Gerência de Memória Virtual

Prof. Dr. Antonio Carlos Sementille

Gerência de Memória

- **POLÍTICA DE SUBSTITUIÇÃO DE PÁGINAS**
- **Memória principal é limitada: necessidade de substituição de páginas.**
- **Escolha de substituição randômica: solução insatisfatória**
- **Alguns sistemas: predição de quais são as páginas menos úteis**

Gerência de Memória

Algorítmo de Substituição de Página ótimo

- **Melhor dos algoritmos, fácil de descrever, mas impossível de implementar.**
  - **No momento da falta de páginas: existe um conjunto de páginas na memória**
  - **Uma delas será referenciada na próxima instrução (página que contém a instrução que gerou a falta de página). As outras só serão referenciadas mais tarde.**
  - **Cada página pode ser rotulada com o número de instruções que serão executadas antes de aquela página ser referenciada pela primeira vez.**
  - **Idéia do algoritmo: remover a página com maior rótulo**

Gerência de Memória

Algorítmo NUR (Não usada recentemente)

- **Substituição da página usada a mais tempo.**
- **Tabela de páginas: cada entrada contém os bits M (modificada) e R (referenciada).**
- **Bit R: colocado em 1 sempre que a página é lida ou escrita**
- **Bit M: colocado em 1 sempre que se escreve na página**
- **Esses bits são atualizados a cada referência a memória**
- **Periodicamente o bit R é limpo (a cada tique do relógio), para diferenciar as páginas que foram referenciadas recentemente, das que não foram.**

Gerência de Memória

Algorítmo NUR (Não usada recentemente)

Quando acontece uma falta de páginas, o S.O. Inspeciona todas as páginas e as separa em 4 categorias:

- **Classe 0: não referenciada, não modificada**
- **Classe 1: não referenciada, modificada**
- **Classe 2: referenciada, não modificada**
- **Classe 3: referenciada, modificada**

O algoritmo NUR remove aleatoriamente uma página de ordem mais baixa que não esteja vazia

Gerência de Memória

Algoritmo FIFO (First In - First Out)

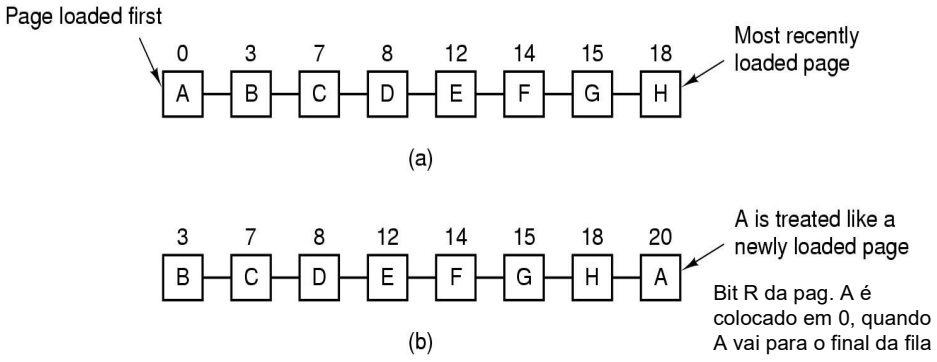
- **O algoritmo de Substituição da página Primeiro a entrar, primeira a sair é um de baixo custo.**
- **Fácil Implementação: O S.O. mantém uma lista de todas as páginas atuais na memória, com a página mais antiga na cabeça da lista e a que chegou mais recentemente no final da lista.**
- **Na ocorrência de uma falta de página: a 1ª. Página da lista é removida e a nova é adicionada no final da lista**
- **Desvantagem: página na memória carregada a mais tempo pode estar ainda sendo referenciada**

Gerência de Memória

Algoritmo da Segunda Chance

- **Consiste em uma pequena mudança no algoritmo FIFO, para evitar que páginas mais usadas sejam descartadas: observa-se o bit de *referenciada* (na tabela de páginas).**
- **A página escolhida deve ter seu bit de referenciada igual a zero.**

