DCC510 – PROGRAMAÇÃO EM BAIXO NÍVEL

AULA 1

Carlos Bruno Oliveira Lopes

Engenheiro de Computação

Mestre em Ciência da Computação

EMENTA

- Linguagens de baixo nível;
- Arquitetura de um microprocessador específico;
- Linguagem Assembly;
- Processo de montagem;
- Ligação com linguagens de alto nível;

OBJETIVO GERAL

- Transmitir os conhecimentos básicos de Programação de Baixo Nível nos domínios da análise e da aplicação;
- Sedimentar a compreensão, através da programação em linguagem de máquina, dos conceitos básicos de programação: variáveis, atribuição, decisão, iteração;
- Desenvolver programas em linguagem de montagem de microprocessadores comerciais;
- Compreender o processo de montagem, ligação e carga de programas em diversas plataformas de desenvolvimento;
- Estudar a arquitetura de um processador específico permitindo o desenvolvimento de programas em sua linguagem de montagem;
- Estudar a implementação de abstrações de controle em linguagem de montagem;
- Conhecer uma linguagem que permita a programação de sistemas básicos;

BIBLIOGRAFIA BÁSICA

ZHIRKOV, Igor. **Programação em Baixo Nível: C, Assembly e Execução de Programas na Arquitetura Intel 64**. 1ª edição. Editora Novatec, 2018.

MANZANO, J. A. N. G. **Fundamentos em Programação Assembly**. 1a. ed. São Paulo: Editora Érica, 2004.

SWAN, T. **Mastering Turbo Assembler**. 2a. ed. Editora Sams, 1995.

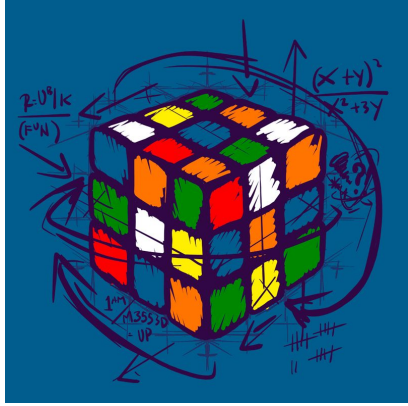
HENNESSY, J. L. & PATTERSON D. A. **Organização e Projeto de Computadores - A Interface Hardware/Software**. 2a. ed. Editora LTC, 2000.

BIBLIOGRAFIA COMPLEMENTAR

Tanenbaum, A. **Organização Estruturada de Computadores**. LTC Editora, 4a ed. 1999.

PAUL, Richard P. **Sparc architecture, assembly language programming and C**. New Jersey: Prentice-Hall, 1994.

Básico sobre arquitetura de computadores



O que um programador faz?

- Palpite: construir algoritmos e realizar sua implementação;
  - Conceber uma ideia e desenvolve-la.

+ Note: A construção de algoritmos dependerá de um conjunto de ações básicas que atuará como os blocos de construção.

  - Instruções de máquina

+ Modelo de computação: é um conjunto de operações básicas e seus respectivos custos.

  - Instrução, memória e ciclos de execução.

