Gerenciamento de Memória Mach

MAC 5755 - Sistemas Operacionais Distribuídos
(novembro 2001)

Rodrigo S. de Castro
Mestrando DCC
rcastro@ime.usp.br

Mach

Por que estudá-lo?

O Projeto Mach

- School of Computer Science - Carnegie Mellon University
- Período: 1985 a 1994
- 31 membros (3 atualmente)
- Não há mais desenvolvimento
- Objetivos de Pesquisa Atingidos
- Última atualização da página: 1997

Objetivos

- base para construção de outros SOs
- espaços de endereçamento grandes e esparsos
- acesso transparente a recursos de rede
- explorar paralelismo para o sistema e aplicações
- portabilidade para uma grande quantidade de máquinas

Gerenciamento de Memória do Mach

Motivação

- Limitações da Hierarquia de Memória do 4.3BSD
  - influenciado pelo VAX
  - dificuldade de portar
  - restrito ao suporte à paginação
  - não pode ser estendido a ambientes distribuídos
  - compartilhamento de memória somente-leitura

Características

- Portável
- Compatível com o UNIX 4.3BSD (ABI)
- Suporta:
  - espaços de endereçamento grandes e esparsos
  - operações de cópia virtual "copy-on-write"
  - compartilhamento de memória ("copy-on-write" e leitura-escrita) entre tarefas
  - mapeamento de arquivos na memória
  - "pagers" e objetos de armazenamento do usuário

Abstrações Básicas do Mach

Tarefa
- ambiente no qual as threads rodam
- espaço de endereçamento virtual paginado
- acesso protegido aos recursos do sistema

Thread
- unidade básica da utilização da CPU
- compartilham recursos da tarefa
- processo = tarefa + uma thread

Port
- fila protegida de mensagens

Mensagem
- coleção de dados com tipo

Objeto de Memória

Operações do VM

Alocação de Memória
- vm_allocate (páginas em branco)
- vm_map (página com um objeto de memória)
- vm_deallocate

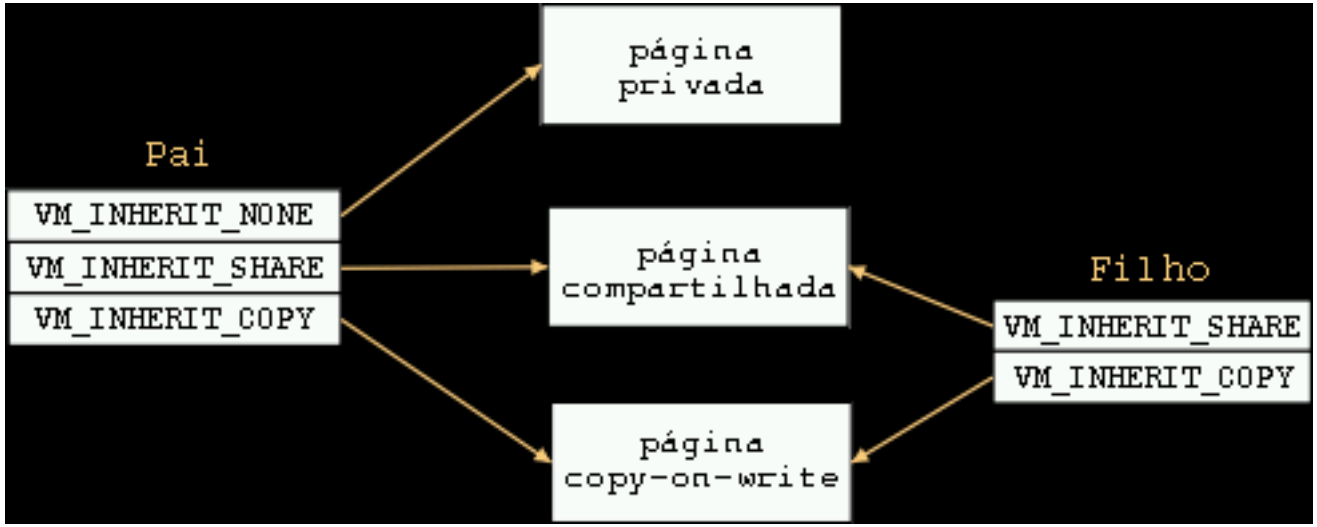
Proteção
- leitura, escrita e execução por página
- depende do hardware (ex. bit de execução)
- proteção atual e máxima
- vm_protect

