaz de:

1. Explicar o ciclo de vida de um objeto executável, desde a sua geração a partir do código fonte, até a execução do código gerenciada pelo sistema operacional;
2. Explicar o papel do sistema operacional na gerência e mediação de recursos, incluindo gerência de processos, memória, e interface com o hardware;
3. Investigar, analisar e formular hipóteses sobre o desempenho de um programa de computador vis-a-vis a interação deste com os elementos fundamentais do hardware
4. Projetar e aperfeiçoar programas de computador para que usem as características do hardware visando atingir um alto desempenho;
5. Desenvolver sistemas de computador que exibem concorrência de execução e de uso de recursos; e
6. Contextualizar tecnologias atuais em relação à evolução de sistemas de computação, e discutir possibilidades futuras de evolução tecnológica nesta área.

**Conteúdo Programático:**

1. Introdução a programação em C
2. Arquitetura de computadores: interação entre CPU, memória e periféricos
3. Representação de dados e aritmética na CPU: inteiros de tamanho fixo, ponto flutuante, ponteiros
4. Engenharia reversa de programas em Assembly: organização de programas na memória, operações aritméticas e ponteiros, estruturas de controle de fluxo e chamada de funções
5. Gerenciamento de memória em sistemas operacionais: memória virtual, cache, alocação de memória, otimização de utilização de cache em programas;
6. Gerenciamento de processos em sistemas operacionais: exceções de hardware, sinais, agendamento, arquivos, Entrada/Saída;
7. Introdução a programação concorrente: threads vs. processos, primitivas de sincronização e controle de deadlocks, concorrência vs. paralelismo.

**Bibliografia Básica**

**Livros:**

1. BRYANT, R. E.; O'HALLARON, D. R. , Computer systems: a programmer's perspective, 3ª ed., Pearson, 2015
2. D. A. PATTERSON; J. L. HENNESSY, Organização e Projetos de Computadores: a Interface Hardware/Software., 4ª ed., Elsevier, 2014
3. KOCHAN, S. G., Programming in C, 4ª ed., Addison-Wesley, 2014

**Artigos:**

KIVITY, A.; COSTA, D.; ENBERG, P.. Optimizing the Operating System for Virtual Machines.. Proceedings of USENIX ATC'14: 2014 USENIX Annual Technical Conference. , p. 61 , 2014.

## **Bibliografia Complementar**

### **Livros:**

1. KERNIGHAN, B. W.; RITCHIE, D. M., The C programming language, 2ª ed., Prentice-Hall, 1988
2. J. H. SALTZER, M. F. KAASHOEK., Principles of Computer System Design: An Introduction, Morgan Kaufmann, 2009
3. STALLINGS, W., Arquitetura e organização de computadores, 8ª ed., Pearson, 2013
4. SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G., Operating System Concepts, 9ª ed., Wiley, 2013
5. TANENBAUM, A. S., Sistemas Operacionais Modernos, 4ª ed., Pearson, 2016

### **Artigos:**

GERBER, F.; MÖSINGER, K.; FURRER, R.. DotCall64: An R package providing an efficient interface to compiled C, C++, and Fortran code supporting long vectors. SoftwareX. , n. 7 , 2018. ; Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352711018300785>. Acesso em: 27 maio 2019.