Básico sobre arquitetura de computadores

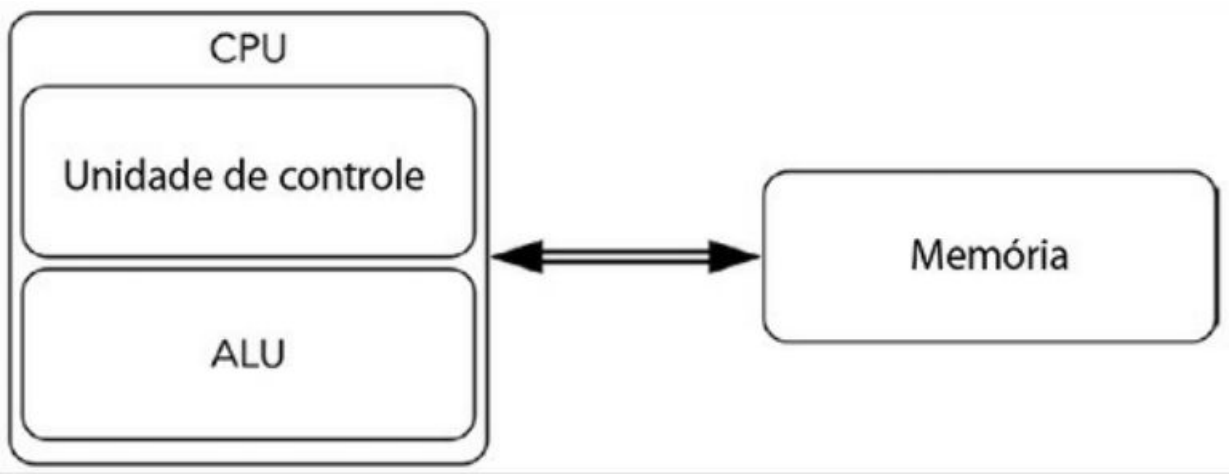
Arquitetura de von Neumann

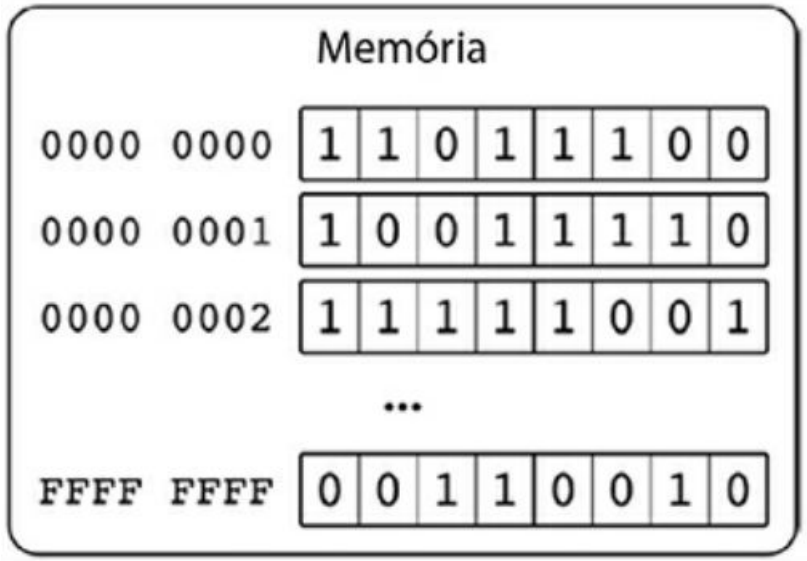
+ Arquitetura: descreve a funcionalidade, a organização e a implementação de sistemas de computadores.

- Computador constituído de um processador, memória e barramento:
  - CPU (Central Processing Unit). Capaz de executar instruções, buscadas na memória por uma unidade de controle.
  - ALU/ULA (Arithmetic Logic Unit). Executa os processamentos.
  - Memória. Armazena dados.

Básico sobre arquitetura de computadores

Arquitetura de von Neumann





Básico sobre arquitetura de computadores

Arquitetura de von Neumann

- Principais características:
  - A memória armazena somente bits.
  - A memória armazena tanto as instruções codificadas quanto os dados sobre os quais as operações serão feitas.
    - Não há nenhuma forma de distinguir dados de código: com efeito, ambos são cadeias de bits.
  - A memória está organizada em células, que são rotuladas com seus respectivos índices de forma natural.
    - Os índices começam em 0;
    - O tamanho das células pode variar;
    - Computadores modernos definem um byte (oito bits) como tamanho de um célula.
  - O programa é constituído de instruções que são buscadas uma após a outra.
    - Sua execução será sequencial, a menos que uma instrução jump especial seja executada.

Básico sobre arquitetura de computadores

Arquitetura de von Neumann

+ Linguagem Assembly: Linguagem de programação constituída de mnemônicos para cada possível instrução binária codificada (código de máquina).

- O programador não precisa memorizar a codificação binária das instruções, apenas seus nomes e os parâmetros.

