

Programação de Entrada e Saída, Diretivas Assembly

Programação Aplicada a Ciência da Computação.

Prof. Dr. Eduardo S. Pereira.

[http:](http://eduardopereira.upcursosetreinamentosonline.com/)

[//eduardopereira.upcursosetreinamentosonline.com/](http://eduardopereira.upcursosetreinamentosonline.com/)

12 de março de 2018

- 1 Introdução
- 2 Princípios do Hardware de E/S
- 3 Dispositivos de E/S
- 4 Controladores de Dispositivos de E/S
- 5 E/S Mapeada na Memória
- 6 Diretivas Assembly
- 7 Exercícios
- 8 Bibliografia

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Introdução

- Programação de Entrada e Saída, Diretivas Assembly

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Princípios do
Hardware de
E/S**

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Princípios do Hardware de E/S

- Hardwares físicos com conexões eletro/eletrônica;
- Interface de programação, comandos que o hardware aceita;
- A programação de um dispositivo de E/S está, muitas vezes, ligada a sua operação interna.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Princípios do
Hardware de
E/S**

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Princípios do Hardware de E/S

- Hardwares físicos com conexões eletro/eletrônica;
- Interface de programação, comandos que o hardware aceita;
- A programação de um dispositivo de E/S está, muitas vezes, ligada a sua operação interna.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Princípios do
Hardware de
E/S**

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Princípios do Hardware de E/S

- Hardwares físicos com conexões eletro/eletrônica;
- Interface de programação, comandos que o hardware aceita;
- A programação de um dispositivo de E/S está, muitas vezes, ligada a sua operação interna.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de E/S

- Podem ser divididos, de forma geral, em:
 - Dispositivos de Bloco
 - Dispositivos de caractere:

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de E/S

- Podem ser divididos, de forma geral, em:
- Dispositivos de Bloco
- Dispositivos de caractere:

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de E/S

- Podem ser divididos, de forma geral, em:
- Dispositivos de Bloco
- Dispositivos de caractere:

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de Bloco

- Armazena informações em blocos de tamanho fixo, cada um com seu próprio endereço;
- Os tamanhos comuns dos blocos é de 512 bytes a 32768 bytes, cada um com seu próprio endereço
- Cada bloco pode ser lido ou escrito independentemente de todos os outros.
- Os discos são dispositivos de blocos mais comuns

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de Bloco

- Armazena informações em blocos de tamanho fixo, cada um com seu próprio endereço;
- Os tamanhos comuns dos blocos é de 512 bytes a 32768 bytes, cada um com seu próprio endereço
- Cada bloco pode ser lido ou escrito independentemente de todos os outros.
- Os discos são dispositivos de blocos mais comuns

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de Bloco

- Armazena informações em blocos de tamanho fixo, cada um com seu próprio endereço;
- Os tamanhos comuns dos blocos é de 512 bytes a 32768 bytes, cada um com seu próprio endereço
- Cada bloco pode ser lido ou escrito independentemente de todos os outros.
- Os discos são dispositivos de blocos mais comuns

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de Bloco

- Armazena informações em blocos de tamanho fixo, cada um com seu próprio endereço;
- Os tamanhos comuns dos blocos é de 512 bytes a 32768 bytes, cada um com seu próprio endereço
- Cada bloco pode ser lido ou escrito independentemente de todos os outros.
- Os discos são dispositivos de blocos mais comuns

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de Caractere

- Recebe ou envia fluxo de caracteres, sem considerar qualquer estrutura de blocos.
- Eles não são endereçáveis e não dispõem de qualquer operação de posicionamento.
- Impressoras, interfaces de redes, mouses, e a maioria dos outros dispositivos que são diferentes do disco podem ser considerados de caractere.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de Caractere

- Recebe ou envia fluxo de caracteres, sem considerar qualquer estrutura de blocos.
- Eles não são endereçáveis e não dispõe de qualquer operação de posicionamento.
- Impressoras, interfaces de redes, mouses, e a maioria dos outros dispositivos que são diferentes do disco podem ser considerados de caractere.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

**Dispositivos
de E/S**

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Dispositivos de Caractere

- Recebe ou envia fluxo de caracteres, sem considerar qualquer estrutura de blocos.
- Eles não são endereçáveis e não dispõe de qualquer operação de posicionamento.
- Impressoras, interfaces de redes, mouses, e a maioria dos outros dispositivos que são diferentes do disco podem ser considerados de caractere.