Herança
- VM_INHERIT_NONE, VM_INHERIT_SHARE, VM_INHERIT_COPY

Outros
- vm_read, vm_write (páginas de outras tarefas)
- vm_regions, vm_statistics

Herança

Herança através do task_create



Multíplos Modos de Compartilhamento

Estruturas de Dados do MM

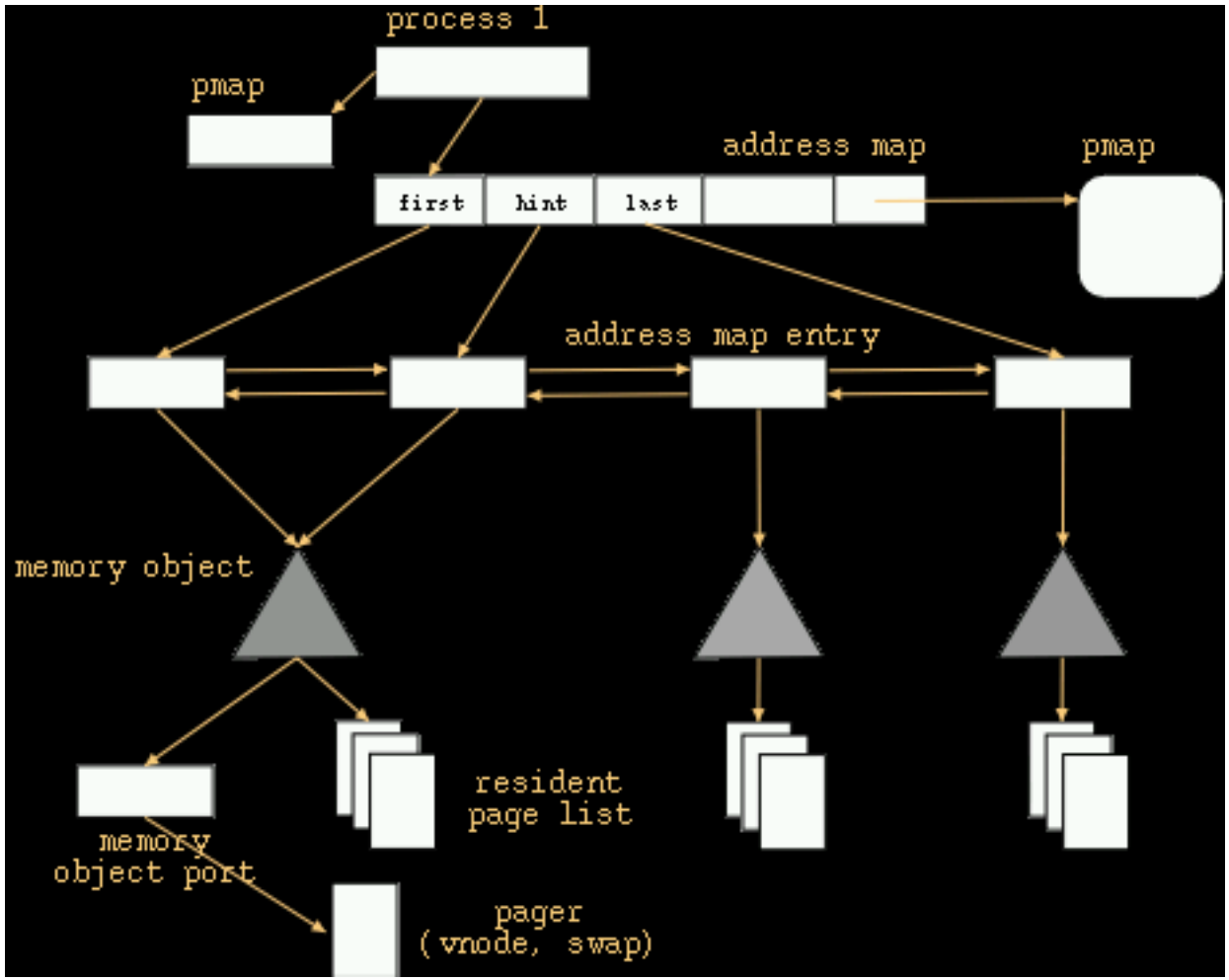
tabela de páginas residente
- informação sobre as páginas físicas