Básico sobre arquitetura de computadores

Desvantagens da arquitetura de von Neumann

- **Não é nada interativa**. O programador esta limitado ao conteúdo que esta na memória e a visualização de seu conteúdo.
- **Não é apropriada à multitarefa**. Se o computador esta executando uma tarefa lenta, ou que envolva, operações de entrada/saída em periféricos. A CPU então precisa esperar a operação terminar, para executar outra tarefa.
- **Todos podem executar qualquer tipo de instrução**. Toda a espécie de comportamentos inesperados podem ocorrer.
  - Necessidade de um sistema operacional para gerenciar os recursos.
- **O desempenho da memória e da CPU diferem drasticamente**. (Velocidade)

Básico sobre arquitetura de computadores

Arquitetura Intel 64

- A Intel desenvolve sua família de processadores desde 1970;
- Seu modelos são desenvolvidos visando preservar a compatibilidade binária com os modelos mais antigos;
  - Isso resulta em um legado

Básico sobre arquitetura de computadores

Arquitetura Intel 64

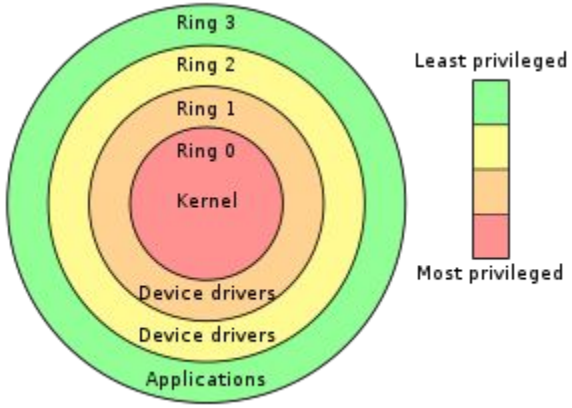
- O Intel 64 incorpora várias extensões da arquitetura de von Neumann:

  + Registradores. Células de memória colocadas diretamente no chip da CPU.

  + Pilha de hardware. Uma estrutura de dados. Operações: push e pop.

  + Interrupções. Recurso que permite alterar a ordem de execução dos programas com base em eventos externos ao próprio programa.

Básico sobre arquitetura de computadores



Arquitetura Intel 64

- O Intel 64 incorpora várias extensões da arquitetura de von Neumann:

  + Anéis de proteção. Uma CPU está sempre em um estado que corresponde a um dos chamados anéis de proteção (protection rings) – proteção hierárquica.

  - Cada anel define um conjunto de instruções permitidas. O anel zero permite executar qualquer instrução do conjunto completo de instruções da CPU e, desse modo, é o mais privilegiado.
  - O terceiro permite somente as instruções mais seguras.
  - Tentativas de executar uma instrução privilegiada resulta em uma interrupção.

  + Memória Virtual. Abstração da memória física.

Básico sobre arquitetura de computadores

Arquitetura de Von Neumann: extensões modernas

Problema	Solução
Nada é possível sem consultar a memória lenta	Registradores, caches
Falta de interatividade	Interrupções
Sem suporte para isolamento de código em procedimentos, ou para salvar contexto	Pilha de hardware
Multitarefa (multitasking): qualquer programa pode executar qualquer instrução	Anéis de proteção
Multitarefa: os programas não estão isolados uns dos outros	Memória virtual

Básico sobre arquitetura de computadores

Registradores

- Em geral, programadores trabalham com registradores de propósito geral
  - Eles são intercambiáveis e podem ser usados em vários comandos distintos
- São registradores de 64 bits, nomeados como r0, r1, ..., r15.
  - 8 primeiros bits podem receber nomes alternativos para representar algum significado em algumas instruções especiais.
    - Ex.: r1 é chamado, de modo alternativo, de rcx, em que c quer dizer “cycle” (ciclo).

    + Algumas instruções loop utilizam o rcx como contador de ciclos.

Básico sobre arquitetura de computadores

Registradores

Tabela – Registradores e 64 bits de propósito geral

Nome	Alias	Descrição
		Espécie de “acumulador”, usado em instruções aritméticas.
		Registrador base.
		Usado para ciclos.
		Armazena dados durante operações de entrada/saída.
		Armazena o enderenço do elemento do topo da pilha de hardware.
		Base do stack frame.
		Índice de origem em comandos de manipulação de strings.
		Índice de destino em comandos de manipulação de strings.
		Usado principalmente para armazenar variáveis temporárias.

Básico sobre arquitetura de computadores

Registradores

Tabela – Registradores de 64 bits de propósito geral

Básico sobre arquitetura de computadores

Registradores

- O endereçamento pode usar 32 bits, 16 bits ou 8 bits menos significativo

+ registrador:

  - d para double word – 32 bits menos significativos;
  - w para word – 16 bits menos significativos;
  - b para byte – 8 bits menos significativos.

