

Vermelho - não fiz/conferi

preto - galucio deu ponto máximo na questão/resposta resumida a uma que o galucio deu ponto máximo

azul - acho que tá certo

rosa - acho que está errado

1. Explique como pode ser feito para que a alocação local de páginas físicas aloque um número “correto” de páginas. Como é possível saber se o número alocado está superior ou inferior ao necessário?

Através do Working set (conjunto de páginas lógicas que um processo utiliza durante o intervalo de tempo observado), se o número de páginas alocadas for maior que o tamanho do working set haverá poucas faltas de páginas e inicialmente o SO não faz nada. Se o número de páginas alocadas for igual ao tamanho do working set haverá poucas faltas de páginas e o SO não fará nada, essa é a situação ideal. E se o número de páginas alocadas for menor que o tamanho do working set vai ter muitas faltas de páginas e o SO vai alocar mais páginas a esse processo.

2. Precisa do bit acessado para saber se a página foi acessada recentemente. Os algoritmos semelhantes ao LRU utilizam essa informação para decidir quais páginas são candidatas a serem retiradas da memória física. O SO dá prioridade às páginas que foram utilizadas recentemente, pois há uma chance maior dessas páginas serem solicitadas novamente pelo processo num futuro próximo.

Mostre 2 razões que justificam que a segmentação sob demanda seja pior que a paginação sob demanda.

Na segmentação o tamanho dos segmentos são diferentes enquanto na paginação todas as páginas tem tamanho iguais, ao levar um segmento para o disco se eu precisar levar ele de volta pra memória pode não ter uma área livre do tamanho certo para este segmento e isso não acontece na paginação já que todas as páginas tem tamanho igual.

Outra desvantagem é que a paginação tem milhares de páginas e a segmentação tem dezenas de segmentos então ao levar um segmento para o disco a probabilidade do processo precisar desse segmento de novo é muito maior do que na paginação.

3. Segmentação sofre de fragmentação interna? Explique. Paginação sofre da fragmentação externa? Explique.

A segmentação não sofre fragmentação interna (fragmentação de áreas com tamanhos iguais), pois os segmentos não possuem tamanhos fixos. Oposto a isso, a paginação não sofre fragmentação externa (fragmentação com áreas com tamanhos diferentes), pois as páginas possuem tamanho fixo.

4. Os endereços na paginação e na segmentação têm quantas partes? Quais são estas partes? Estas partes têm impacto direto na programação em linguagens de máquina(Justifique)?

Os endereços na paginação e na segmentação são divididos em duas partes. Para segmentação, as partes são número de segmentos e deslocamento dentro do segmento e para paginação número de páginas e deslocamento de página. Na segmentação impacta, uma instrução CALL por exemplo ficaria CALL 8:500 separando o número do segmento e o deslocamento dele, já na paginação isso não acontece, o endereço é um número "normal".

5. Para que serve o bit alterado(sujo)? Explique. Se o hardware não tiver o bit alterado, o propósito deste não será alcançado? Explique.

Para dizer se uma página que foi vítima foi alterada ou não, assim quando ela for vítima novamente se o bit alterado estiver ligado ele tem que salvar novamente, se estiver desligado significa que a página não foi alterada e não precisa salvar novamente em disco. Em hardwares que não possuem o campo alterado, é possível fazer uma simulação com o campo alterável. Quando uma página foi vítima e salva em disco o hardware coloca bit 0 em alterável. No caso de alteração, ocorre uma interrupção e o SO verifica o mapa de memória dos processos e corrige colocando o valor certo na tabela.

6. Mostre a fórmula para o tempo de acesso à memória lógica supondo as probabilidades de:

a. conversão via TLB

b. falta de página

Explique o que são todas as variáveis envolvidas. Considere que a tabela de página tem 2 níveis.

t = tempo de acesso a TLB

x = tempo de acesso a mem física

p = probabilidade de acesso via TLB

tempo de conversão = $(t) \cdot p + (t+x) \cdot (1-p)$

tempo de acesso a log = tempo de conversão + log em end

fis + tempo de acesso a mem fis

2 níveis: $[(t)*p + (t + x + x) * (1-p)] + x$

obs: $x + x$ pq tem 2 acessos a mem física para acessar os níveis da tabela.

