

Sistemas Operacionais

Aula 03 – Conceitos de Hardware e Software SCC5854

Prof. Dr. Jonathan Ramos jonathan@unir.br

Departamento Acadêmico de Ciências de Computação – DACC Núcleo de Tecnologia – NT

27/09/2022

1 / 40

Sumário

- 1 Introdução
- 2 Hardware
 - Processador
 - Memória Principal
 - Memória Cache
 - Memória Secundária
 - Dispositivos de Entrada e Saída E/S
 - Barramento
 - Pipelining
 - Arquiteturas RISC e CISC
- 3 Software
 - Tradutor
 - Interpretador
 - Linker
 - Loader
 - Depurador
- 4 Exercícios

27/09/2022

Introdução



- Os conceitos básicos de hardware e software relativos à arquitetura de computadores são necessários para a compreensão dos temas que serão vistos na aula
- Não veremos de forma tão profunda. Porém as referências podem ser consultadas (TANENBAUM et al., 2009; HENNESSY; PATTERSON, 2011; STALLINGS, 2009).

4 / 40

Hardware

Um sistema Computacional é:

- Um conjunto de circuitos eletrônicos interligados;
 - Processadores, memórias, registradores, barramentos, monitores de vídeo, impressoras, mouse, teclado, discos magnéticos, entre outros (hardware).
- Manipulam dados na forma digital:

Hardware

Maneira confiável de representação e transmissão de dados.



Figura: Fonte: https://static.todamateria.com.br/upload/ha/rd/hardwareelementosfinal-cke.ipg>

Prof. Dr. Jonathan Ramos Sistemas Operacionais 27/09/2022

Hardware

Os componentes são divididos em três unidades:

Estes subsistemas estão presentes em qualquer tipo de computador digital, independentemente da arquitetura ou fabricante.

- Processador ou Unidade Central de Processamento (UCP);
- Memória Principal;
- O Dispositivos de Entrada e Saída E/S.

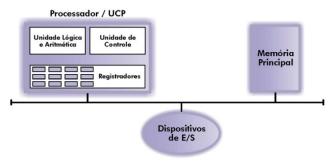


Figura: Sistema computacional.



27/09/2022

5 / 40

 Software

Exercícios

Processador





Processador

Unidade Central de Processamento (UCP)

- Gerencia todo sistema computacional;
- Controla as operações realizadas por cada unidade funcional;
- Controla e executa instruções presentes na memória principal;
- Somar, subtrair, comparar e movimentar dados.

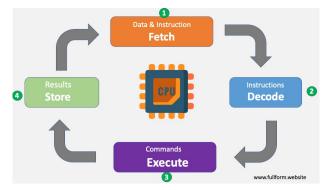


Figura: Visão Geral do Funcionamento da CPU.

Processador



Pode ser dividido em três partes principais:

- 1 Unidade de Controle (UC);
 - Responsável por gerenciar as atividades de todos os componentes do computador, como a gravação de dados em discos ou a busca de instruções na memória
- Unidade Lógica e Aritmética (ULA);
 - É responsável pela realização de operações lógicas (testes e comparações) e aritméticas (somas e subtrações).
- 3 Registradores
 - Memória temporária;

Processador



Pode ser dividido em três partes principais:

- 1 Unidade de Controle (UC);
 - Responsável por gerenciar as atividades de todos os componentes do computador, como a gravação de dados em discos ou a busca de instruções na memória
- Unidade Lógica e Aritmética (ULA);
 - É responsável pela realização de operações lógicas (testes e comparações) e aritméticas (somas e subtrações).
- 3 Registradores
 - Memória temporária;

Sinal de Clock (relógio)!

- Permite a sincronização entre todas as funções do processador.
- Gerado ciclicamente por um cristal de quartzo em uma frequência estável e bem determinado (overclock?).

Processador: registradores



Os registradores:

- Armazenam dados temporariamente.
- Alta velocidade de leitura e gravação (interno ao processador);
- Menor capacidade de armazenamento devido ao alto custo;
- A quantidade/capacidade de registradores depende da arquitetura de cada processador;

Uso específico

Responsáveis por armazenar informações de controle interno do processador/SO.