mapa de endereçamento
- lista de entradas, cada uma descreve o mapeamento de uma faixa de endereços para uma região de um objeto de memória
- herança e proteção por região de memória
- mudança de proteção implica na divisão em duas regiões

objeto de memória
- objeto abstrato representando uma coleção de bytes de dados para os quais diversas operações, como leitura e escrita, são definidas (por um pager)

pmap
- mapa de tradução de endereços virtuais-físico
- supõe apenas que exista um MMU simples e paginado

Estruturas de Dados do MM



Memória Residente

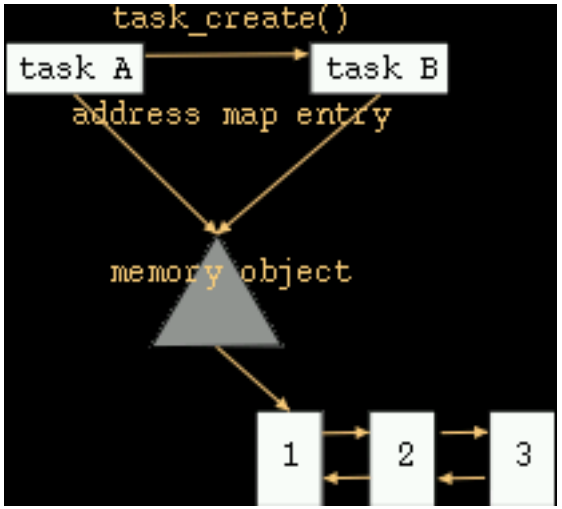
Cada página da Tabela de Páginas Residente pode ser mantida em:

- lista de objetos de memória
  - tem todas as páginas de um objeto para tornar mais rápida a desalocação e operações de cópia virtual
- filas de alocação de memória
  - usadas pelo demônio da paginação
  - existem três filas: ativa, inativa e livre
- tabela de hash objeto/offset
  - para procura rápida de uma página na memória

