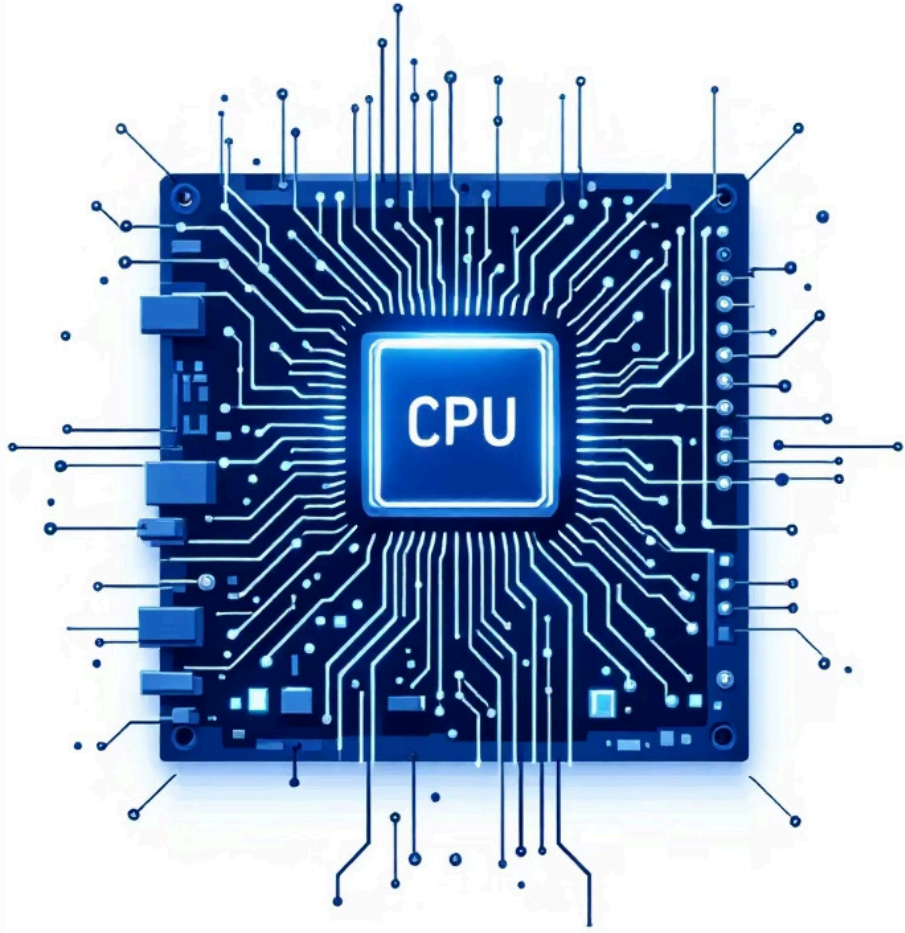
Arquitetura de Computadores

Prof. Cloves Rocha · Engenharia da Computação & Ciência da Computação

Esta disciplina oferece uma visão abrangente dos fundamentos que sustentam o funcionamento dos computadores modernos — desde a organização interna do hardware, passando pela linguagem de máquina e pelo pipeline do processador, até os conceitos de sistemas operacionais, gerenciamento de memória, sistemas de arquivos e segurança. O objetivo é capacitar o estudante a compreender, avaliar e projetar sistemas computacionais com base sólida em teoria e prática.

GRADUAÇÃO | ENGENHARIA & CIÊNCIA DA COMPUTAÇÃO | PROF. CLOVES ROCHA



Ementa e Competências da Disciplina

O que você vai estudar

A disciplina cobre um espectro amplo e integrado de tópicos, indo da camada mais baixa do hardware até os serviços do sistema operacional:

- Modelo e histórico dos sistemas de computação
- Operações aritméticas e lógicas; ULA
- Linguagem de máquina e representação de instruções
- Processador: CPU, clock, registradores, pipeline
- Hierarquia de memória: cache, virtual, paginação
- Periféricos (I/O) e multiprocessadores
- Processos, threads, sincronização e comunicação
- Sistema de arquivos e segurança computacional

Competências Específicas

Ao concluir a disciplina, o estudante será capaz de:

Fundamentos de Hardware

Conhecer profundamente a organização interna do computador — memória, CPU e dispositivos de I/O —, identificando as partes fundamentais e suas respectivas funções no ciclo de execução de instruções.

Avaliação de Sistemas Operacionais

Avaliar criticamente os diferentes sistemas operacionais disponíveis no mercado, com base nas características de gerenciamento de processos, memória, arquivos e segurança, escolhendo a solução mais adequada para cada contexto.