Uso geral

Podem ser manipulados diretamente por instruções.

Registradores específicos

Alguns deles merecem uma atenção especial, como veremos adiante.

Processador: Registradores de uso específico

Contador de Instruções (CI) ou Program Counter (PC)

- Contém o endereço da próxima instrução que o processador deve buscar e executar.
- Toda vez que o processador busca uma nova instrução, o PC é atualizado com o endereço de memória da instrução seguinte (a executar).

Apontador de Pilha (AP) ou Stack Pointer (SP)

- Contém o endereço de memória do topo da pilha.
- Na pilha ficam informações sobre programas que estão sendo executados.

Registrador de Instruções (RI) ou Instruction Register (IR)

Armazena a instrução que será decodificada e executada pelo processador.

Registrado de Status (RS) ou Program Status Word (PSW)

- Mantém um status do que aconteceu com as instruções (overflow, underflow).
- A maioria das instruções, quando executadas, altera o registrador de status conforme o resultado.

Prof. Dr. Jonathan Ramos Sistemas Operacionais 10 / 40

Processador: Ciclo de Busca

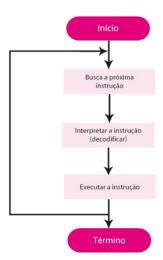


Figura: Ciclo de busca e execução de Instrucão

Ciclo de busca e instrução

- O processador busca na memória principal a instrução armazenada no endereço indicado pelo CI e a armazena no RI
- 2 O processador incrementa o CI para que o registrado contenha o endereço da próxima instrução.
- O processador decodifica a instrução armazenada no RI.
- O processador busca os operandos na memória, se houver.
- O processador executa a instrução decodificada.

Após este último passo, o ciclo é reiniciado do passo (1).

Memória Principal

Memória principal, primária ou real:

- São armazenados instruções e dados:
- Composta por unidades de acesso (células);
- Cada célula possui uma quantidade de bits;
- O bit é a unidade básica de memória (0 ou 1);
- Mais comum atualmente 8 bits por célula:

Em conclusão:

Cada célula possui um **determinado número de bits**. Uma memória é composta por **várias células**.

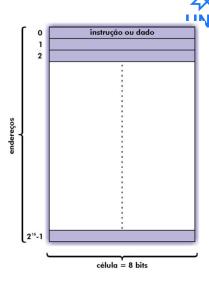


Figura: Memória principal com 64 Kbytes.

12 / 40

Para acessar uma célula

É necessário o endereço desta:

Uma referência única

Para um programa poder gravar ou ler algo na memória:

Primeiro precisa especificar o endereço da memória onde o dado será lido ou gravado. A especificação do endereço é realizada por um registrador denominado

Address	Value		
0x00	01001010		
0x01	10111010		
0x02	01011111		
0x03	00100100		
0x04	01000100		
0x05	10100000		
0x06	01110100		
0x07	01101111		
0x08	10111011		
0xFE	11011110		
0xFF	10111011		

Registrador de endereco de memória (Memory Address Register - MAR)

Com isso, a UC sabe qual endereço de memória será acessada.

Outro registrador usado em operações com a memória é

Registrador de dados da memória (Memory Buffer Register - MBR).

Guarda temporariamente o conteúdo de uma ou mais células de memória.

Prof. Dr. Jonathan Ramos

Memória Principal: acesso às células (MBR)

Registrador de dados da memória (*Memory Buffer Register* – MBR).

Ciclo de leitura e gravação:

Operação de Leitura

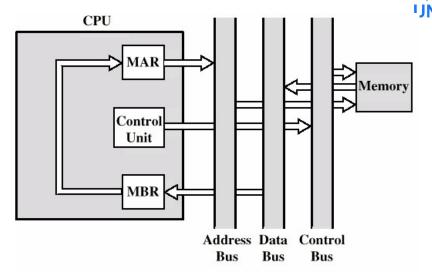
- 1 A UCP armazena no MAR o endereço da célula a ser lida.
- A UCP gera um sinal de controle para a memória principal, indicando que uma operação de leitura deve ser realizada.
- O conteúdo da(s) célula(s), identificada(s) pelo endereço contido no MAR, é transferido para o MBR.
- 4 O conteúdo do MBR é transferido para um registrador da UCP.

