



Gerência de Memória

Universidade Federal de Uberlândia
Faculdade de Computação
Prof. Dr. rer. nat. Daniel D. Abdala

Na Aula Anterior...

- Comunicação entre Processos
 - Pipes;
 - Memória compartilhada;
 - Sockets;
- Seção Crítica;
- Suporte em Hardware para Sincronismo;
- Semáforos;
- Mutexes;
- Barreiras;

2

Nesta Aula

- Gerenciamento sem abstração de memória;
 - Multiprogramação sem abstração de memória;
 - Realocação estática;
 - Hardware de proteção;
- Espaço de endereçamento;
 - Realocação Dinâmica (registradores base | limite);
 - Troca de Memória;
 - Memória Virtual;

3

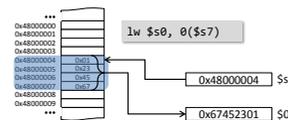
Considerações Iniciais

- Memória RAM é um recurso fundamental em sistemas computacionais;
- Idealmente:
 - máximo de memória possível;
 - Mais rápida possível;
 - Mais barata possível;
 - Não volátil;
- Realisticamente → **Hierarquia de Memórias**;
- **Gerenciador de Memória** é a parte do SO responsável por gerenciar de forma eficiente (parte) a Hierarquia de Memória;
 - Quais partes da memória em uso e disponíveis
 - Alocação e liberação de memória para processos quando requisitado

4

Sem Abstração de Memória

- Modo mais simples é simplesmente não gerenciar;
- Todos os endereços gerados são endereços reais;
- Usada nos primeiros computadores;
- Hoje em dia apenas em sistemas muito simples como sistemas embutidos (rádio, microondas, etc)



5

Multiprogramação sem Abstração de Memória

- Multiprogramação em Memória Absoluta?
 - É possível mas geralmente não é feito!
- Realocação Estática:
 - Recalcula endereços durante o carregamento;
 - Carregamento mais lento;
- Hardware de proteção:
 - Memória dividida em blocos (2k);
 - Adiciona-se uma chave de K bits a cada bloco

6

Realocação Estática

Program 1

```

0x0000.0000 add $t0, $zero, $zero
0x0000.0004 add $t1, $zero, $zero
0x0000.0008 add $t1, $zero, $zero
0x0000.000C bne $t0, $t1, 0x000012
0x0000.0010 add $t0, $t0, $t0
0x0000.0014 add $t0, $t0, $t0
0x0000.0018 add $t0, $t0, $t0
0x0000.001C add $t0, $t0, $t0
0x0000.0020 add $t0, $t0, $t0
0x0000.0024 add $t0, $t0, $t0
0x0000.0028 add $t0, $t0, $t0
0x0000.0034 add $t0, $t0, $t0
0x0000.0038 add $t0, $t0, $t0
0x0000.0044 add $t0, $t0, $t0
0x0000.0048 add $t0, $t0, $t0
0x0000.0054 add $t0, $t0, $t0
0x0000.0058 add $t0, $t0, $t0
0x0000.0064 add $t0, $t0, $t0
0x0000.0068 add $t0, $t0, $t0
0x0000.0074 add $t0, $t0, $t0
0x0000.0078 li $v0, 10
0x0000.007C syscall
    
```