A disciplina integra teoria e prática por meio de projetos integradores que aplicam os conceitos estudados em cenários reais de engenharia.

Linha do Tempo: Conteúdo Programático

A disciplina está organizada em **quatro unidades progressivas**, cada uma aprofundando os conceitos da anterior. Acompanhe abaixo a jornada completa do curso, do hardware ao software, da teoria à prática.

1

Unidade I

Fundamentos & Hardware

Introdução à arquitetura, operações lógico-aritméticas, ULA, linguagem de máquina, representação de instruções e componentes do processador (clock, PC, registradores).

2

Unidade II

Processador Avançado & Memória

Caminho de dados, pipeline, conflitos e desvios condicionais, hierarquia de memória, cache, memória virtual, dispositivos de I/O e multiprocessadores.

3

Unidade III

Sistema Operacional & Processos

Componentes e serviços do SO, processos, estados, threads, modelos multithreading, sincronização (semáforos, monitores), gerenciamento e memória virtual.

4

Unidade IV

Arquivos, I/O & Segurança

Sistema de arquivos (estrutura, acesso, proteção), subsistemas de I/O (polling, DMA, drivers), segurança (autenticação, criptografia, detecção de intrusão) e projetos integradores.

Cada unidade é avaliada progressivamente, e os projetos integradores ao final do curso consolidam o aprendizado em aplicações práticas de engenharia.

Unidade I — Fundamentos de Arquitetura e Hardware

Organização Interna do Computador

O computador moderno é compreendido a partir de um **modelo de sistema de computação** que integra três grandes subsistemas: a **CPU** (responsável pelo processamento), a **Memória Principal** (responsável pelo armazenamento temporário de dados e instruções) e os **Dispositivos de I/O** (responsáveis pela comunicação com o mundo externo). Esses componentes se interconectam por meio de barramentos de dados, endereço e controle.

A disciplina contextualiza o surgimento e a evolução desse modelo desde os primeiros computadores eletrônicos, passando pelas gerações de transistores, circuitos integrados e microprocessadores, até a arquitetura superescalar dos processadores modernos.

Operações Lógicas e Aritméticas

O estudante aprende a trabalhar com números com e sem sinal em binário, compreendendo representações como complemento de dois. São estudadas as **principais portas lógicas** (AND, OR, NOT, NAND, NOR, XOR), a construção de **somadores** e **subtratores**, e a organização da **Unidade Lógica Aritmética (ULA)** — bloco funcional responsável por todas as operações matemáticas e lógicas do processador.

Números com Sinal

Complemento de 2, overflow, extensão de sinal.

Portas Lógicas

AND, OR, NOT, NAND, NOR, XOR e suas tabelas verdade.

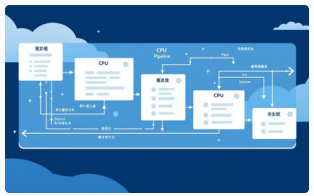
ULA

Somadores ripple-carry, carry-lookahead e operações lógicas integradas.

Linguagem de Máquina

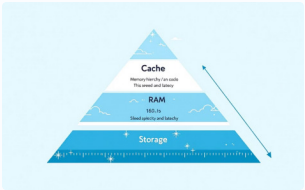
A transição do hardware para o software começa com a **linguagem de máquina**: o conjunto de instruções que o processador interpreta diretamente. São abordadas as operações executadas pelo hardware, os operandos (registradores, memória e constantes), a representação binária das instruções (formato R, I e J no MIPS, por exemplo) e as **instruções de desvio** (branch, jump) que controlam o fluxo de execução de programas.

Unidade II — Processador, Pipeline e Hierarquia de Memória



Caminho de Dados e Pipeline

O processador é detalhado em suas **unidades funcionais**: clock, Program Counter (PC), memória de instruções, banco de registradores, ULA e memória de dados. A construção do caminho de dados mostra como uma instrução percorre busca (IF), decodificação (ID), execução (EX), acesso à memória (MEM) e escrita de resultado (WB). O **pipeline** permite que várias instruções sejam executadas em sobreposição, aumentando o throughput. Conflitos de dados (data hazards) e conflitos em desvios condicionais (control hazards) são tratados com técnicas de forwarding, stalling e branch prediction.



Hierarquia de Memória

A hierarquia de memória equilibra **velocidade e custo**: registradores → cache L1/L2/L3 → RAM → disco. A **memória cache** explora localidade temporal e espacial para reduzir a latência. São estudados mapeamento direto, associatividade por conjuntos e totalmente associativo. A **memória virtual** estende a RAM usando disco, com paginação por demanda, tabelas de páginas, TLB (Translation Lookaside Buffer) e políticas de substituição de páginas (LRU, FIFO, ótima).