Operação de Gravação

- A UCP armazena no MAR o endereço da célula que será gravada.
- A UCP armazena no MBR a informação que deverá ser gravada.
- 3 A UCP gera um sinal de controle para a memória principal, indicando que uma operação de gravação deve ser realizada.
- A informação contida no MBR é transferida para a célula de memória endereçada pelo MAR.

O número de células endereçadas na memória principal é limitado pelo tamanho do MAR. No caso de o registrador possuir n bits, a memória poderá no máximo endereçar 2^n células

Memória Principal: exemplo visual de MBR e MAR



Memória Principal: tipos de memórias principais

Voláteis (RAM)

Não preserva conteúdo sem energia.

Não-Voláteis (ROM)

Preserva conteúdo mesmo sem energia.

Qual é melhor? Pq?

- DRAM:
 - Dynamic RAM;
- SRAM:
 - Static RAM.

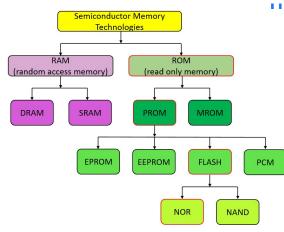


Figura: Tipos de memórias principais

https://www.techtarget.com/whatis/definition/SRAM-static-random-access-memory#:

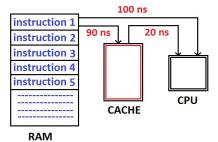
Sistemas Operacionais

~:text=SRAM%20(static%20RAM)%20is%20a,performance%20and%20lower%20power%20usage.>

16 / 40

Memória Volátil de alta velocidade

- Pouca capacidade de armazenamento (Alto Custo!).
- Diminui disparidade entre tempo que processador opera e tempo que lê/grava dados da RAM.
- É uma memória transicional. Armazena uma pequena parte da memória principal temporariamente.
- Antes de buscar na RAM, processar olha primeiro agui.
- Se achar, dizemos que houve um cache hit, caso contrário, cache miss.
- Isso diminui consideravelmente o tempo de acesso.



Memória Cache: Princípio da localidade

Princípio da localidade

- Códigos estruturados de forma modular possuem uma alta probabilidade que instruções de um mesmo programa terão endereços de memória muito próximos um do outro.
- Alta probabilidade de cache hit;
- Alta incidência de estruturas de repetição, chamadas a sub-rotinas e acesso a estrutura de dados como vetores e tabelas (matrizes).
- Otimização de tempo de acesso ao dado.

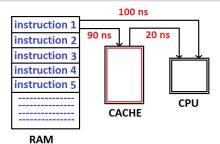


Figura: Exemplo de tempo de acesso a mémoria

19 / 40

Memória Cache

Múltiplos níveis de cache

- Algumas arquiteturas de processadores possuem essa vantagem.
- L1, L2, L3 etc.
- Quanto menor é a capacidade de armazenamento da memória cache, mais rápido é o acesso ao dado;
- Contudo, a probabilidade da ocorrência de cache hits é menor.

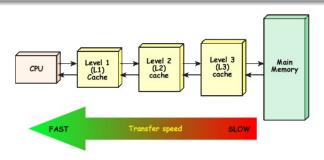


Figura: Memórias Cache e velocidade de transferência.

Prof. Dr. Jonathan Ramos Sistemas Operacionais 27/09/2022

Memória Cache

Um processador pode ser projetado com diversos níveis de cache, conforme especificação do fabricante:

Nome	L1	L2	L3
Ryzen 3 5300G	256 KB	2 MB	8 MB
Ryzen 5 5600G	384 KB	3 MB	16 MB
Ryzen 7 5700G	512 KB	4 MB	16 MB

Tabela: Exemplos de memórias caches em versões diferentes do mesmo processador.