Algoritmo da Segunda Chance



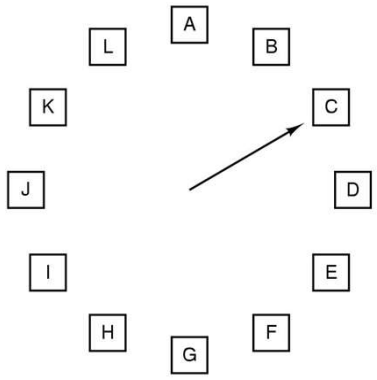
- Operação de uma Segunda Chance
  - Páginas classificadas na ordem FIFO
  - Situação da lista de páginas quando de uma falta de página no instante 20, com o bit R da página *A* em 1 (os números acima das páginas são os instantes de carregamento)

Algoritmo de Substituição de Página do *Relógio*

-Algoritmo da Segunda Chance:
Fica reinserindo páginas no final da lista

-Estratégia melhor: manter todas as páginas em uma lista circular em forma de relógio
-Um ponteiro aponta a página mais Antiga (“cabeça” da lista)

- Quando ocorre uma falta de página, a página apontada é examinada.
- A atitude a ser tomada depende do bit R:
-Se R=0, retira a página
-Se R=1: faz R=0 e avança o ponteiro



9

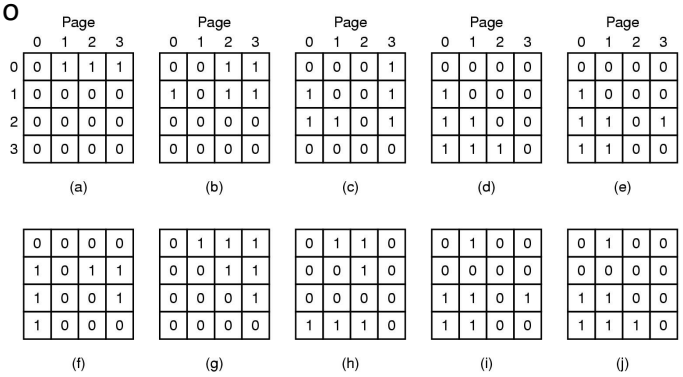
Algorítmo MRU (Menos Recentemente Usada)

- Descarta a página que a mais tempo não é usada.
- Implementação onerosa: manter uma lista encadeada de todas as páginas na memória, com a página mais recentemente usada no início da lista e a página menos recentemente usada no final da lista

- **Desvantagem:** a lista encadeada deve ser atualizada a cada referência à memória

Algoritmo MRU com uso de hardware especial

- Para máquina com n molduras de páginas: matriz $n_x n$ bits
- Sempre que uma moldura k for referenciada: linha k preenchida com 1s e coluna k preenchida com zeros
- Escolhida para sair aquela página cuja linha tiver menor valor binário

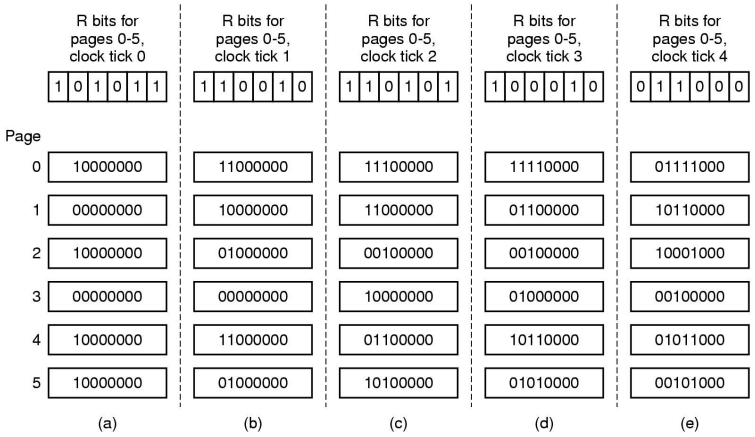


LRU using a matrix – pages referenced in order 0,1,2,3,2,1,0,3,2,3

11

Simulando MRU em software

- Requer contadores em software associados a cada página
- Os contadores tem número finito de bits



- O algoritmo do aging (envelhecimento) simula o MRU em software
- São mostradas 6 páginas para 5 interrupções do relógio (a) – (e)