Dispositivos fora da definição anterior

- Observe que os relógios não são endereçáveis por blocos nem enviam ou recebem fluxo de caracteres, tudo que fazem é causas interrupções em intervalos bem definidos;
- Os vídeos mapeados na memória também não se enquadram;
- É preciso atenção ao usar a definição anterior, mas será válida em muitos casos.

Dispositivos fora da definição anterior

- Observe que os relógios não são endereçáveis por blocos nem enviam ou recebem fluxo de caracteres, tudo que fazem é causar interrupções em intervalos bem definidos;
- Os vídeos mapeados na memória também não se enquadram;
- É preciso atenção ao usar a definição anterior, mas será válida em muitos casos.

Dispositivos fora da definição anterior

- Observe que os relógios não são endereçáveis por blocos nem enviam ou recebem fluxo de caracteres, tudo que fazem é causar interrupções em intervalos bem definidos;
- Os vídeos mapeados na memória também não se enquadram;
- É preciso atenção ao usar a definição anterior, mas será válida em muitos casos.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

**Controladores
de
Dispositivos
de E/S**

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Controladores de E/S

- As unidades de E/S consistem, usualmente, num componente mecânico e um componente eletrônico.
- O componente eletrônico é chamado de controlador do dispositivo ou adaptador
- O componente mecânico é o dispositivo propriamente dito

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

**Controladores
de
Dispositivos
de E/S**

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Controladores de E/S

- As unidades de E/S consistem, usualmente, num componente mecânico e um componente eletrônico.
- O componente eletrônico é chamado de controlador do dispositivo ou adaptador
- O componente mecânico é o dispositivo propriamente dito

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

**Controladores
de
Dispositivos
de E/S**

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Controladores de E/S

- As unidades de E/S consistem, usualmente, num componente mecânico e um componente eletrônico.
- O componente eletrônico é chamado de controlador do dispositivo ou adaptador
- O componente mecânico é o dispositivo propriamente dito

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

**Controladores
de
Dispositivos
de E/S**

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Controladores de E/S

- A placa controladora tem, em geral, um conector, no qual pode ser plugado um cabo que a conecta aos dispositivos propriamente ditos;
- A interface entre o controlador do dispositivo e o dispositivo é, com frequência, uma interface de baixo nível;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

**Controladores
de
Dispositivos
de E/S**

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Controladores de E/S

- A placa controladora tem, em geral, um conector, no qual pode ser plugado um cabo que a conecta aos dispositivos propriamente ditos;
- A interface entre o controlador do dispositivo e o dispositivo é, com frequência, uma interface de baixo nível;

Controladores de E/S

- Um disco, pode ser formatado com 256 setores de 512 bytes por trilha;
- O que realmente é entregue pela unidade de disco é um fluxo serial de bits, começando por um preâmbulo, depois 4096 bits em um setor
- por fim uma soma de verificação (checksum), código de correção de erro (error-correcting code -ECC);
- o preâmbulo é escrito quando o disco é formatado e contém o número do cilindro e do setor, o tamanho do setor e dados similares, bem como as informações de sincronização

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

**Controladores
de
Dispositivos
de E/S**

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Controladores de E/S

- Um disco, pode ser formatado com 256 setores de 512 bytes por trilha;
- O que realmente é entregue pela unidade de disco é um fluxo serial de bits, começando por um preâmbulo, depois 4096 bits em um setor
- por fim uma soma de verificação (checksum), código de correção de erro (error-correcting code -ECC);
- o preâmbulo é escrito quando o disco é formatado e contém o número do cilindro e do setor, o tamanho do setor e dados similares, bem como as informações de sincronização