Página Física no Mach

- potência de 2 múltipla do tamanho dependente da máquina
- parâmetro de boot

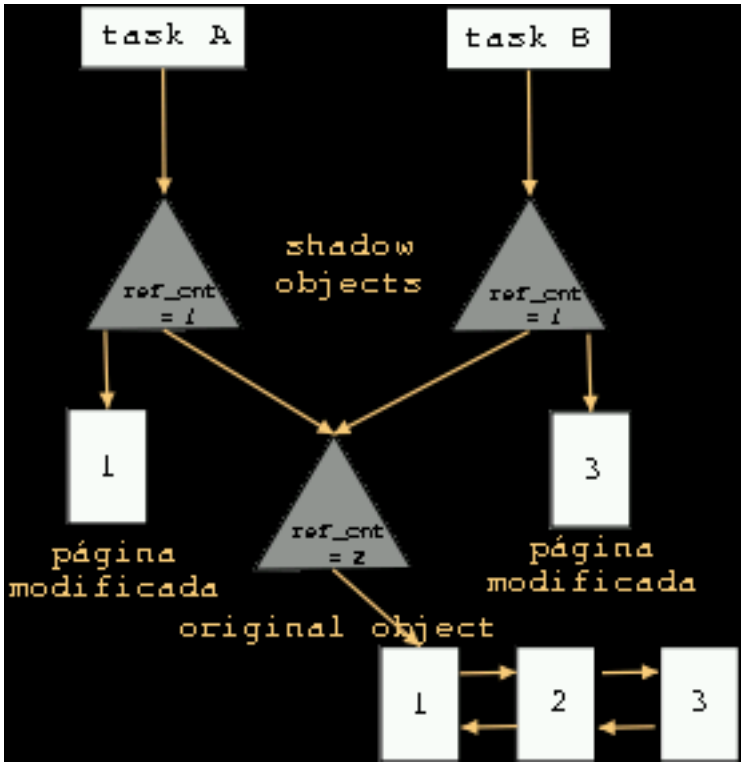
Herança Copy-On-Write



- tarefa A cria tarefa B
- proteção é definida como somente-leitura
- contador de referências é aumentado de 1 para 2

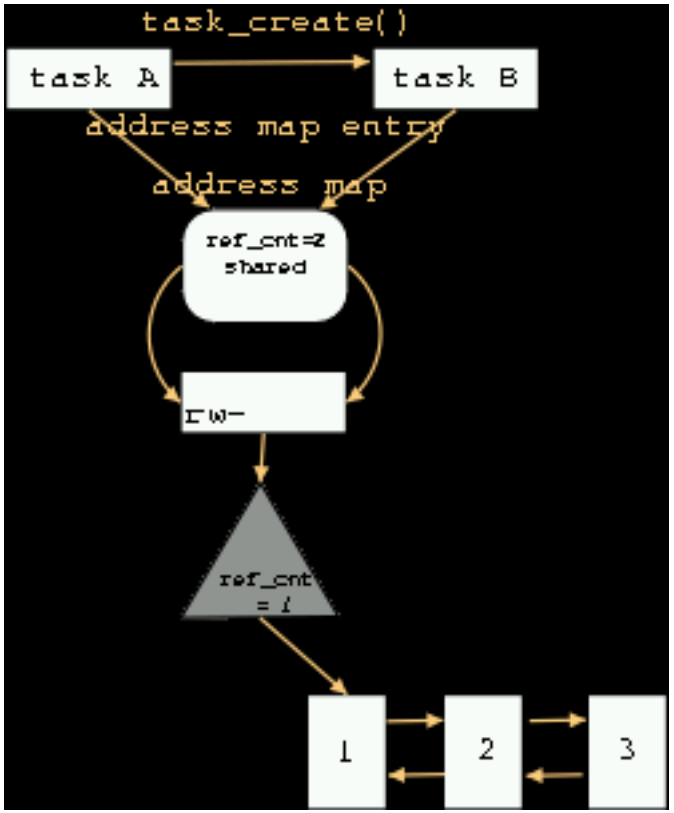
- um dos usos é para grandes transferências de mensagens
- colapsa os shadow objects quando possível

Shadow Objects



- se qualquer uma delas tenta escrever, gera-se um page-fault
- cada tarefa obtém um shadow object
- o shadow object gerencia a(s) página(s) modificadas

