



Sistemas Operacionais

CCP/SIF

UNISUL – Tubarão

Cassio Brodbeck Caporal

`cassio{NOSPAM}ostec.com.br`

Agenda

- Revisão;
- Gerenciamento de memória;
- Alocação contígua simples;
- Overlay;
- Alocação particionada (estática e dinâmica);
- Alocação de partição (*{best, worst, first}-fit*);

Agenda

- Técnica de *swapping*;

Gerenciamento da Memória

- Uma das pérolas de sistemas operacionais:
 - Desenvolvimento focado em ocupar o mínimo possível de memória permitindo maior otimização dos recursos;
- Sistemas monoprogramáveis:
 - Implementação simplificada: um único programa executado por vez;

Gerenciamento da Memória

- Sistemas multiprogramáveis:
 - Implementação complexa: maximizar o uso eficiente da memória principal por diversas aplicações;
- Por que utilizar memória principal?
 - O processador só executa instruções que estão nesta memória;

Gerenciamento da Memória

- Por que utilizar memória principal?
 - Programas são armazenados em memórias secundárias, muito mais lentas, mas de maior armazenamento;
 - Diminuição de operações de E/S: o sistema operacional carrega o programa da memória secundária para principal, antes de executar!

Gerenciamento da Memória

- Desafios:
 - Manter na memória principal o maior número de processos;
 - Mesmo sem espaço livre, processos devem ser executados;
 - Garantir execução de programas que são maiores que a memória disponível;

Gerenciamento da Memória

- Desafios:
 - Proteção de acessos de memória:
 - Um processo não pode, sem permissão, interferir na área de memória de outro processo;
 - Compartilhamento de memória para que processos possam trocar informações de maneira protegida.

Alocação Contígua Simples

- Implementação pré-histórica, utilizada em sistemas monoprogramáveis;
- Subdivisão em duas áreas:
 - Sistema operacional;
 - Programa de usuário;
- O usuário possui controle sobre a memória principal (perigo!);

Memória principal

**Sistema
Operacional**

**Programas
de
Usuário**

Alocação Contígua Simples

- Ter acesso a memória do SO, não é legal:
 - Registradores para delimitação de áreas do SO e do Usuário;
 - ***Violação de acesso a memória!!!***
- Problemas?
 - Apenas um usuário utiliza a memória por vez;
 - Programas pequenos inutilizam espaço;

Windows

An exception 06 has occurred at 0028:C11B3ADC in VxD DiskTSD(03) + 00001660. This was called from 0028:C11B40C8 in VxD voltrack(04) + 00000000. It may be possible to continue normally.

- * Press any key to attempt to continue.
- * Press CTRL+ALT+RESET to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue

0x00000005,0x0016A950,0x00000001,0x00000005)
HANDLEP*** Address 0016a950 has base at 00100000

.6.2 irq1:1f SYSVER 0xf0000565

Name	Dll Base	DateStmp	Name
ntoskrnl.exe	80010000	33247f88	al.dll
atapi.sys	80007000	3324804	SIPORT.
Disk.sys	801db000	336015a	ASS2.SY
Ntfs.sys	80237000	344eeb4	lvid.sy
NTice.sys	f1f48000	31ec6e8d	loppy.SY
Cdrom.SYS	f228e000	31ec6e9	ull.SYS
KSecDD.SYS	f2290000	335a	.SYS
win32k.sys	fe0e2000	34	dll
Cdfs.SYS	fdca2000	3	
mbf.sys	fdc35000		ys
netbt.sys	f1f68000		.a
ard.sys	f2008000		.a
Parport.SYS	fdc14000		
acpi.sys	f1dd0000		

Overlay

- Permite dividir o programa em módulos para execução:
 - Na alocação contígua simples, os programas são limitados ao tamanho de memória disponível para o usuário;
- Cada módulo utiliza a mesma área de memória em momentos diferentes;