  Ex.:

  - r7b é o byte menos significativo do registrador r7;
  - r3w é constituído dos dois bytes menos significativos de r3;
  - r0d é constituído dos quatros bytes menos significativos de r0.

Básico sobre arquitetura de computadores

Registradores

- Pode-se usar nomes alternativos para endereçar as partes menores de um registrador

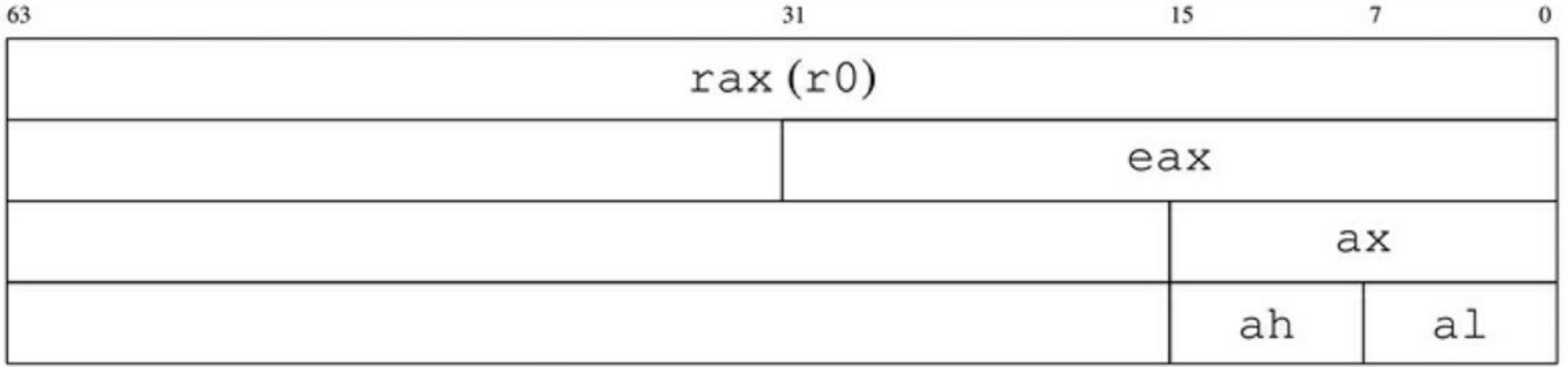


Figura - Decomposição de rax dos registradores de propósito geral em partes menores

Básico sobre arquitetura de computadores

Registradores

- Pode-se usar nomes alternativos para endereçar as partes menores de um registrador
- A convenção da nomenclatura para acessar partes de rax, rbx, rcx e rdx segue o mesmo padrão demonstrado na figura anterior.
  - Somente a letra do meio muda (a para rax).
- A nomenclatura do byte menos significativo difere um pouco para rsi, rdi, rsp e rbp
- Na prática os nome r0-r7 raramente são usados.
  - Os programadores se atêm aos nomes alternativos para os oitos primeiros registradores de propósito geral
- Os registradores r8-r15 só podem ser nomeados usando certas convenção com indexação.