Prof. Dr. Jonathan Ramos

Memória Secundária



É um meio permanente de armazenamento:

- Meio não volátil de armazenamento de dados e programas;
- Não precisa estar energizada (não precisa de energização);
- Acesso lento se comparado com a memória principal;
- Porém, tem um custo baixo com capacidade de armazenamento bem superior;
- Memória principal: nanossegundos:
- Memória secundária: milissegundos;
- Exemplos: HD, CD, etc

Memória Secundária

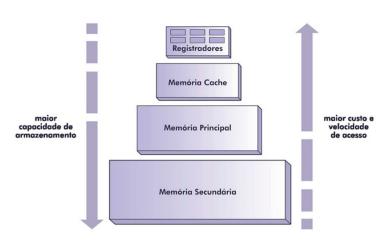


Figura: Relação entre dispositivos de armazenamento e seu custo.

Memória Secundária

	Access time	Capacity	Managed By
On the datapath Registers	I cycle	I KB	Software/Compiler
Level I Cache	2-4 cycles	32 KB	Hardware
Level 2 Cache	10 cycles	256 KB	Hardware
On chip	40 cycles	10 MB	Hardware
Other Main Memory	200 cycles	10 GB	Software/OS
chips Flash Drive	10-100us	100 GB	Software/OS
Mechanical Hard Disk	10ms	I TB	Software/OS

Figura: Comparativo: toda memória é cache para alguma outra memória (reflexão).

Dispositivos de Entrada e Saída – E/S



Permite a comunicação entre o sistema computacional e o mundo externo

Podem ser divididos em duas categorias:

- Os que são utilizados como memória secundária;
- E os que servem para a interface usuário-máquina;

Dispositivos de Entrada e Saída – E/S



Permite a comunicação entre o sistema computacional e o mundo externo

Podem ser divididos em duas categorias:

- Os que são utilizados como memória secundária:
- E os que servem para a interface usuário-máguina:

Usados como memória secundária:

- caracterizam-se por ter capacidade de armazenamento bastante superior ao da memória principal;
- Custo relativamente baixo;
- Tempo de acesso inferior ao da memória principal;

Dispositivos de Entrada e Saída - E/S



Permite a comunicação entre o sistema computacional e o mundo externo

Podem ser divididos em duas categorias:

- 1 Os que são utilizados como memória secundária;
- E os que servem para a interface usuário-máquina;

Usados como memória secundária:

- caracterizam-se por ter capacidade de armazenamento bastante superior ao da memória principal;
- Custo relativamente baixo;
- Tempo de acesso inferior ao da memória principal;

Usados como interface usuário-máquina:

- Teclado, mouse, monitor, etc
- Interfaces amigáveis permitem que usuário necessitem de menos experiência para usar;
- Quanto mais intuitivo melhor;
- Dispositivos sensíveis a voz human, tato, etc

Barramento



Barramentos

- Meio de comunicação compartilhado;
- Permite a comunicação entre as unidades funcionais de um sistema computacional:
- Por meio deste que o processador se comunica com as demais partes do sistema computacional e vice-versa. como na Figura:
 - Memória RAM, HD, teclado, mouse, etc:

Em geral possui:

Linha de controle

trafegam informações de sinalização como, por exemplo, o tipo de operação que está sendo realizada

Leitura, gravação, soma, divisão, etc

Linha de dados

Nela são transferidos, entre as unidades funcionais, informações de instruções, operandos e endereços de memória.

Sistemas Operacionais

Barramento

São divididos em três tipos principais:

- Barramento Processador-memória: curta extensão e alta velocidade;
- Barramentos de E/S: maior extensão, são mais lentos e permitem a conexão de diferentes dispositivos;
- Backplane

Memória Principal UCP Barramento processador-memória Adaptador Adaptador

Adaptador:

Permite compatibilizar as diferentes velocidades dos barramentos.

Figura: Barramentos processador-memória e de E/S.

Barramento: backplane

Tem a função de integrar os dois barramentos

 Reduz o número de adaptadores existentes no barramento processadormemória, otimizando o desempenho

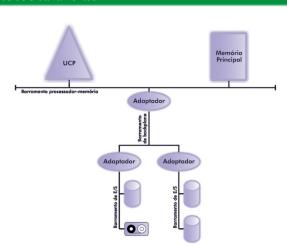


Figura: Barramento de backplane.