Controladores de E/S

- Um disco, pode ser formatado com 256 setores de 512 bytes por trilha;
- O que realmente é entregue pela unidade de disco é um fluxo serial de bits, começando por um preâmbulo, depois 4096 bits em um setor
- por fim uma soma de verificação (checksum), código de correção de erro (error-correcting code -ECC);
- o preâmbulo é escrito quando o disco é formatado e contém o número do cilindro e do setor, o tamanho do setor e dados similares, bem como as informações de sincronização

Controladores de E/S

- Um disco, pode ser formatado com 256 setores de 512 bytes por trilha;
- O que realmente é entregue pela unidade de disco é um fluxo serial de bits, começando por um preâmbulo, depois 4096 bits em um setor
- por fim uma soma de verificação (checksum), código de correção de erro (error-correcting code -ECC);
- o preâmbulo é escrito quando o disco é formatado e contém o número do cilindro e do setor, o tamanho do setor e dados similares, bem como as informações de sincronização

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

**Controladores
de
Dispositivos
de E/S**

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Controladores de E/S

- O trabalho do controlador é converter o fluxo serial de bits num bloco de bytes e executar toda correção de erro necessário.
- O bloco de bytes é normalmente montado, bit a bit, num buffer dentro do controlador.
- Após sua soma de verificação (checksum) ter sido checada e o bloco declarado estar livre de erros, ele pode então ser copiado para a memória principal
-

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

**Controladores
de
Dispositivos
de E/S**

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Controladores de E/S

- O trabalho do controlador é converter o fluxo serial de bits num bloco de bytes e executar toda correção de erro necessário.
- O bloco de bytes é normalmente montado, bit a bit, num buffer dentro do controlador.
- Após sua soma de verificação (checksum) ter sido checada e o bloco declarado estar livre de erros, ele pode então ser copiado para a memória principal

Controladores de E/S

- O trabalho do controlador é converter o fluxo serial de bits num bloco de bytes e executar toda correção de erro necessário.
- O bloco de bytes é normalmente montado, bit a bit, num buffer dentro do controlador.
- Após sua soma de verificação (checksum) ter sido checada e o bloco declarado estar livre de erros, ele pode então ser copiado para a memória principal

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

**Controladores
de
Dispositivos
de E/S**

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Controladores de E/S

- O trabalho do controlador é converter o fluxo serial de bits num bloco de bytes e executar toda correção de erro necessário.
- O bloco de bytes é normalmente montado, bit a bit, num buffer dentro do controlador.
- Após sua soma de verificação (checksum) ter sido checada e o bloco declarado estar livre de erros, ele pode então ser copiado para a memória principal

■

E/S Mapeada na Memória

- Cada controlador tem alguns registradores usados para a comunicação com a CPU;
- Por meio da escrita nesses registradores, o sistema operacional pode comandar o dispositivo para entregar ou aceitar dados,
- alternar por si mesmo entre ligar e desligar ou executar alguma outra tarefa;
- A partir da leitura desses registradores, o sistema operacional pode descobrir o estado do dispositivo, se ele está preparado para aceitar um novo comando e assim por diante.

E/S Mapeada na Memória

- Cada controlador tem alguns registradores usados para a comunicação com a CPU;
- Por meio da escrita nesses registradores, o sistema operacional pode comandar o dispositivo para entregar ou aceitar dados,
 - alternar por si mesmo entre ligar e desligar ou executar alguma outra tarefa;
 - A partir da leitura desses registradores, o sistema operacional pode descobrir o estado do dispositivo, se ele está preparado para aceitar um novo comando e assim por diante.

E/S Mapeada na Memória

- Cada controlador tem alguns registradores usados para a comunicação com a CPU;
- Por meio da escrita nesses registradores, o sistema operacional pode comandar o dispositivo para entregar ou aceitar dados,
- alternar por si mesmo entre ligar e desligar ou executar alguma outra tarefa;
- A partir da leitura desses registradores, o sistema operacional pode descobrir o estado do dispositivo, se ele está preparado para aceitar um novo comando e assim por diante.

