

Gerenciamento de memória

Pedro Cruz

EEL770 – Sistemas Operacionais

Lembretes



- Trabalho
 - Dia 22 de setembro de 2017
 - Calendário Gregoriano
- James Kurose
 - Amanhã, CT2, 15h

Relembrando: processo na memória



Gerenciamento de memória



- Sistema de processo único
 - Memória dividida entre operacional e programa do usuário
- Sistema de multiprocessos
 - Memória dividida entre operacional e múltiplos programas de usuário

Requisitos do gerenciamento de memória



- Realocação
- Proteção
- Compartilhamento
- Organização lógica
- Organização física

- Garantir que informação entre e saia da memória
 - Processos
 - Código
 - Dados
- Garantir que referências dentro do programa estejam corretas

- Processos não podem interferir na memória dos outros
 - Leitura
 - Escrita
- Referências só são conhecidas em tempo de execução
 - Como o operacional pode garantir que não haja referências à memória que não é do processo?

- Processos não podem interferir na memória dos outros
 - Leitura
 - Escrita
- Referências só são conhecidas em tempo de execução
 - Como o operacional pode garantir que não haja referências à memória que não é do processo?
 - Operacional não pode garantir
 - Processador dá suporte a isso

Compartilhamento



- Diversos processos podem precisar acessar a mesma área
 - Processos com o mesmo código
 - Processos que compartilham dados

- Memória é um espaço linear de endereçamento
- Programas não são lineares
 - Módulos que se relacionam
 - Referências não podem ser decididas em tempo de compilação
 - Permissões de leitura e escrita não são uniformes
 - Módulos podem ser compartilhados por diversos processos

- Programas devem ser organizados na memória
 - Primária
 - “Rápida”
 - “Pequena”
 - Volátil
 - Secundária
 - “Lenta”
 - Grande
 - Permanente
- Programadores não sabem a memória disponível em tempo de execução
- Operacional deveria organizar os programas nas memórias

Sem abstração de memória



- Processo único em execução
- Processo único em memória
 - Outros em disco
- Chaves de memória
 - Memória é dividida em blocos
 - Cada um tem uma chave
 - Processador só permite acesso a um bloco com a chave
 - Processos são escritos para começar no endereço 0x00
 - Realocação estática
 - Somar o endereço inicial do programa a cada referência de memória realizada no programa em tempo de carga

- IBM 360
 - *Mainframe*
- Sistemas embarcados
- Internet das Coisas

Abstrações de memória



- Espaços de endereçamento
- Memória virtual

- Representa uma fatia independente de memória
 - Cada processo recebe uma fatia
- Implementado na forma de registradores (realocação dinâmica)
 - Registrador base
 - Registrador limite
 - Referência do processo sofre alterações e verificações em tempo de execução
 - Somada ao registrador base
 - Comparada com registrador limite

Problema da abordagem?

- Processos nem sempre cabem na memória
 - Limitar o uso de memória
 - Executar um número limitado de processos
 - Matar processos que precisarem de memória indisponível
 - Elaborar estratégias para “criar memória”
 - Troca de processos – *Swapping*
 - Memória virtual

Troca de processos – *Swapping*



- Processo fica completamente armazenado em disco
- Processo é trazido de forma completa quando é acordado

Expansão da alocação

- Processo pode precisar de mais memória
- Expansão pode se dar de duas maneiras



- Processos entram e saem da memória
 - Processos recém-criados precisam entrar na memória
 - Vamos assumir que processos são contíguos
- Operacional deve manter controle sobre o que pode ser alocado e o que já está alocado
 - Mapas de bits
 - Listas livres

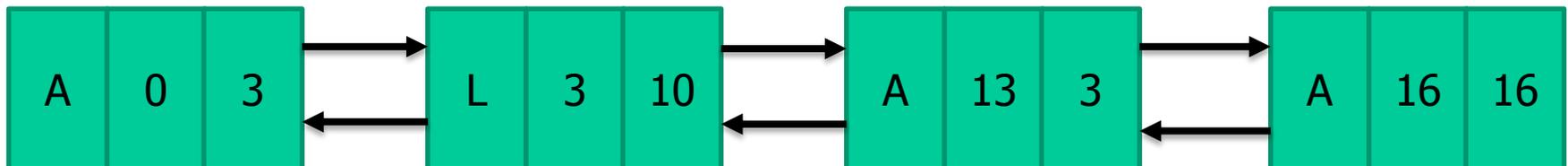
Mapa de bits



- Memória é dividida em palavras
 - k bytes para cada
- Cada palavra é mapeada em um bit
 - 0 se memória está livre
 - 1 se memória está ocupada
- Mapa de bits ocupa $1/k$ da memória
- Alocação de n palavras deve procurar por n 0's contíguos

Lista encadeada

- Lista encadeada guarda posições livres da memória
 - Cada elemento da lista representa um segmento contíguo de memória
 - Localização
 - Início
 - Tamanho
 - Estado
 - Livre
 - Ocupado



- Processos recém-criados precisam ir pra memória
 - Onde?
- Algoritmos de alocação
 - Primeiro encaixe (*first fit*)
 - Gerenciador procura o primeiro espaço livre mínimo
 - Melhor encaixe (*best fit*)
 - Gerenciador procura o menor espaço livre mínimo
 - Pior encaixe (*worst fit*)
 - Gerenciador procura o maior espaço livre

Problemas dos algoritmos de alocação



- Primeiro encaixe (*first fit*)
 - Pode deixar espaços livres na memória
- Melhor encaixe (*best fit*)
 - Deixa espaços pequenos livres na memória
 - Inúteis
 - Demora pra achar o melhor
- Pior encaixe (*worst fit*)
 - Não funciona tão bem quanto promete

Gerenciamento de memória

Pedro Cruz

EEL770 – Sistemas Operacionais