12

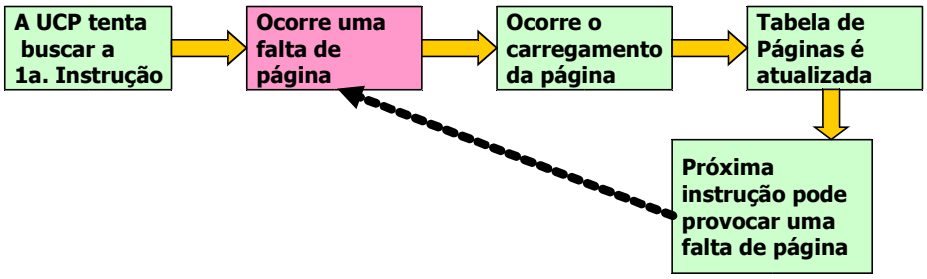
Algoritmo de Substituição de Página do Conjunto de Trabalho

- Referência a um endereço de uma página que não está na memória principal: falta de página (page fault)

- Após a falta de página, o S.O.:
  - Lê a página requerida da memória secundária;
  - Coloca sua nova posição de memória física, na tabela de páginas;
  - Repete a instrução que executou a falta.

- É possível iniciar um programa, mesmo que nenhuma página esteja na memória principal: tabela de páginas deve indicar que todas as páginas estão na memória secundária.

- Isto é chamado de paginação por demanda.



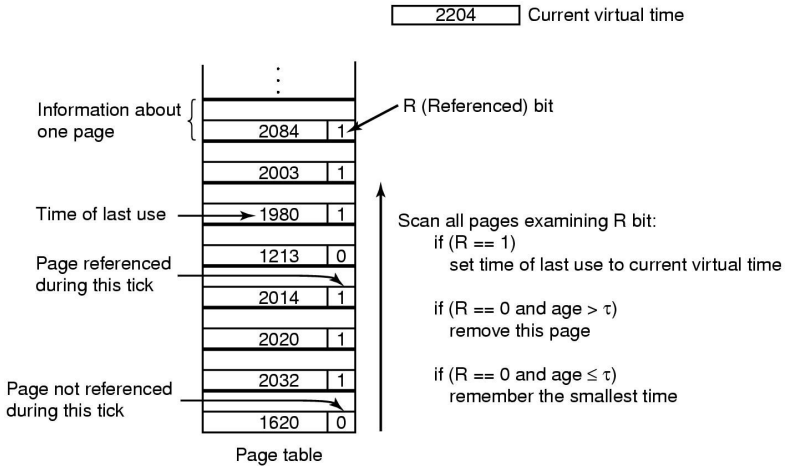
- **Questão da demanda: só relevante na iniciação do sistema.**
- **Sistemas Multiprogramados: problema crítico, devido à comutação constante de processos.**
- **Abordagem alternativa: característica da "localidade de referência" permite antecipação das páginas mais usadas.**

Gerência de Memória

- **Conjunto de Trabalho: tem-se que, em qualquer instante de tempo t, existe um conjunto que consiste de todas as páginas usadas nas k referências mais recentes à memória**

W(k,t) representa o tamanho do conjunto de trabalho no instante t

Idéia principal: encontrar uma página que não esteja no conjunto de trabalho e removê-la da memória



Variação lenta do conjunto de trabalho: antecipação das páginas baseada na última execução.

Vantagem: economia de tempo.

Desvantagem: se conjunto de trabalho não estiver estabilizado haverá perda do tempo de carregamento.

Gerência de Memória

- **Um programa que gera falta de páginas freqüente e continuamente é dito provocar "trashing".**
- **Isto acontece quando o número de páginas físicas é inferior ao tamanho do conjunto de trabalho.**

Algoritmo de Substituição de Página do WSClock

- O algoritmo básico do conjunto de trabalho é enfadonho
- WSClock: algoritmo melhorado baseado no relógio

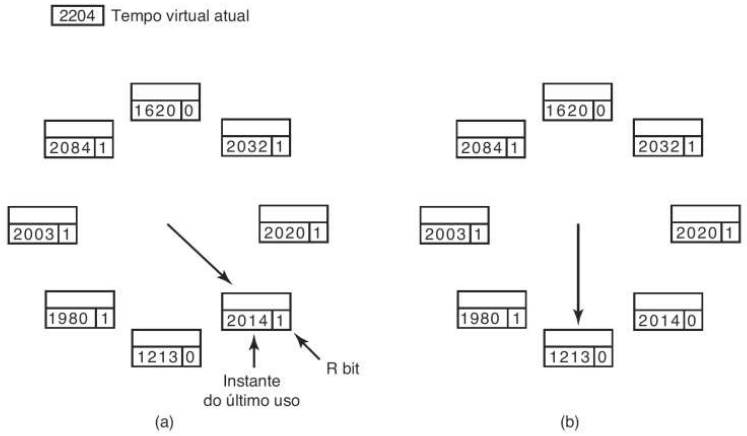
- **Funcionamento:**

- Lista circular de molduras de página: inicialmente a lista está vazia
- A medida que as páginas são carregadas, elas são inseridas na lista, formando um anel: cada entrada contém: *instante da última referência*, bit R e bit M