E/S Mapeada na Memória

- Cada controlador tem alguns registradores usados para a comunicação com a CPU;
- Por meio da escrita nesses registradores, o sistema operacional pode comandar o dispositivo para entregar ou aceitar dados,
- alternar por si mesmo entre ligar e desligar ou executar alguma outra tarefa;
- A partir da leitura desses registradores, o sistema operacional pode descobrir o estado do dispositivo, se ele está preparado para aceitar um novo comando e assim por diante.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória

- Além dos registradores, muitos dispositivos têm um buffer de dados que o sistema operacional pode ler ou escrever.
- Como o CPU se comunica com os registradores dos controladores e com os buffers de dados dos dispositivos?
- Existem duas possibilidades:

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória

- Além dos registradores, muitos dispositivos têm um buffer de dados que o sistema operacional pode ler ou escrever.
- Como o CPU se comunica com os registradores dos controladores e com os buffers de dados dos dispositivos?
- Existem duas possibilidades:

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória

- Além dos registradores, muitos dispositivos têm um buffer de dados que o sistema operacional pode ler ou escrever.
- Como o CPU se comunica com os registradores dos controladores e com os buffers de dados dos dispositivos?
- Existem duas possibilidades:

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória - Caso 1.

- Cada registrador de controle é associado a um número de porta de E/S, um inteiro de 8 ou 16 bits.
- Usando um operação especial de E/S como: IN REG, PORT
- A CPU pode ler o registrador de controle PORT e armazenar o resultado no registrador REG da CPU.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória - Caso 1.

- Cada registrador de controle é associado a um número de porta de E/S, um inteiro de 8 ou 16 bits.
- Usando um operação especial de E/S como: IN REG, PORT
- A CPU pode ler o registrador de controle PORT e armazenar o resultado no registrador REG da CPU.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória - Caso 1.

- Cada registrador de controle é associado a um número de porta de E/S, um inteiro de 8 ou 16 bits.
- Usando um operação especial de E/S como: IN REG, PORT
- A CPU pode ler o registrador de controle PORT e armazenar o resultado no registrador REG da CPU.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória - Caso 1.

- Como no caso anterior usando: OUT PORT, REG
- a CPU pode escrever os conteúdos de REG para o registrador de controle PORT.
- A maioria dos principais computadores funcionam assim.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória - Caso 1.

- Como no caso anterior usando: OUT PORT, REG
- a CPU pode escrever os conteúdos de REG para o registrador de controle PORT.
- A maioria dos principais computadores funcionam assim.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória - Caso 1.

- Como no caso anterior usando: OUT PORT, REG
- a CPU pode escrever os conteúdos de REG para o registrador de controle PORT.
- A maioria dos principais computadores funcionam assim.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória - Caso 2.

- Visa mapear todos os registradores de controle no espaço de endereçamento da memória.
- Cada registrador de controle é associado a um endereço de memória único ao qual nenhuma memória é associada.
- Esse sistema é chamado de E/S mapeada na memória.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória - Caso 2.

- Visa mapear todos os registradores de controle no espaço de endereçamento da memória.
- Cada registrador de controle é associado a um endereço de memória único ao qual nenhuma memória é associada.
- Esse sistema é chamado de E/S mapeada na memória.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

**E/S Mapeada
na Memória**

Diretivas
Assembly

Exercícios

Bibliografia

E/S Mapeada na Memória - Caso 2.

- Visa mapear todos os registradores de controle no espaço de endereçamento da memória.
- Cada registrador de controle é associado a um endereço de memória único ao qual nenhuma memória é associada.
- Esse sistema é chamado de E/S mapeada na memória.

E/S Mapeada na Memória - Caso 2.