Program 2

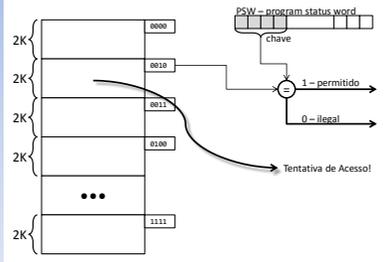
```

0x0000.0000 lui $at, 0x0000
0x0000.0004 ori $t0, $at, 0x0100
0x0000.0008 add $t1, $zero, $zero
0x0000.000C add $t1, $zero, $zero
0x0000.0010 add $t1, $zero, $zero
0x0000.0014 add $t1, $zero, $zero
0x0000.0018 add $t1, $zero, $zero
0x0000.001C add $t1, $zero, $zero
0x0000.0020 add $t1, $zero, $zero
0x0000.0024 add $t1, $zero, $zero
0x0000.0028 add $t1, $zero, $zero
0x0000.0034 add $t1, $zero, $zero
0x0000.0038 add $t1, $zero, $zero
0x0000.0044 add $t1, $zero, $zero
0x0000.0048 add $t1, $zero, $zero
0x0000.0054 add $t1, $zero, $zero
0x0000.0058 add $t1, $zero, $zero
0x0000.0064 add $t1, $zero, $zero
0x0000.0068 add $t1, $zero, $zero
0x0000.0074 add $t1, $zero, $zero
0x0000.0078 li $v0, 10
0x0000.007C syscall
    
```

Segundo Segmento de 2K
Começa em 0x0000 (0x0200 words)

4 bits mais significativos do PC

Hardware de Proteção



Espaço de Endereçamento

- Conjunto de endereços que um processo pode endereçar a memória;
- Individual para cada processo;
- Endereços não são absolutos, devem ser realocados;
- Endereços fora da faixa do **espaço de endereçamento** são **ilegais** e geram uma **exceção**;
- Várias técnicas:
 - Realocação Dinâmica (registadores base|limite);
 - Troca de Memória;
 - Memória Virtual;

Espaço de Endereçamento

Tradução de endereços

Espaço de endereçamento

Registadores Base | Limite

- Modo de realocação dinâmico de endereços de memória;
- Usado desde o CDC 6600 (primeiro supercomputador);
- Dois registadores são adicionados ao processador;
 - Suporte a realocação em hardware;
 - Registrador base (RB) e registrador limite (RL)
 - (RB) é adicionado ao endereço gerado pelo processo;
 - O endereço gerado é verificado para detectar acessos fora do espaço de endereçamento do processo via comparação com o valor contido em (RL)

Registadores Base | Limite

Program 2

```

0x0000.0000 add $t0, $zero, $zero
0x0000.0004 add $t1, $zero, $zero
0x0000.0008 add $t1, $zero, $zero
0x0000.000C bne $t0, $t1, 0x000012
0x0000.0010 add $t0, $t0, $t0
0x0000.0014 add $t0, $t0, $t0
0x0000.0018 add $t0, $t0, $t0
0x0000.001C add $t0, $t0, $t0
0x0000.0020 add $t0, $t0, $t0
0x0000.0024 add $t0, $t0, $t0
0x0000.0028 add $t0, $t0, $t0
0x0000.0034 add $t0, $t0, $t0
0x0000.0038 add $t0, $t0, $t0
0x0000.0044 add $t0, $t0, $t0
0x0000.0048 add $t0, $t0, $t0
0x0000.0054 add $t0, $t0, $t0
0x0000.0058 add $t0, $t0, $t0
0x0000.0064 add $t0, $t0, $t0
0x0000.0068 add $t0, $t0, $t0
0x0000.0074 add $t0, $t0, $t0
0x0000.0078 li $v0, 10
0x0000.007C syscall
    
```

RB 0x00000800

0x000012

RL 0x00001000

1 - permitido

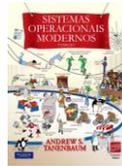
0 - ilegal

Quando a Memória é menor que o Apetite dos Processos ...

- Os esquemas de gerência de memória vistos até aqui funcionam bem se a memória física for maior ou igual a memória necessária para os processos;
- O que ocorre quando há necessidade de mais memória que a disponível:
- Duas possibilidades:
 1. Swapping;
 2. Memória Virtual;

13

Bibliografia - Básica



- 3ª Edição
- Páginas 57-68

14

Bibliografia - Básica

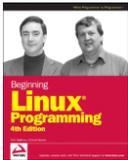


- 7ª edição
- Páginas 127-150

15

Bibliografia - Adicional

- Capítulo 12



16