Básico sobre arquitetura de computadores

Registradores

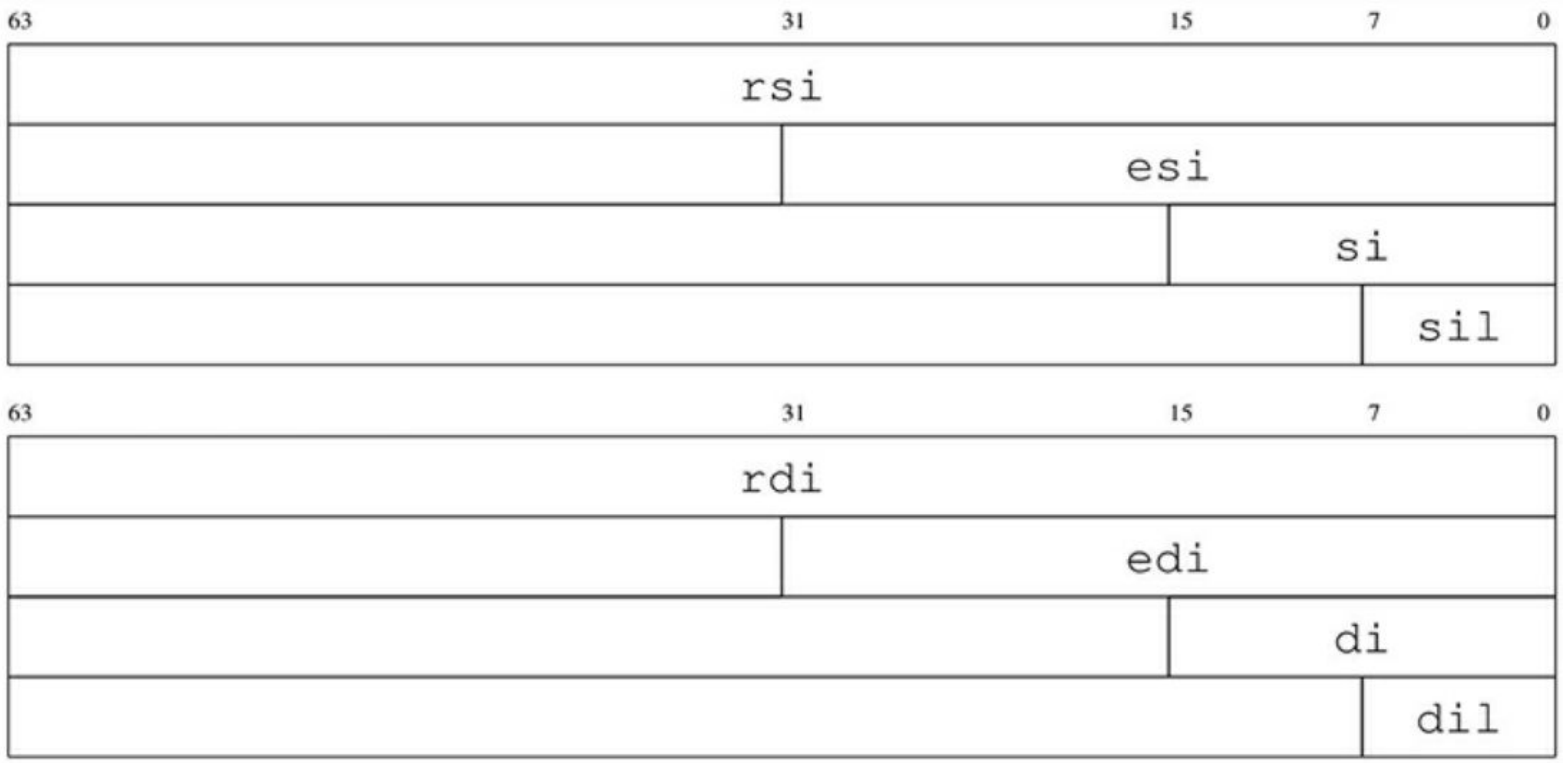


Figura - Decomposição de rsi e de rdi.