- Quando são necessárias instruções especiais de E/S para ler ou escrever nos registradores dos dispositivos, o acesso a eles requer o uso de códigos específicos em Assembly, pois não existe nenhum modo de executar instruções IN ou OUT em C/C++;
- Com E/S mapeado na memória, os registradores de controle do dispositivo são apenas variáveis na memória e podem ser endereçados em C da mesma maneira que qualquer outra variável.
- Com E/S mapeado na memória, um driver pode ser escrito em C. Sem E/S mapeada na memória, algum código em Assembly é necessário.
- Mas o mapeamento em memória leva a complexidade na construção de projetos de hardware.

E/S Mapeada na Memória - Caso 2.

- Quando são necessárias instruções especiais de E/S para ler ou escrever nos registradores dos dispositivos, o acesso a eles requer o uso de códigos específicos em Assembly, pois não existe nenhum modo de executar instruções IN ou OUT em C/C++;
- Com E/S mapeado na memória, os registradores de controle do dispositivo são apenas variáveis na memória e podem ser endereçados em C da mesma maneira que qualquer outra variável.
- Com E/S mapeado na memória, um driver pode ser escrito em C. Sem E/S mapeada na memória, algum código em Assembly é necessário.
- Mas o mapeamento em memória leva a complexidade na construção de projetos de hardware.

E/S Mapeada na Memória - Caso 2.

- Quando são necessárias instruções especiais de E/S para ler ou escrever nos registradores dos dispositivos, o acesso a eles requer o uso de códigos específicos em Assembly, pois não existe nenhum modo de executar instruções IN ou OUT em C/C++;
- Com E/S mapeado na memória, os registradores de controle do dispositivo são apenas variáveis na memória e podem ser endereçados em C da mesma maneira que qualquer outra variável.
- Com E/S mapeado na memória, um driver pode ser escrito em C. Sem E/S mapeada na memória, algum código em Assembly é necessário.
- Mas o mapeamento em memória leva a complexidade na construção de projetos de hardware.

E/S Mapeada na Memória - Caso 2.

- Quando são necessárias instruções especiais de E/S para ler ou escrever nos registradores dos dispositivos, o acesso a eles requer o uso de códigos específicos em Assembly, pois não existe nenhum modo de executar instruções IN ou OUT em C/C++;
- Com E/S mapeado na memória, os registradores de controle do dispositivo são apenas variáveis na memória e podem ser endereçados em C da mesma maneira que qualquer outra variável.
- Com E/S mapeado na memória, um driver pode ser escrito em C. Sem E/S mapeada na memória, algum código em Assembly é necessário.
- Mas o mapeamento em memória leva a complexidade na construção de projetos de hardware.

E/S Mapeada na Memória

Arquitetura 8086

CPU	Registradores de 16 bits
Memória	Memória limitada a 1MB
	Dividida em segmentos 64kb
	Somente modo real
	Bytes na memória não possuem endereço único
	Organização <i>little endian</i>

E/S Mapeada na Memória

• Mapeamento da memória

0x00000 - 0x003FF	Tabela de interrupções (ISR)
0x00400 - 0x005FF	Área de BIOS (BDA)
0x00600 - 0x9FFFF	Área livre
0xA0000 - 0xAFFFF	Memória de vídeo EGA/VGA
0xB0000 - 0xB7FFF	Memória de texto monocromático
0xB8000 - 0xBFFFF	Memória de vídeo CGA
0xC0000 - 0xDFFFF	ROM instalada
0xE0000 - 0xFDFFF	ROM fixa
0xFE000 - 0xFFFFF	ROM da BIOS

Diretivas Assembly

- **Artifício do assembler, não da CPU**
- Usadas tanto para instruir o assembler para fazer alguma coisa ou informar o assembler de alguma coisa
- Elas não são traduzidas em código de máquina. Usos comuns de diretivas são:
 - O rótulo `section` serve para definir seções de código

Diretivas Assembly

- Artifício do assembler, não da CPU
- Usadas tanto para instruir o assembler para fazer alguma coisa ou informar o assembler de alguma coisa
- Elas não são traduzidas em código de máquina. Usos comuns de diretivas são:

■ Definir constantes;

■ Definir memória para armazenar dados;

■ Definir rotinas para serem chamadas;