Periféricos e Multiprocessadores

O estudo dos **dispositivos de I/O** cobre tipos e características (disco, teclado, rede), conexão ao processador via barramentos e controladores, e a interface com a memória. Multiprocessadores são abordados em topologias de barramento compartilhado, interconexão por rede e **clusters**. A programação de sistemas multiprocessados introduz conceitos de coerência de cache e consistência de memória, fundamentais para aplicações paralelas de alto desempenho.

Unidade III — Sistema Operacional e Processos

Componentes e Serviços do Sistema Operacional

O **Sistema Operacional (SO)** é o software que faz a ponte entre o hardware e as aplicações do usuário. Seus principais componentes são: **Gerenciador de Processos** (escalonamento, criação/terminação, comunicação), **Gerenciador de Memória** (alocação, proteção, fragmentação), **Gerenciador de Arquivos** (estrutura, acesso, proteção) e **Gerenciador de I/O** (drivers, independência de dispositivos). Os serviços oferecidos incluem execução de programas, operações de I/O, manipulação de arquivos, comunicação entre processos, detecção de erros, alocação de recursos e proteção do sistema.

Processos e Threads

Um **processo** é a unidade fundamental de execução, descrita por seu Bloco de Controle de Processo (PCB), que armazena estado (novo, pronto, executando, esperando, terminado), registradores, informações de memória e I/O. **Threads** são fluxos de execução dentro de um processo, compartilhando espaço de endereçamento. São estudados modelos de threading (N:1, 1:1, N:M), threads do kernel versus threads do usuário e os benefícios do multithread para desempenho e responsividade.

Sincronização de Processos

A comunicação entre processos e threads concorrentes exige mecanismos de sincronização para evitar condições de corrida. São estudados:

1

Seção Crítica

Exclusão mútua, progresso e espera limitada — as três condições de Dijkstra para solução correta do problema.

2

Semáforos

Primitivas P (wait) e V (signal) para controle de acesso a recursos compartilhados, com exemplos de produtor-consumidor e leitores-escritores.

3

Monitores

Abstração de alto nível que encapsula variáveis compartilhadas e seus procedimentos de acesso, simplificando o raciocínio sobre sincronização.

4

Gerência de Memória

Swapping, paginação (básica, hardware e proteção), segmentação, fragmentação interna/externa e segmentação com paginação combinada.

Unidade IV — Sistema de Arquivos, I/O e Segurança

1

Sistema de Arquivos

Conceito, atributos (nome, tipo, tamanho, permissões), operações (criar, ler, escrever, excluir), tipos e estruturas internas. Métodos de acesso: **sequencial**, **direto** e outros. Estrutura de diretório (flat, árvore, grafo acíclico), montagem, compartilhamento e proteção via listas de controle de acesso (ACL).

2

Sistemas de I/O

Técnicas de comunicação com dispositivos: **polling** (consulta ativa ao status), **interrupção** (notificação assíncrona pelo hardware) e **DMA** (transferência direta à memória sem intervenção da CPU). Interface de aplicação via chamadas de sistema, subsistemas do kernel, independência de dispositivos e arquitetura de **drivers**.

3

Segurança

O problema da segurança envolve confidencialidade, integridade e disponibilidade. Tópicos: autenticação de usuário (senhas, biometria, tokens), ameaças de programa (vírus, trojans, worms) e de sistema (exploits, escalada de privilégio). Sistemas de segurança, detecção de intrusão (IDS) e fundamentos de **criptografia** simétrica (AES) e assimétrica (RSA).

Profundidade no Sistema de Arquivos

O sistema de arquivos é a abstração que organiza o armazenamento persistente. A estrutura interna pode ser baseada em **inodos** (Unix/Linux), FAT (Windows legado) ou B-trees (sistemas modernos como NTFS e ext4). A montagem (mount) integra diferentes sistemas de arquivos em um único namespace hierárquico. O compartilhamento de arquivos em redes requer protocolos como NFS e SMB, enquanto a proteção é implementada via bits de permissão (rwx) e ACLs que especificam direitos por usuário ou grupo.

Criptografia e Segurança Aplicada

A criptografia é o alicerce da segurança moderna. **Criptografia simétrica** (mesma chave para cifrar e decifrar — AES-256) oferece alta performance para grandes volumes de dados. **Criptografia assimétrica** (par de chaves pública/privada — RSA, ECC) resolve o problema de distribuição de chaves e permite assinaturas digitais. Protocolos como TLS/SSL combinam ambas as abordagens. A **detecção de intrusão** pode ser baseada em assinaturas (IDS baseado em regras) ou em anomalias (machine learning aplicado à segurança).