Básico sobre arquitetura de computadores

Registradores

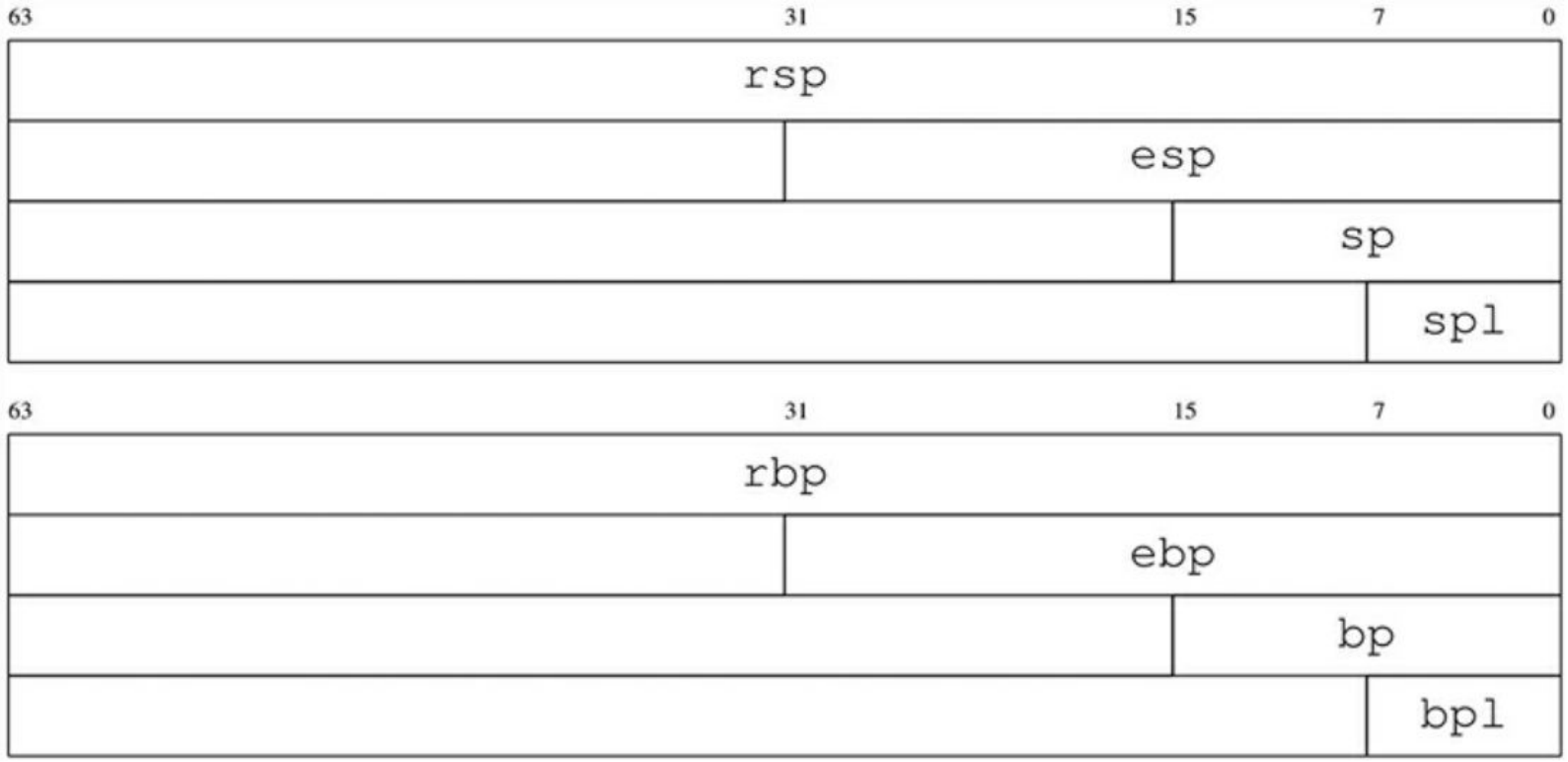


Figura - Decomposição de rsp e de rbp.

Básico sobre arquitetura de computadores

Registradores

- Outros registradores apresentam significados especiais. E em geral, são reservados e não podem ser alterados pelo programador.
  - Exceto pelo sistema operacional
- O registrador **rip**, que armazena um endereço da próxima instrução a ser executada pode ser acessado pelo programador.
  - Ex.: Usando a Instruções que fazem ramificações como jmp
- Outro registrador acessível é o **rflags**, que armazena flags, ou seja, refletem o estado atual do programa.
  - Ex.: resultado aritmético para os casos negativos ou overflow, etc.

Básico sobre arquitetura de computadores

Registradores de sistema

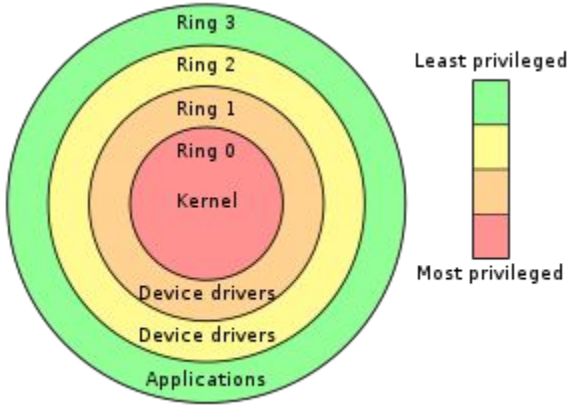
- São registradores projetados especificamente para serem usados pelo sistema operacional.
- Eles armazenam informações necessárias às estruturas de dados utilizadas por todo o sistema.
- Eles oferecem suporte para framework resultante de uma simbiose entre SO e a CPU.
- Esses registradores tem ser inacessíveis pelas aplicações, logo, as aplicações não devem ser capazes de modifica-los. (Por isso o uso do modo privilegiado)