■ Definir macros para serem chamadas;

■ Definir seções de código;

- O rótulo `section` serve para definir seções de código

Diretivas Assembly

- Artifício do assembler, não da CPU
- Usadas tanto para instruir o assembler para fazer alguma coisa ou informar o assembler de alguma coisa
- Elas não são traduzidas em código de máquina. Usos comuns de diretivas são:
 - i Definir constantes;
 - ii Definir memória para armazenar dados;
 - iii Agrupar memória em segmentos;
 - iv Condicionalmente incluir código fonte;
 - v Incluir outros arquivos.
- O rótulo `section` serve para definir seções de código

Diretivas Assembly

- Artífício do assembler, não da CPU
- Usadas tanto para instruir o assembler para fazer alguma coisa ou informar o assembler de alguma coisa
- Elas não são traduzidas em código de máquina. Usos comuns de diretivas são:
 - i Definir constantes;
 - ii Definir memória para armazenar dados;
 - iii Agrupar memória em segmentos;
 - iv Condicionalmente incluir código fonte;
 - v Incluir outros arquivos.
- O rótulo `section` serve para definir seções de código

Diretivas Assembly

- Artífício do assembler, não da CPU
- Usadas tanto para instruir o assembler para fazer alguma coisa ou informar o assembler de alguma coisa
- Elas não são traduzidas em código de máquina. Usos comuns de diretivas são:
 - i Definir constantes;
 - ii Definir memória para armazenar dados;
 - iii Agrupar memória em segmentos;
 - iv Condicionalmente incluir código fonte;
 - v Incluir outros arquivos.
- O rótulo `section` serve para definir seções de código

Diretivas Assembly

- Artifício do assembler, não da CPU
- Usadas tanto para instruir o assembler para fazer alguma coisa ou informar o assembler de alguma coisa
- Elas não são traduzidas em código de máquina. Usos comuns de diretivas são:
 - i Definir constantes;
 - ii Definir memória para armazenar dados;
 - iii Agrupar memória em segmentos;
 - iv Condicionalmente incluir código fonte;
 - v Incluir outros arquivos.
- O rótulo `section` serve para definir seções de código

Diretivas Assembly

- Artifício do assembler, não da CPU
- Usadas tanto para instruir o assembler para fazer alguma coisa ou informar o assembler de alguma coisa
- Elas não são traduzidas em código de máquina. Usos comuns de diretivas são:
 - i Definir constantes;
 - ii Definir memória para armazenar dados;
 - iii Agrupar memória em segmentos;
 - iv Condicionalmente incluir código fonte;
 - v Incluir outros arquivos.
- O rótulo `section` serve para definir seções de código

Diretivas Assembly

- Artífício do assembler, não da CPU
- Usadas tanto para instruir o assembler para fazer alguma coisa ou informar o assembler de alguma coisa
- Elas não são traduzidas em código de máquina. Usos comuns de diretivas são:
 - i Definir constantes;
 - ii Definir memória para armazenar dados;
 - iii Agrupar memória em segmentos;
 - iv Condicionalmente incluir código fonte;
 - v Incluir outros arquivos.
- O rótulo `section` serve para definir seções de código

Diretivas Assembly

- Artifício do assembler, não da CPU
- Usadas tanto para instruir o assembler para fazer alguma coisa ou informar o assembler de alguma coisa
- Elas não são traduzidas em código de máquina. Usos comuns de diretivas são:
 - i Definir constantes;
 - ii Definir memória para armazenar dados;
 - iii Agrupar memória em segmentos;
 - iv Condicionalmente incluir código fonte;
 - v Incluir outros arquivos.
- O rótulo section serve para definir seções de código

Seções

- **session .data:** região do código que será usada para tratar as informações, os dados, as variáveis. Nesse trecho (geralmente o inicial), declaramos e inicializamos as variáveis.
- **session .text:** é o local onde irá ficar armazenado suas instruções, que irão trabalhar com os dados previamente declarados. Essa é a única seção obrigatória, pois conterà a label (rótulo) `_start/_main/main/cmain`, que é o local onde os executáveis são inicializados
- **session .bss:** parecido com a `.data`, mas não faz inicialização do conteúdo.

