Departamento de Computação

SISTEMAS OPERACIONAIS 1
21270 A

Departamento de Computação
Prof. Kelen Cristiane Teixeira Vivaldini

Departamento de Computação

Lab. Gerenciamento de Memória

Registradores

Registradores

- Internos a CPU
- Extremamente rápidos
- Otimizações de código podem mover temporariamente ´ variáveis para registradores.
- Programas podem dar palpites sobre o que deve ficar armazenado nos registradores

```
register int r;
```

- Veja o código register.c e o código assembly gerado sem otimização (register-O0.s) e com otimização (register-O2.s)

Lista encadeada

- Técnica com Listas Encadeadas

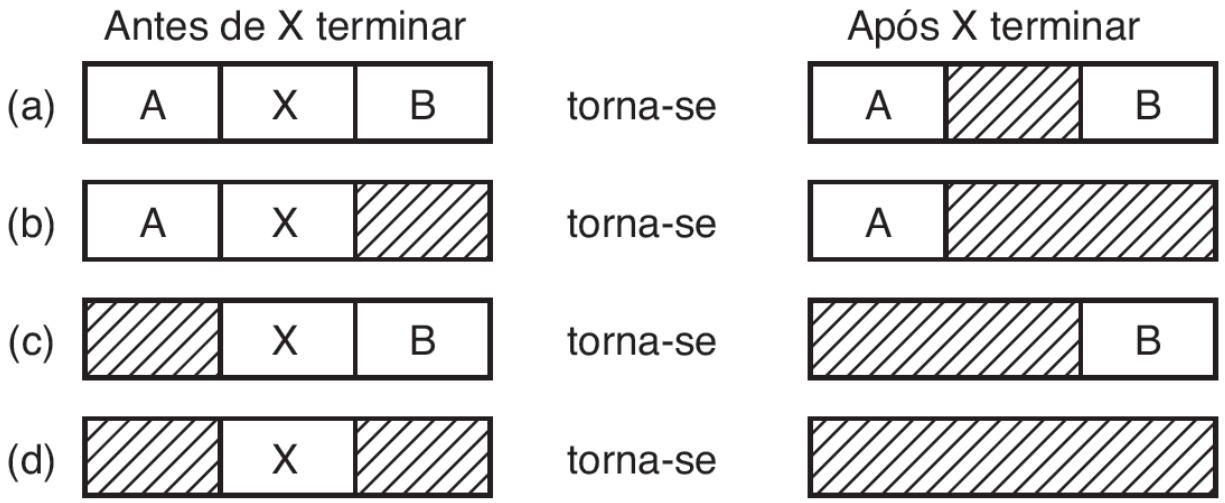

Figura 3.7 Quatro combinações de vizinhos para o processo que termina, *X*.

Veja os códigos: erro malloc.c erro calloc.c

Departamento de Computação

malloc, calloc e free

- São funções que permitem alocar e/ou liberar memória dinamicamente.
- A função *malloc( )* aloca *size* bytes e retorna um ponteiro para a memória alocada. A memória não é inicializada.
- A função *calloc( )* aloca um vetor de *nmemb* elementos, onde cada elemento tem *size* bytes, e retorna um ponteiro para a memória alocada. A memória alocada é inicializada com zeros.
- A função *free( )* libera o espaço de memória apontado por *ptr*.

Alocação de memória

- Existem três métodos que podem ser usados para selecionar uma região para um processo. Os algoritmos de alocação são:
  - *First fit*: ir sequencialmente através dos blocos, e tomar o primeiro grande o suficiente.
  - *Next fit*: proximo bloco grande o suficiente ´
  - *Best fit*: colocar o processo no bloco com o mínimo resto de memória;
  - *Worst fit:* usar o bloco com o maior resto de memória;

Departamento de Computação

Alocação de Memória

- Veja o código fit.c e descubra a política de alocação do malloc
  Blocos pequenos (< 64 bytes) =
  Blocos médios =
  Blocos grandes (> 512 bytes) =

Departamento de Computação

E o kmalloc?
Alocação de memória no kernel Linux

```
#include <linux/slab.h>
/* ... */
        int *s1 = kmalloc (size, flags);
        printk("KMALLOC: %p:\n", (void *) s1);
```