Básico sobre arquitetura de computadores

Registradores de sistema

- cr0, cr4 armazenam flags relacionadas a diferentes modos do processador e à memória virtual.
- cr2, cr3 são usados para suporte à memória virtual.
- cr8 (cujo alias é tpr) é usado para efetuar um ajuste fino no mecanismo de interrupções.
- efer é outro registrador de flag usado para controlar os modos do processador e as extensões
- idtr armazena o endereço da tabela de descritores de interrupção.
- gdtr e ldtr armazenam os endereços das tabelas de descritores.
- cs, ds, ss, es, gs, fs são os registradores de segmento.

Básico sobre arquitetura de computadores



Anéis de proteção

- são um dos mecanismos projetados para limitar a capacidade das aplicações por razões de segurança e de robustez.
- Cada anel corresponde a um determinado nível de privilégio.
- Cada tipo de instrução está ligado a um ou mais níveis de privilégio, e não é executável em outros níveis.

Intel 64

+ Possui 4 níveis de privilégio.

+ Somente dois são usados na prática:

  Anel 0 e o Anel 3

+ Os anéis intermediários foram planejados para serem utilizados para drivers e serviços do sistema operacional, no entanto, os sistemas populares não adotaram essa abordagem.

Básico sobre arquitetura de computadores

- Pilha de hardware
  - É uma estrutura de dados, ou seja, contêiner com duas operações:
    - Um novo elemento pode ser inserido no topo da pilha (push);
    - E o elemento do topo pode ser retirado da pilha (pop).
  - "Espécie de emulação" implementada com duas instruções de máquina (push e pop) e um registrador (rsp).
  - O registrador rsp armazena o endereço do elemento que está no topo da pilha.
  - As instruções executam do seguinte modo:
    - **push** argumento
      1. Conforme o tamanho do argumento, o valor de rsp será decrementado de 2, 4 ou 8.
      2. Um argumento é armazenado na memória começando no endereço obtido do rsp modificado.
    - **pop** argumento
      1. O elemento que está no topo da pilha é copiado para o registrador/memória.
      2. rsp é incrementado com o tamanho de seu argumento.
  - Elas são mais conveniente para implementar chamadas de função em linguagens de alto nível.
  - Quando uma função A chama outra função B, ela utiliza a pilha para salvar o contexto dos processamentos e retornar a ele depois que B terminar.