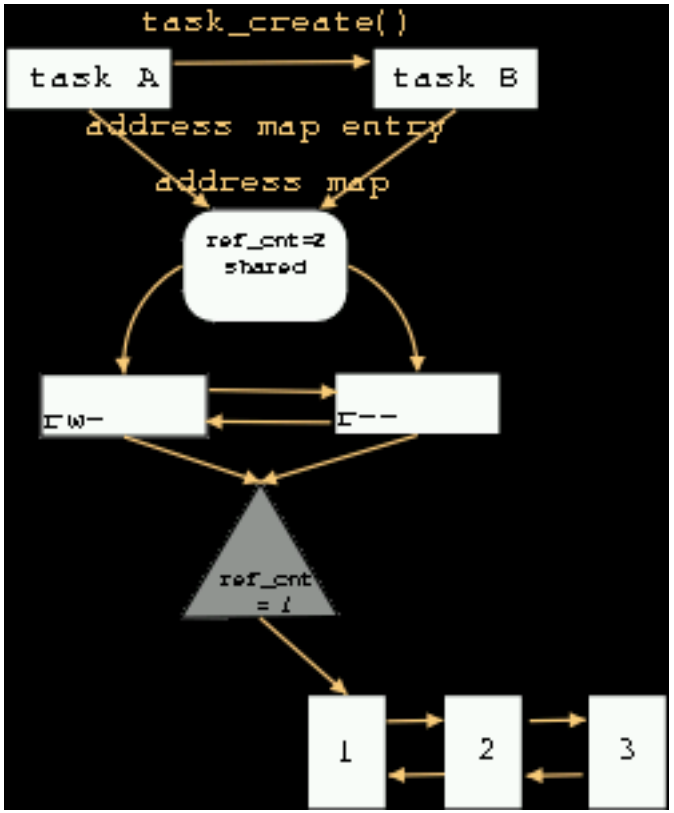
Compartilhamento Leitura-Escrita



- mudanças de proteções e outros atributos: share map

Compartilhamento de Memória

Dividindo uma região compartilhada

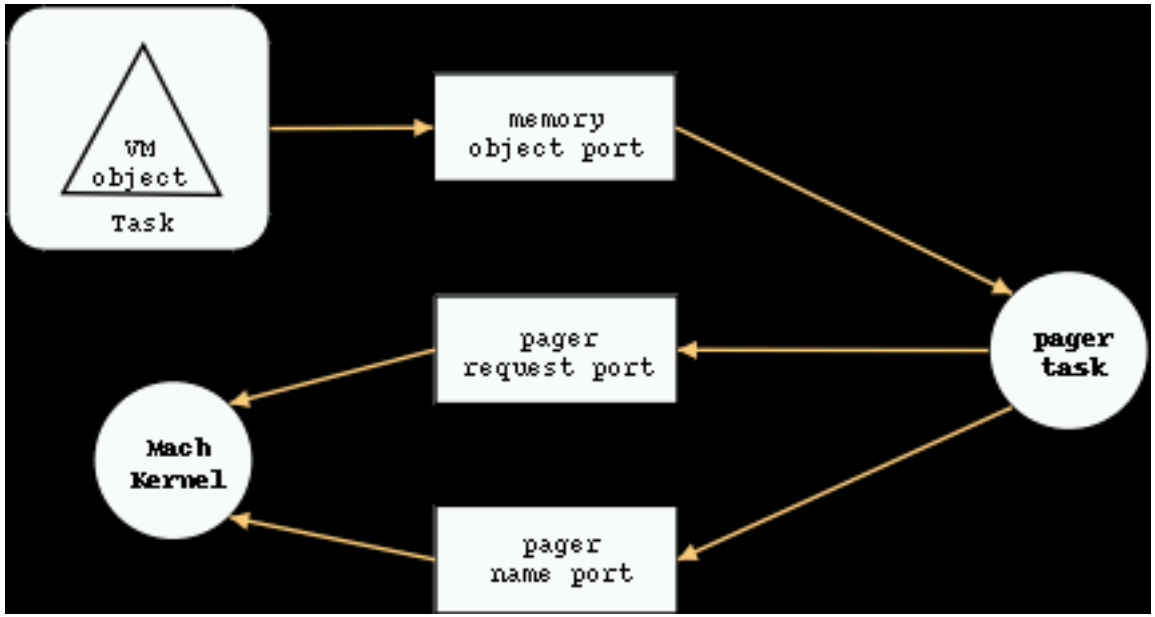


Objetos de Memória e Pagers

Pager

- Tarefa que gerencia um ou mais objetos de memória
- Interage com o kernel para movimentar dados entre o objeto e a memória fí sica

Comunicação com o Pager



Interações Kernel-Pager

Page faults

- Quando trata uma "page fault", o kernel verifica o mapa de endereçamento para localizar o objeto da VM apropriado que contém uma referência para o objeto de memória que pode forncer os dados requisitados.