Pipelining

Permite ao processador executar múltiplas instruções paralelamente em estágios diferentes.

- Linha de montagem: uma tarefa é dividida em uma sequência de subtarefas.
- a execução de uma instrução pode ser dividida em subtarefas:
 - fases de busca da instrução e dos operandos, execução e armazenamento dos resultados
 - enquanto uma instrução se encontra na fase de execução, uma outra instrução possa estar na fase de busca simultaneamente

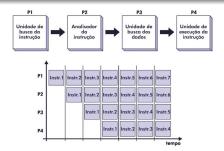


Figura: Arquitetura pipeline com quatro instruções em estágios diferentes.

Prof. Dr. Jonathan Ramos Sistemas Operacionais 27/09/2022

28 / 40

Arquiteturas RISC e CISC: linguagem de máquina

Linguagem realmente entendida pelo computador, zeros e uns:

- Cada processador possui um conjunto definido de instruções de máquina;
- As instruções de máquina fazem referências a detalhes, como registradores, modos de endereçamento e tipos de dados;
- RISC: Reduced Instruction Set Computer;
- CISC: Complex Instruction Set Computers.

Arquitetura × Arquitetura CISC:

Arquitetura RISC

- Poucas instruções
- Instruções executadas pelo hardware
- Instruções com formato fixo
- Instruções utilizam poucos ciclos de máquina
- Instruções com poucos modos de endereçamento
- 6 Arquitetura com muitos registradores
- 7 Arquitetura pipelining

Arquitetura CISC

- Muitas instruções
- Instruções executadas por microcódigo
- Instruções com diversos formatos
- 4 Instruções utilizam múltiplos ciclos
- Instruções com diversos modos de endereçamento
- 6 Arquitetura com poucos registradores
- 7 Pouco uso da técnica de pipelining

Arquiteturas RISC e CISC: linguagem de máquina



Figura: Máquina de níveis.

Na RISC

- Um programa em linguagem de máquina é executado diretamente pelo hardware
- Não ocorre nos processadores CISC, como na Figura.
- Nível intermediário, microprogramação (CISC).
- Microprogramas definenm a linguagem de máquina para CISC

Software



Interface entre as necessidade do usuário e as capacidades do hardware.

Torna o trabalho mais fácil...

Softwares são adequados para às diversas tarefas e aplicações, fazendo-os mais simples e eficientes.

Software



Interface entre as necessidade do usuário e as capacidades do hardware.

Torna o trabalho mais fácil...

Softwares são adequados para às diversas tarefas e aplicações, fazendo-os mais simples e eficientes.

Utilitários

São aqueles softwares relacionados mais diretamente com serviços complementares ao ${\sf SO}$:

- Linkers;
- Depuradores.

Software



Interface entre as necessidade do usuário e as capacidades do hardware.

Torna o trabalho mais fácil...

Softwares são adequados para às diversas tarefas e aplicações, fazendo-os mais simples e eficientes.

Utilitários

São aqueles softwares relacionados mais diretamente com serviços complementares ao SO:

- Linkers;
- Depuradores.

Software desenvolvidos pelo usuário

Chamados de aplicativos ou aplicações.

Tradutor



Apesar das inúmeras vantagens proporcionadas pelas linguagens de montagem e de alto nível, os programas escritos nessas linguagens não estão prontos para ser diretamente executados pelo processador (programas-fonte).

Etapa de conversão

Toda representação simbólica das instruções é traduzida para código de máquina. Esta conversão é realizada por um utilitário denominado tradutor.

Programa objeto

Ainda não pode ser ainda executado: dependência com sub-rotinas externas. Resolvido com o *linker*.

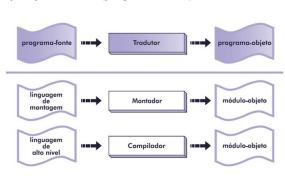


Figura: Tradutor.