Seções

- `session .data`: região do código que será usada para tratar as informações, os dados, as variáveis. Nesse trecho (geralmente o inicial), declaramos e inicializamos as variáveis.
- `session .text`: é o local onde irá ficar armazenado suas instruções, que irão trabalhar com os dados previamente declarados. Essa é a única seção obrigatória, pois conterá a label (rótulo) `_start/_main/main/cmain`, que é o local onde os executáveis são inicializados
- `session .bss`: parecido com a `.data`, mas não faz inicialização do conteúdo.

Seções

- `session .data`: região do código que será usada para tratar as informações, os dados, as variáveis. Nesse trecho (geralmente o inicial), declaramos e inicializamos as variáveis.
- `session .text`: é o local onde irá ficar armazenado suas instruções, que irão trabalhar com os dados previamente declarados. Essa é a única seção obrigatória, pois conterá a label (rótulo) `_start/_main/main/cmain`, que é o local onde os executáveis são inicializados
- `session .bss`: parecido com a `.data`, mas não faz inicialização do conteúdo.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

global

- A diretiva global mostra que o label relacionado é global, ou seja, é passível de uso externamente.
- Usamos para declarar o rótulo principal (`_start/_main/main/cmain`)
- `session .bss`: parecido com a `.data`, mas não faz inicialização do conteúdo.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

global

- A diretiva global mostra que o label relacionado é global, ou seja, é passível de uso externamente.
- Usamos para declarar o rótulo principal (`_start/_main/main/cmain`)
- `session .bss`: parecido com a `.data`, mas não faz inicialização do conteúdo.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

global

- A diretiva global mostra que o label relacionado é global, ou seja, é passível de uso externamente.
- Usamos para declarar o rótulo principal (`_start/_main/main/cmain`)
- `session .bss`: parecido com a `.data`, mas não faz inicialização do conteúdo.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Esqueleto de um código assembly.

```
%include "io.inc"
SECTION .data
;
SECTION .bss
;
section .text

global CMAIN
CMAIN:
    ;write your code here
    xor eax, eax
    ret
```

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Pseudo-Instruções - NASM

- DB, DW, DD, DQ, DT, DO, DY, DZ - São usadas para declarar dados inicializados no arquivo de saída.
- DT, DO, DY e DZ não aceitam constantes numéricas como operandos.
- `session .bss:` parecido com a `.data`, mas não faz inicialização do conteúdo.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Pseudo-Instruções - NASM

- DB, DW, DD, DQ, DT, DO, DY, DZ - São usadas para declarar dados inicializados no arquivo de saída.
- DT, DO, DY e DZ não aceitam constantes numéricas como operandos.
- `section .bss:` parecido com a `.data`, mas não faz inicialização do conteúdo.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Pseudo-Instruções - NASM

- DB, DW, DD, DQ, DT, DO, DY, DZ - São usadas para declarar dados inicializados no arquivo de saída.
- DT, DO, DY e DZ não aceitam constantes numéricas como operandos.
- `session .bss`: parecido com a `.data`, mas não faz inicialização do conteúdo.

Pseudo-Instruções - NASM

```
db    0x55                ; just the byte 0x55
db    0x55,0x56,0x57      ; three bytes in succession
db    'a',0x55             ; character constants are OK
db    'hello',13,10,'$'    ; so are string constants
dw    0x1234               ; 0x34 0x12
dw    'a'                  ; 0x61 0x00 (it's just a number)
dw    'ab'                 ; 0x61 0x62 (character constant)
dw    'abc'                ; 0x61 0x62 0x63 0x00 (string)
dd    0x12345678           ; 0x78 0x56 0x34 0x12
dd    1.234567e20          ; floating-point constant
dq    0x123456789abcdef0   ; eight byte constant
dq    1.234567e20          ; double-precision float
dt    1.234567e20          ; extended-precision float
```

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Pseudo-Instruções - NASM

- RESB, RESQ, RESD, RESQ, REST, RESO, RESY, RESZ, são usadas na seção BSS. Elas declaram espaço de memória não inicializado.