7. Em uma máquina em que o tamanho da memória física é exatamente igual ao tamanho da memória lógica (em outras palavras o número de páginas físicas é igual ao número de páginas lógicas), é necessário utilizar o bit válido? Por quê?

Sim, poderia começar o processo com todas as páginas/segmentos com o bit válido 0 e só carregar na memória quando a página/segmento fosse referenciada.

8. Qual perda média de memória por processo devido à fragmentação interna? Seja T o tamanho da página e N o número médio de áreas.

Tamanho da memória = $T*N$

Melhor caso: 0

Pior caso: $(T*N) - 1$

perda média = $(\text{melhor caso} + \text{pior caso})/2$

perda média = $((T*N) - 1)/2$

9. Algoritmos de escolha de página vítima considerados bons necessitam de alguma facilidade adicional de hardware (além da tabela de página com bit válido)? Por quê?

Caso esta facilidade adicional exista, para que ela é usada?

Resposta 1:

Não é necessário, os algoritmos do tipo NRU utilizam o bit acessado que diz se a página foi acessada recentemente e o bit alterado que indica se página vítima foi alterada após ter sido vítima. O bit alterado possibilita saber se é necessário salvar a página em disco novamente. E o bit acessado é usado pra saber qual página deverá ser levada para o disco. O algoritmo da segunda chance por exemplo checa o bit acessado da primeira página da fila, se ele tiver o valor 1 essa página é levada para o final da fila e ele checa o bit acessado da próxima página até encontrar uma página com o bit acessado 0 e essa será a página vítima. Mas caso o HW não disponha do bit acessado o SO pode simular esse bit usando o bit válido. E o bit alterado também pode ser simulado utilizando o bit alterável.

Resposta 2: A existência de um hardware adicional não é obrigatória. Algoritmos do tipo NRU utilizam um bit adicional na tabela de páginas, chamado de bit “acessado”. O algoritmo da segunda chance utiliza esse bit para verificar se uma página escolhida como vítima foi utilizada

recentemente. Caso esse bit esteja ativo, o bit é zerado e a página “ganha uma segunda chance”, sendo movida para o fim da fila de prioridade, e a próxima página dessa fila é selecionada. Isso se repete até que uma página com o bit acessado desligado seja encontrada. Esse bit é ligado pelo hardware sempre que uma página sofre qualquer tipo de acesso. Apesar do algoritmo fazer uso desse bit, ele pode ser simulado pelo SO caso o hardware não o possua, o que torna a existência do hardware não obrigatória.

10.

a. O que é alcançado com a correção de endereços em tempo de carga?

Com a correção de endereços em tempo de carga é possível carregar um processo em qualquer posição da memória e assim conseguimos carregar mais de um processo ao mesmo tempo já que não há uma posição pré definida.

b. A correção de endereços em tempo de carga é realizada em uma máquina com segmentação? Por quê?

A correção de endereços em tempo de carga não é realizada em uma máquina que possua segmentação, pois todas as referências a memória são feitas baseadas nos endereços iniciais do segmento. Isto significa que podemos carregar o programa em uma área livre da memória e fazer os ajustes necessários nos endereços para que sejam acessados corretamente dentro do segmento.

c. A correção de endereços em tempo de carga é realizada em uma máquina com paginação? Por quê?

Não, cada processo tem a sua memória lógica (falsa), cada um tem seu início no zero que é falso, não tem que fazer correção de endereços.

11.

a. Descreva simultaneamente os três principais algoritmos de alocação de memória.

First Fit - Aloca sempre na primeira área livre encontrada, este é o algoritmo mais rápido

Best Fit - Procura uma área de tamanho igual ao processo que quer alocar, se não encontrar aloca em uma área maior mas com o menor tamanho possível

Worst Fit - Procura uma área de tamanho igual ao do processo para alocar, se não encontrar aloca em uma área maior mas com o maior tamanho possível.

b. Em uma máquina SEM paginação estes algoritmos devem ser usados para

i. Alocar as áreas do processo?

Sim, pois é responsabilidade do SO alocar as áreas do processo

ii. As variáveis dinâmicas?