Interpretador



Interpretador

- Um tradutor que não gera módulo-objeto;
- Durante a execução do programa, traduz cada instrução e a executa imediatamente;
- Não gera código executável

Desvantagem

- Tempo gasto na tradução das instruções.
- Feito toda vez que o programa é executado.

Vantagem

- Flexibilidade: Tipos de dados dinâmicos.
- Mudam de tipo durante a execução do programa.

Linker

Gera um único código executável a partir de um ou mais códigos-objeto.

UNIR

Linker

- Resolve as referências simbólicas entre módulos e aloca memória.
- Pesquisa bibliotecas do sistema (diversos módulos-objeto)

Relocação de memória

- Feita também pelo linker.
- Porém, mais complexa de ser feita em sistemas multi-programáveis (memória compartilhada)
- Código realocável: diferentes regiões de memória toda vez que for rodar
- Solução: loader.

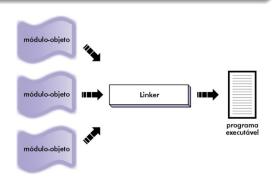


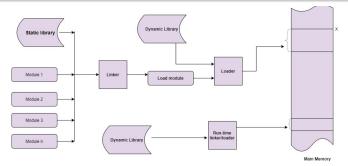
Figura: Linker.

Loader

Carrega na memória um programa para ser executado

Depende do código gerado pelo linker

- Absoluto: o loader só necessita conhecer o endereço de memória inicial e o tamanho do módulo para realizar o carregamento:
 - o loader transfere o programa da memória secundária para a memória principal e inicia sua execução.
- Relocável: o programa pode ser carregado em qualquer posição de memória
 - o loader é responsável pela relocação no momento do carregamento=.



Depurador (Debugger)

Verifica possíveis erros lógicos no código:

- Acompanha a execução de um programa instrução por instrução;
- Possibilita a alteração e a visualização do conteúdo de variáveis:
- Implementa pontos de parada dentro do programa (breakpoint), de forma que, durante a execução, o programa pare nesses pontos;
- Especificar que, toda vez que o conteúdo de uma variável for modificado, o programa envie uma mensagem (watchpoint).

0000

Exercícios

Exercícios I

- 1 Quais são as unidades funcionais de um sistema computacional?
- Quais os componentes de um processador e quais são suas funções?
- 3 Como a memória principal de um computador é organizada?
- 4 Descreva os ciclos de leitura e gravação da memória principal.
- 5 Qual o número máximo de células endereçadas em arquiteturas com MAR de 16, 32 e 64 bits?
- 6 O que são memórias voláteis e não voláteis?
- 7 Conceitue memória cache e apresente as principais vantagens no seu uso.
- Qual a importância do princípio da localidade na eficiência da memória cache?
- Quais os benefícios de uma arquitetura de memória cache com múltiplos níveis?
- Quais as diferenças entre a memória principal e a memória secundária?
- Diferencie as funções básicas dos dispositivos de E/S.
- 12 Caracterize os barramentos processador-memória, E/S e backplane.
- Como a técnica de pipelining melhora o desempenho dos sistemas computacionais?
- 14 Compare as arquiteturas de processadores RISC e CISC.
- Conceitue a técnica de benchmark e como é sua realização.

Exercícios II

- For que o código-objeto gerado pelo tradutor ainda não pode ser executado?
- 7 Por que a execução de programas interpretados é mais lenta que a de programas compilados?
- 18 Quais as funções do linker?
- 19 Qual a principal função do loader?
- Quais as facilidades oferecidas pelo depurador?



Exercícios

Referências I

HENNESSY, J.; PATTERSON, D. Computer Architecture: A Quantitative Approach. Elsevier Science, 2011. (ISSN). ISBN 9780123838735. Disponível em: ">.

STALLINGS, W. Computer Organization and Architecture: Designing for Performance. 8th. ed. USA: Prentice Hall Press, 2009. ISBN 9780136073734.

TANENBAUM, S. A. et al. Sistemas Operacionais: Projetos e Implementação. Brasil: Grupo A - Bookman, 2009.

Prof. Dr. Jonathan Ramos



FIM!

jonathan@unir.br