Jornada do Conhecimento: Evolução ao Longo do Curso

A estrutura do curso foi projetada para conduzir o estudante de forma progressiva, partindo dos conceitos mais fundamentais do hardware até as abstrações de alto nível do sistema operacional e segurança. Cada etapa consolida a anterior, criando uma base sólida e coesa.

Semanas 1–3

Modelo de computação, histórico da arquitetura, representação numérica binária e operações lógico-aritméticas. Portas lógicas, somadores e construção da ULA.

Semanas 4–6

Linguagem de máquina: conjunto de instruções, operandos, formatos e instruções de desvio. Introdução ao processador: clock, PC, banco de registradores e caminho de dados básico.

Semanas 7–9

Pipeline: visão geral, estágios IF/ID/EX/MEM/WB, hazards de dados e controle, forwarding e branch prediction. Hierarquia de memória e fundamentos de cache.

Semanas 10–12

Memória virtual, paginação por demanda, TLB. Dispositivos de I/O. Multiprocessadores: barramento, rede e clusters. Introdução ao Sistema Operacional e seus componentes.

Semanas 13–15

Processos, threads, sincronização (semáforos, monitores), gerenciamento de memória pelo SO, sistema de arquivos, I/O no nível de SO, segurança e projetos integradores.

Projetos Integradores: Ao longo do semestre, projetos práticos integram os conteúdos de múltiplas unidades, estimulando o raciocínio sistêmico e a aplicação real dos conceitos estudados em sala de aula.

Metodologia de Ensino e Avaliação

Metodologia de Ensino e Aprendizagem

As atividades são realizadas de forma **síncrona e mediada**, combinando diferentes ferramentas e estratégias pedagógicas para o desenvolvimento de competências teóricas e práticas:

- Aulas expositivas dialogadas com resolução de problemas em tempo real
- Exercícios práticos de programação em linguagem de máquina e montagem
- Simulações de pipeline e hierarquia de memória com ferramentas computacionais
- Projetos integradores em grupos para consolidação dos conteúdos
- Discussões sobre casos reais de arquiteturas modernas (ARM, x86, RISC-V)

A frequência mínima obrigatória é de **75% da carga horária**. O não cumprimento desta exigência resulta em reprovação automática, independentemente do desempenho nas avaliações.

Critérios de Avaliação e Aprovação

O aproveitamento escolar é medido por notas de **0 a 10**, com base em, no mínimo, **duas verificações parciais** por período letivo. Podem ser empregadas avaliações colegiadas, trabalhos, exercícios e projetos a critério do professor.

Média Parcial ≥ 7,0

APROVADO diretamente. Não há necessidade de Avaliação Final.

4,0 ≤ Média Parcial < 7,0

AVALIAÇÃO FINAL obrigatória. A Média Final será a média aritmética entre a Parcial e a Final.

Média Parcial < 4,0

REPROVADO sem direito à Avaliação Final.

Da Avaliação Final: **Média Final ≥ 5,0 → APROVADO** | **Média Final < 5,0 → REPROVADO**.

A Média Final é calculada como: **(Média Parcial + Nota da Avaliação Final) ÷ 2**.

Próximos Passos e Considerações Finais

A disciplina de **Arquitetura de Computadores** é um pilar central da formação em Engenharia e Ciência da Computação. O domínio dos conceitos aqui apresentados é pré-requisito para disciplinas avançadas como Sistemas Distribuídos, Computação de Alto Desempenho, Sistemas Embarcados e Segurança da Informação.

Leia Antes das Aulas

Consulte o livro base *Organização e Projeto de Computadores* (Patterson & Hennessy) e os slides disponibilizados antes de cada aula. A leitura prévia potencializa a absorção do conteúdo durante a mediação presencial.

Pratique Programação

Implemente os conceitos em linguagem de montagem (MIPS ou RISC-V) usando simuladores como MARS ou RARS. Experimente pipelines, chamadas de sistema e gerenciamento de memória em ambientes controlados.

Projetos Integradores

Forme grupos desde o início do semestre. Os projetos integradores exigem planejamento, divisão de tarefas e entregas incrementais. A organização antecipada garante qualidade e reduz estresse próximo às datas de entrega.

Monitore seu Desempenho

Acompanhe suas notas continuamente. Lembre-se: média parcial abaixo de 4,0 implica reprovação direta. Busque atendimento com o professor ou monitores antes que as dificuldades se acumulem e comprometam o aproveitamento.

> "Entender como o computador funciona internamente transforma você de um simples usuário de tecnologia em alguém capaz de criar, otimizar e inovar. Esta é a essência da Engenharia de Computação." — **Prof. Cloves Rocha**