Pseudo-Instruções - NASM

buffer:	resb	64	; reserve 64 bytes
wordvar:	resw	1	; reserve a word
realarray	resq	10	; array of ten reals
ymmval:	resy	1	; one YMM register
zmmvals:	resz	32	; 32 ZMM registers

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Pseudo-Instruções - NASM

- EQU: define um símbolo a um valor constante. Quando EQU é usado, a linha de comando precisa ter um rótulo.
`message db 'oi, mundo'`
`msglen equ $-message`
- `msglen` não poderá ser redefinido depois.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Pseudo-Instruções - NASM

- EQU: define um símbolo a um valor constante. Quando EQU é usado, a linha de comando precisa ter um rótulo.
`message db 'oi, mundo'`
`msglen equ $-message`
- msglen não poderá ser redefinido depois.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Pseudo-Instruções - NASM

- **TIMES**: faz com que a instrução seja montada várias vezes.
buffer: db 'oi, mundo'
times 64-\$-buffer db ' '
- Irá armazenar espaço exato para fazer com que o comprimento de buffer vezes 64.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Pseudo-Instruções - NASM

- **TIMES:** faz com que a instrução seja montada várias vezes.
buffer: db 'oi, mundo'
times 64-\$-buffer db ' '
- Irá armazenar espaço exato para fazer com que o comprimento de buffer vezes 64.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Endereço Efetivo - NASM

- É um operador para uma instrução de referência em memória.
- `wordvar dw 123`
`mov eax, [wordvar]`

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

**Diretivas
Assembly**

Exercícios

Bibliografia

Endereço Efetivo - NASM

- É um operador para uma instrução de referência em memória.
- `wordvar dw 123`
`mov eax, [wordvar]`

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Exercícios

- Uma forma de acessar dados numa string é dados pelo código a seguir:

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

```
%include "io.inc"
SECTION .data
    TEXTO: db 'ABCDEFGF'
    msglen: equ $-TEXTO
    MSG: db ''
    formatout: db "%", 10, 0
SECTION .text
    global main
    extern scanf
    extern printf
```

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

```
main:
    push ebx
    push ecx
    mov ebx, TEXTO
    add ebx, 1
    inc ebx
    mov al, [ebx]
    mov [MSG], al
    push MSG
    push formatout
    call printf
    add esp, 8
    pop ecx
    pop ebx
    mov eax, 0
    ret
```

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Exercícios

- Escreva um programa que printe na tela um elemento por linha da string. Use o `inc`, também use a informação: `cmp ebx, TEXTO + msglen` irá comparar se o `ebx` chegou ao valor esperado para o tamanho do texto. E que a verificação que os dois são iguais é dada por `JB LOOP`, ou seja, se o valor em `ebx` já for igual ao tamanho efetivo do texto, pule para o rótulo `LOOP`.
- Com isso em mente, qual o motivo dos índices de arrays em C começar com zero?

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Livros Texto

- AHO, A; ULLMANN, J; REVI, S. Compiladores: Princípios, técnicas e ferramentas 3 ed. Rio de Janeiro: LTC-Livros
- LOUDEN, Kenneth C; SILVA, Flávio Soares Corrêa. Compiladores : princípios e práticas. 1a ed. São João da Boa Vista: Pioneira - Thomson Learning, 2004.
- PRICE, Ana M. A.; TOSCANI, Simão S.. Implementação de linguagens de programação: compiladores. 3a ed. Porto Alegre: Bookman, 2008.
- SETZER, V.W.. Construção de um Compilador. 1a ed. Rio de Janeiro: Campus - Elsevier, 1983.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Princípios do
Hardware de
E/S

Dispositivos
de E/S

Controladores
de
Dispositivos
de E/S

E/S Mapeada
na Memória

Diretivas
Assembly

Exercícios

Bibliografia

Grato

MUITO OBRIGADO.