Algoritmo de Substituição de Página do WSClock

- Cada página apontada é examinada, se R=1, foi referenciada durante interrupção atual do relógio e não será removida, o bit R é colocado em zero e o ponteiro avança
- Se R=0 ,sua idade for maior que *t* e estiver limpa: não está no conjunto de trabalho e há cópia válida no disco: remove a página (entra uma nova)
- Se a página estiver suja: não será reinvidicada imediatamente, pois não há cópia válida no disco
- Para evitar a troca do processo: a escrita a disco é escalonada, mas o ponteiro é avançado e o algoritmo continua com a próxima página
- A final de contas, pode haver uma página velha e limpa mais adiante, apta a ser usada imediatamente

21



(a) e (b) exemplificam o que acontece quando R=1 (continua)

22

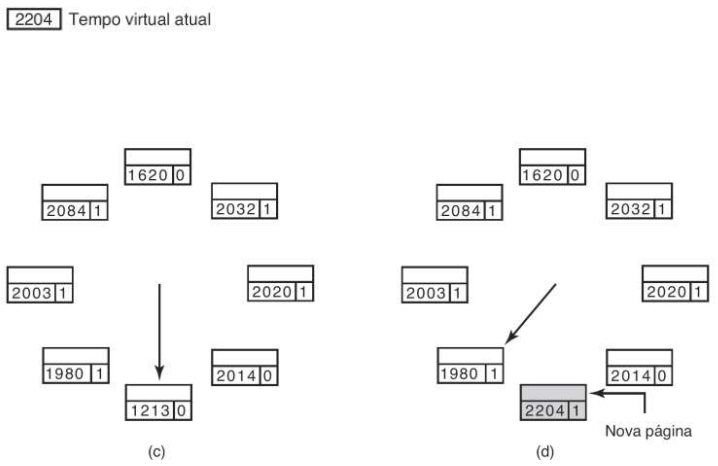


Figura 3.20 Funcionamento do algoritmo WSClock. (a) e (b) exemplificam o que acontece quando $R = 1$. (c) e (d) exemplificam a situação $R = 0$.

(c) e (d) exemplificam o que acontece quando R=0

23

Resumo dos algoritmos de substituição de páginas

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NRU (não usada recentemente)	Aproximação muito rudimentar do LRU
FIFO (primeiro a entrar, primeiro a sair)	Pode descartar páginas importantes
Segunda chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
LRU (usada menos recentemente)	Excelente algoritmo, porém difícil de ser implementado de maneira exata
NFU (não frequentemente usada)	Aproximação bastante rudimentar do LRU
Envelhecimento (*aging*)	Algoritmo eficiente que aproxima bem o LRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente

Tabela 3.2 Algoritmos de substituição de página discutidos no texto.

Questões de Projeto para Sistemas de Paginação

A) Política de Alocação Global versus Alocação Local

	Age
A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6

(a)

A0
A1
A2
A3
A4
A6
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

(b)

A0
A1
A2
A3
A4
A5
B0
B1
B2
A6
B4
B5
B6
C1
C2
C3

(c)

(a) Configuração original (b) Substituição Local (c) Substituição Global

- Geralmente, os algoritmos globais funcionam melhor, especialmente quando o conjunto de trabalho varia durante o tempo de vida de um processo

25

B) Controle de Carga

- A despeito de bons projetos, o sistema ainda pode gerar thrash
- Pois pode acontecer de:
  - Alguns processos necessitarem de mais memória
  - Mas nenhum processo necessita menos
- Solução :
  Reduzir o número de processos competindo por memória
  - trocar (swap) um ou mais para o disco, liberando-se a memória por eles alocadas
  - Reconsiderar o grau de multiprogramação

26

C) Tamanho da Página(1)