Pageouts

- Quando o kernel decide escrever uma página suja, ele a envia para o pager.

Proteção

- O kernel pode requisitar permissões adicionais para a página. O pager pode, por sua vez, pedir para se baixar as permissões de acesso.

Gerenciamento de Cache do Kernel

Pagers Externos e Internos

Mach provê um pager default
- objetos sem armazenamento permanente
- shadow objects e regiões zero-fill (pilha, heap, bss)

Pager pode ser remoto

Pagers implementados como tarefas do usuário
- pagers externos
- servidor de memória compartilhado pela rede
  - permite que um conjunto de páginas seja compartilhado pela rede
  - provê dados para os clientes quando ocorre page-fault
  - sincroniza acessos aos dados compartilhados

Algoritmo de Substituição de Páginas

Algoritmo: FIFO with second chance

- não provê políticas de substituição alternativas
- extensão da interface de pager para tarefas de usuário terem a sua própria política

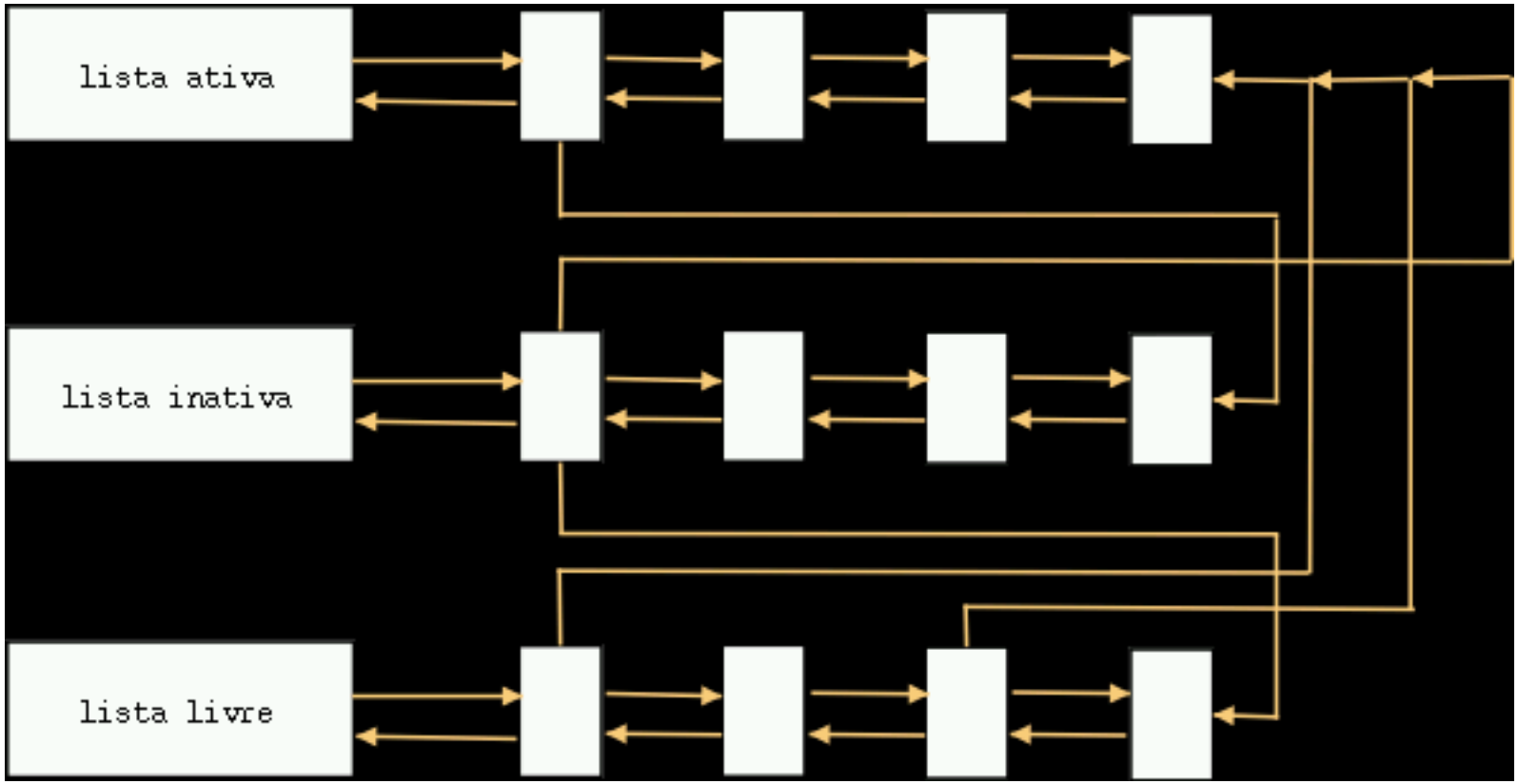
Diferente do SVR4 e do 4.3BSD

Listas (Ativa, Inativa e Livre)

- Livre ▷ Ativa: page fault
- Ativa ▷ Inativa: memória livre abaixo de um limiar
- Inativa ▷ Ativa: o demônio verifica as páginas no começo da lista em relação aos bits de referência
- Inativa ▷ Livre: se os bits de referências das páginas no começo da lista estão desligados
- Livre ▷ Ativa: se a página for referenciada enquanto estiver na livre, ela pode ser reclamada.

Algoritmo de Substituição de Páginas

Listas



Parâmetros do Demônio da Paginação

vm_page_free_min
- acorda o demônio se o número de páginas na lista livre ficar abaixo desse limite

vm_page_free_target
- move páginas da lista das inativas para a lista das livres até que a lista das livres atinja esse tamanho

vm_page_free_reserved
- número de páginas livres reservadas para o demônio da paginação