Alocação Particionada

- Estática:
 - Memória dividida em partições (tamanhos fixos);
 - Definido no momento do *boot* do SO;
 - Programas carregados e executados em uma partição específica, mesmo se outras estiverem livres;
 - Código absoluto;
 - *Alocação particionada estática absoluta.*

Alocação Particionada

- Estática:
 - Código relocável;
 - Permite que programas possam ser executados em qualquer partição;
 - Endereços do programa são relacionados ao código, não a endereços físicos da memória principal;
 - *Alocação particionada estática relocável;*

Alocação Particionada

- Estática:
 - *Alocação particionada estática relocável;*
 - Existência de uma tabela de partições:
 - Conhecimento acerca da alocação (uso) de partições;
 - Proteção de acesso a memória baseado em registradores de limite inferior e superior;

Alocação Particionada

- Estática:
 - Fragmentação interna:
 - Os programas não preenchem totalmente as partições para que são destinados;
 - Existência de fragmentos (áreas) livres;

Alocação Particionada

- Dinâmica:
 - Necessidade de aumentar o grau de compartilhamento da memória, acabando com a fragmentação interna;
 - O conceito de partição de tamanho *fixo* foi substituído por *variável*:
 - Um programa utilizará o que for necessário (esta será sua partição);

Alocação Particionada

- Dinâmica:
 - Surgimento de *outro* problema: fragmentação externa;
 - Programas terminam e deixam cada vez espaço menores impedindo que outros possam ser alocados;
 - Existência de “ilhas” de espaços livres intercalados com utilização de memória por programas “em execução”.

Alocação Particionada

- Dinâmica:
 - Soluções para fragmentação externa:
 - Quando um programa termina, somente os espaços adjacentes são liberados: áreas livres de tamanho maior;
 - Quando um programa termina, toda a memória é reorganizada, oferecendo um espaço livre contíguo;
 - Como sempre, tudo que é bom, tem seu custo!

Alocação de Partições

- Estratégia para alocação de partições:
 - Cada SO implementa seu método para definir qual área livre um programa será carregado;
 - Objetivo de diminuir a fragmentação externa;
 - Interferência direta do tipo de finalidade do SO;
- 03 modelos básicos: *best-fit*, *worst-fit*, *first-fit*;

Alocação de Partições

- *Best-fit*:
 - A *melhor* partição é escolhida:
 - Programa que deixa o menor espaço sem utilização;
 - Problema?
 - Probabilidade alta da memória ficar com pequenas áreas não-contíguas, aumentando a fragmentação;

Alocação de Partições

- *Worst-fit*:
 - A *pior* partição é escolhida:
 - Programa que deixar o maior espaço sem utilização;
 - Permite manter áreas maiores livres, garantindo maior número de programas executando;

Alocação de Partições

- *First-fit:*
 - Primeira partição livre, que tenha espaço para carregar o programa é escolhida;
 - A lista de áreas livres é ordenada crescentemente, havendo grande possibilidade de espaços livres em endereços altos;
 - É a que menos consome recursos do sistema;

Swapping

- E se não estiver partição disponível na memória principal?
 - O programa não será executado?
 - Não! O sistema operacional utilizará de técnicas de *swapping* (área de troca);
- Até o momento: programas ficam em memória do início ao término;

Swapping

- Um programa na memória em estado de espera não é bom!
- O *swapping* permite que programas não esperem para serem executados;
- O sistema escolhe um processo em memória e o transfere para memória secundária (operação de *swap out*);

Swapping

- A operação de *swap in* faz o inverso, carrega da memória secundária para principal;
- Qual processo escolher para fazer *swapping*?
 - Depende de critérios do SO, em geral programas com poucas chances de ter execução imediata, ou caracterizados como *I/O-bound*;

Swapping

- Resumindo:
 - O *swapping* permite maior grau de compartilhamento da memória principal;
 - Problema? Alto *overhead* nas operações de *swap in* e *swap out*;
 - Na existência de pouca memória principal, o *swapping* pode comprometer todo o SO.