Página de tamanho pequeno

- **Vantagens**
  - Diminui a fragmentação interna
  - Alocação melhor para várias estruturas de dados e seções de código
  - Menos partes não usadas de um programa na memória
- **Desvantagens**
  - Os programas precisarão de mais páginas, e portanto, serão maiores as tabelas de páginas

27

C) Tamanho da Página (2)

- Sobrecarga (overhead) da tabela de página e da fragmentação interna

$$overhead = \frac{s \cdot e}{p} + \frac{p}{2}$$

page table space

internal fragmentation

- Onde
  - s = tamanho médio do processo em bytes
  - p = tamanho da página em bytes
  - e = tamanho de uma entrada na tabela de páginas

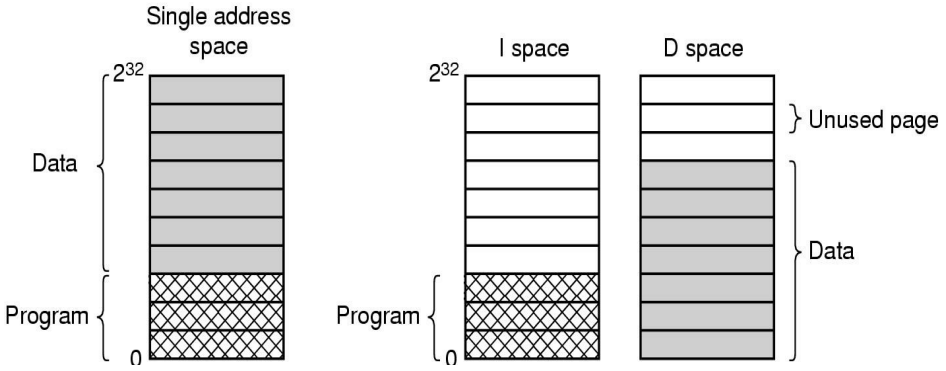
Valores típicos: 4KB ou 8KB

Otimizado quando

$$p = \sqrt{2se}$$

28

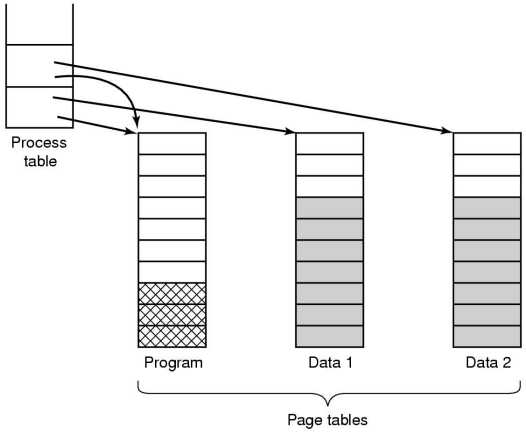
D) Separar os espaços de Instruções e Dados



- Espaço de endereçamento único
- Espaços separados de instruções (I) e Dados (D): cada um pode ser paginado (tendo suas próprias tabelas de páginas)

29

E) Páginas Compartilhadas



Dois processos compartilhando o mesmo código de programa, e conseqüentemente, a mesma tabela de páginas para as instruções

30

F) Política de Limpeza

- A paginação funciona melhor quando existe uma grande quantidade de molduras de páginas disponíveis
- Necessidade de um processo em background, o **daemon de paginação**
  - Ele periodicamente inspeciona o estado da memória
- Quando poucas molduras estão disponíveis
  - Seleciona páginas para serem removidas da memória, usando um algoritmo de substituição

31

Questões de Implementação

O envolvimento do Sistema Operacional com a paginação

Existem 4 circuntâncias em que o S.O. se envolve com a paginação

1. **Criação do Processo**
   - Determinação do tamanho do programa
   - Criação da tabela de páginas
2. **Execução do processo**
   - MMU é reinicializada para um novo processo
   - TLB é esvaziada
3. **Quando ocorre uma Falta de Página**
   - Determina o endereço virtual que causou a falta
   - Faz *swap out* da página que deve sair e *swap in* da que deve entrar
4. **Quando um processo termina**
   - Libera a tabela de páginas, suas páginas e o espaço em disco que ocupam

32

Tratamento de Faltas de Página (1)

1. Hardware gera uma interrupção (trap) e desvia a execução para o núcleo (kernel)
2. Os registradores de uso geral são salvos
3. O S.O. determina qual página virtual é necessária
4. O S.O. verifica a validade do endereço e procura uma moldura de página
5. Se a moldura selecionada está suja, escreve-a no disco