Básico sobre arquitetura de computadores

Pilha de hardware

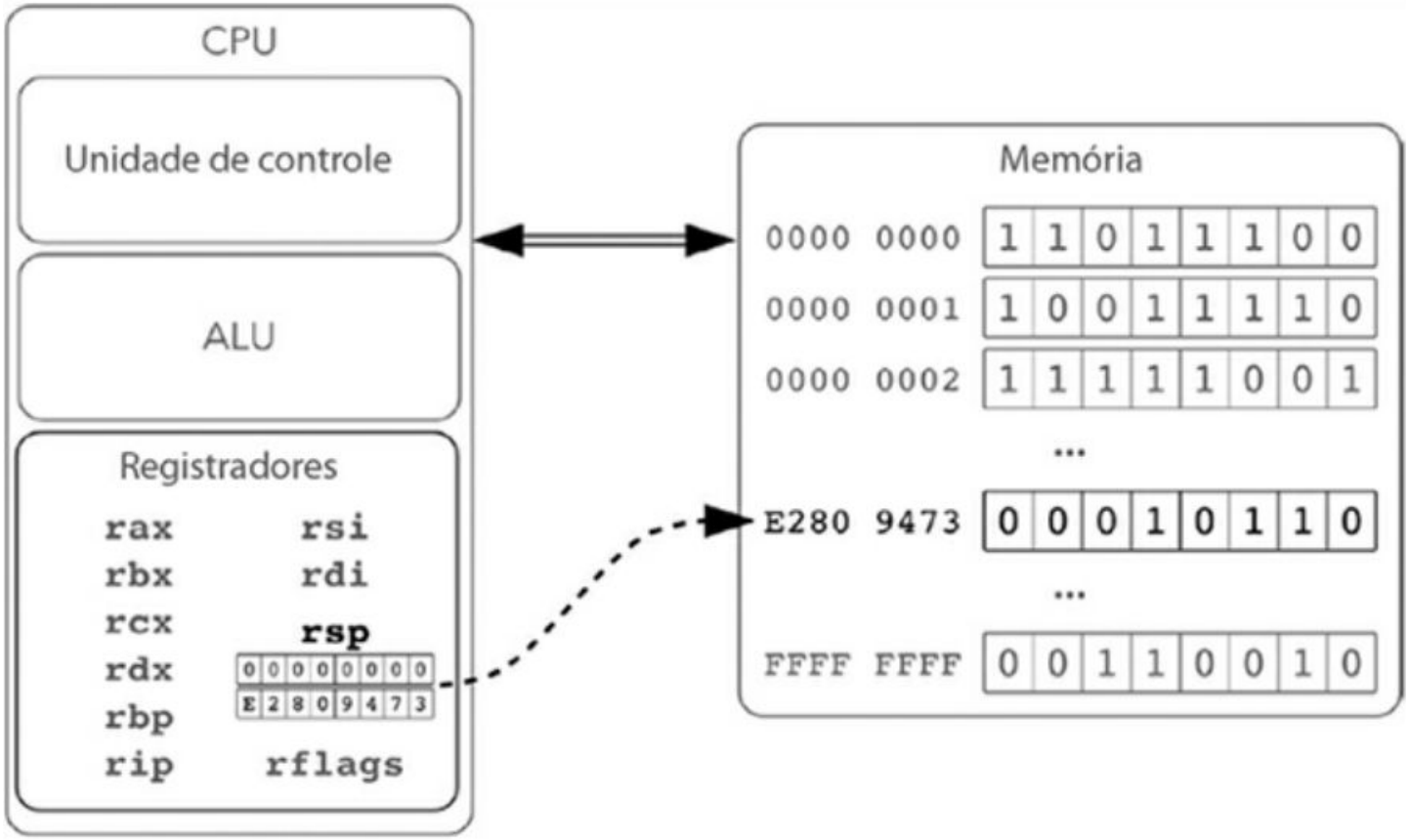


Figura – Intel 64, registradores e pilha.

Básico sobre arquitetura de computadores

- Pilha de hardware
  - Alguns detalhes relevantes sobre pilhas:
    1. Não há uma situação de pilha vazia, mesmo que não tenhamos executado **push** nenhuma vez. Um algoritmo **pop** pode ser executado de qualquer modo, provavelmente devolvendo um elemento do "topo" da pilha contendo lixo.
    2. A pilha cresce em direção ao endereço zero.
    3. Quase todos os tipos de operando são considerados inteiros com sinal e, desse modo, podem ser expandidos com um bit de sinal. Por exemplo, executar um **push** com um argumento B916 resultará na seguinte unidade de dados armazenada na pilha:

       0xff b9, 0xffffffb9 ou 0xff ff ff ff ff ff ff b9

       Por padrão, **push** utiliza um tamanho de operando de 8 bytes. Portanto, uma instrução **push** -1 armazenará 0xff ff ff ff ff ff ff ff na pilha.
    4. A maioria das arquiteturas que aceitam pilha utiliza o mesmo princípio, com seu topo definido por algum registrador. O que difere, porém, é o significado do respectivo endereço. Em algumas arquiteturas, é o endereço do próximo elemento, que será escrito no próximo **push**. Em outras, é o endereço do último elemento já inserido na pilha.

Exercícios

Baixe o documento da Microsoft que é o “Combined Volume Set of Intel® 64 and IA-32 Architectures Software Developer’s Manuals”:

https://www.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-sdm-combined-volumes-1-2a-2b-2c-2d-3a-3b-3c-3d-and-4.html

E com base na arquitetura Intel 64 responda as questões a seguir.

1. Quais são os principais registradores de propósito geral do Intel 64?
2. Qual é o propósito do ponteiro de pilha (stack pointer)?
3. Para que é usado o registrador de ponteiro de instrução e como ele é conhecido no arquitetura de 32 bits e na de 64 bits?

RESUMO

- Ementa da disciplina
- Básico sobre arquitetura de computadores
  - Arquitetura de von Neumann;
  - Arquitetura Intel 64;
  - Registradores;
  - Anéis de proteção;
  - Pilha de Hardware;