vm_page_inactive_target
- move páginas da lista das ativas para a lista das inativas até que a lista das inativas atinja esse tamanho

Portabilidade

Seções

- Dependente de Arquitetura
  - operações relativas às estruturas de dados de mapeamento
  - mapeamentos cruciais para execução do sistema
  - pode fazer coleta de lixo para informações de mapeamento não importantes
  - ex: mapa do kernel e mapeamento para os endereços de tarefas ref. frequentemente
  - não tem conhecimento das estruturas de dados da parte independente
  - pmap.c e pmap.h
- Independente de Arquitetura
  - toda informação importante da memória virtual

Arquiteturas Portadas

- VAX, IBM RT PC, SUN 3, Encore MultiMAX e o SequentBalance

Análise

Arquitetura de VM bem desenhado

Vários recursos avançados

Muito similar ao SVR4 como compartilhamento "copy-on-write", acesso a arquivos mapeados na memória e suporte a espaços de endereçamento grandes e esparsos

Abordagem orientada a objetos

Separação das partes do código dependente e independente de arquitetura

Facilidades de Compartilhamento de Memória, com separação das noções de compartilhamento e herança

Alta integração do VM com o IPC

Trabalhos em Cima do Mach

Gerenciador de Memória Padrão

- "Moving the Default Memory Manager out of the Mach Kernel"
  - David B. Golub e Richard P. Draves
  - CMU

Gerenciamento de Memória Distribuída

- "A New Approach to Distributed Memory Management in the Mach Microkernel"
  - Stephan Zeisset et al.
  - Intel GmbH

Referências

Machine-Independent Virtual Memory Management for Paged Uniprocessor and Multiprocessor Architectures

- Richard Rashid et al.

UNIX Internals

- Uresh Vahalia

Modern Operating Systems

- Andrew S. Tanenbaum

The Mach Project Home Page

http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/mach/public/www/mach.html