33

Tratamento de Faltas de Página (2)

6. O S.O traz a nova página do disco
7. As tabelas de páginas são atualizadas
   - A instrução causadora da falta é recuperada para o estado que estava
8. O processo causador da falta é escalonado para execução
9. Os registradores são restaurados
   - O programa continua

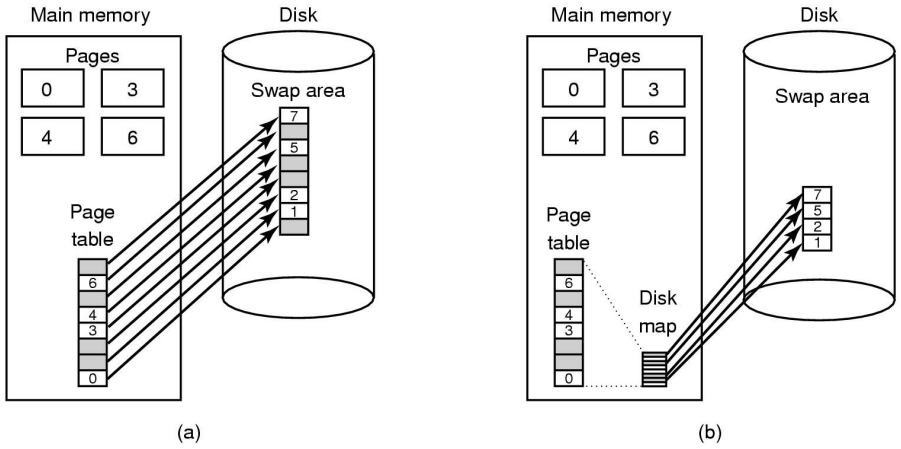
34

Fixação de Páginas na Memória

- Memória virtual e E/S interagem ocasionalmente
- Um processo chama o sistema para ler algum dispositivo para um buffer de seu espaço de endereçamento
  - Enquanto está esperando por E/S, um outro processo entra em execução
  - Este outro processo causa uma falta de página
  - O buffer do primeiro processo pode ser escolhido para ser retirado (paged out)
- Pode ser necessário que algumas páginas sejam fixadas (bloqueadas)

35

Memória Secundária



(a) Paginação para uma área de troca estática

(b) Paginação com páginas alocadas dinamicamente em disco

36

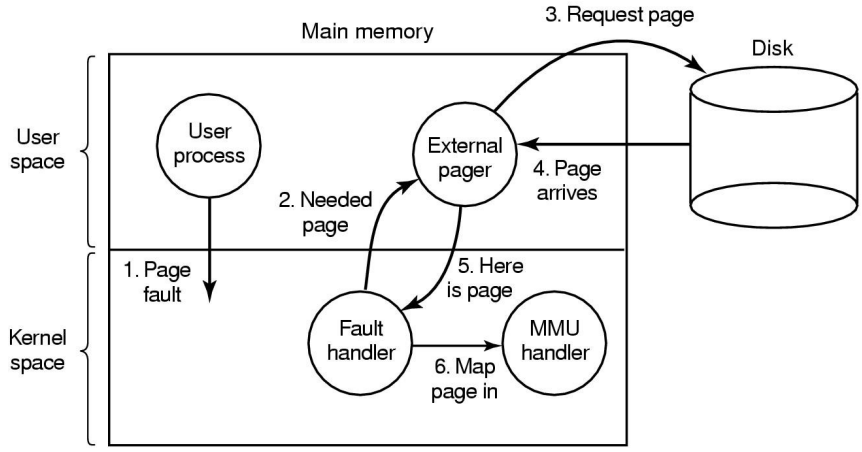
Separação da Política e do Mecanismo

- Para separar a política do mecanismo: fazer com que muitos dos gerenciadores de memória sejam executados como processos no nível do usuário

- Um exemplo é o que faz o S.O. Mach:
- Nele o gerenciamento de memória é separado em 3 partes:
  1. Um tratador de MMU (*MMU handler*) de baixo nível
  2. Um tratador de falta de página (*fault handler*) que faz parte do núcleo
  3. Um paginador externo (*extern Pager*) executado no espaço do usuário

37

Separação da Política e do Mecanismo



Tratamento de faltas de páginas com paginador externo

38