Não, é responsabilidade do processo alocar suas próprias variáveis dinâmicas

c. Em uma máquina COM paginação estes algoritmos devem ser usados para

i. Alocar as áreas do processo?

Não, as páginas têm tamanhos fixos e não importa o tamanho do que quero alocar pois sempre sobrá o mesmo espaço

ii. As variáveis dinâmicas?

Não, as páginas têm tamanhos fixos e não importa o tamanho do que quero alocar pois sempre sobrá o mesmo espaço

12. Em que algoritmo de 2ª chance se parece com o do relógio e em que eles se diferenciam?

Os dois analisam páginas que não tiveram uso recente para escolha de página vítima.

No algoritmo da segunda chance quando uma página entra no final da fila e recebe o valor do bit acessado como 1, quando é necessário tirar uma página da memória ele olha o primeiro da fila e se o bit acessado for 1 ele zera esse bit e joga a página pro final da fila dando uma segunda chance.

Já no algoritmo do relógio essa fila não existe, ele tem um ponteiro que percorre as páginas zerando os bits acessados e dando a segunda chance até encontrar uma página com o bit 0 e transformá-la em página vítima.

13. Em uma máquina de 32 bits, com 1GB de memória física, tamanho da página 4Kbytes, tamanho da entrada na tabela de página igual a 8 bytes e 10 processos em execução, diga, justificando:

a. Qual o tamanho total gasto com tabela de página se esta não fosse dividida?

Memória lógica 32 bits = 2^{32} bytes

Quantidade de páginas lógicas = Memória lógica / 4kb = $2^{32}/4Kb = 1M$

Quantidade de entradas da tabela = Quantidade de Páginas Lógicas = 1M

Tamanho da Tabela de páginas = Quantidade de entradas da tabela * 8B (Tamanho da entrada) = $1M * 8B * 10$

b. Qual tamanho total gasto com tabela de página invertida (tamanho da entrada = 8 bytes e sem considerar tabela hash)?

Qtd de entradas da tabela invertida = Qtd de páginas físicas

Qtd de páginas física = tam mem fis/ tam da página = 1GB/4Kb = 250000 páginas

Qtd Entrada * Tamanho Entrada = 250k * 8B = 2000kB = 2MB

c. Qual o tamanho total gasto com tabela de página se for usada tabela em dois níveis e os processos tiveram apenas as áreas do código, dados e pilha com uma página cada.

10 bits | 10 bits | 12 bits

$2^{10} * 8 + 2^{10} * 8 + 2^{10} * 8 = 24kb * 10 \text{ processos} = 240 \text{ kb}$

14.

A. Compare, dizendo vantagens e desvantagens, o swap de processos e a segmentação sob demanda.

Swap de processos é quando um processo é removido temporariamente da memória e armazenado no disco e ,em seguida, retornado a memória para continuar sua execução.

Vantagens: poder executar mais processos do que cabe na memória, o programador não precisa fazer nada, pois o SO é responsável por tudo.

Desvantagens: demora em executar processos devido a troca do processo para o disco ou vice versa, o processo retirado da memória não é alocado no mesmo espaço de endereçamento que foi retirado.

Segmentação sob demanda é quando um segmento do processo é escolhido para ser removido temporariamente da memória e armazenado no disco, os segmentos dos processos possuem tamanhos variados.

Vantagens:não tem endereços absolutos, melhor controle pelo HW da memória. não importa para o programa onde está o segmento de memória, pois, o programa só sabe que é um endereço bidimensional, um processo consegue ser executado mesmo com falta de memória.

Desvantagens: é complicado escolher o segmento vítima, segmentos são grandes e demora a ser salvo em disco e para trazê-lo de volta, e cada segmento tem um tamanho diferente.

B. Compare, dizendo vantagens e desvantagens, a segmentação sob demanda e a paginação sob demanda.

Segmentação sob demanda é quando um segmento do processo é escolhido para ser removido temporariamente da memória e armazenado no disco, os segmentos dos processos possuem tamanhos variados.

Vantagens:não tem endereços absolutos, melhor controle pelo HW da memória. não importa para o programa onde está o segmento de

memória, pois, o programa só sabe que é um endereço bidimensional, um processo consegue ser executado mesmo com falta de memória.

Desvantagens: é complicado escolher o segmento vítima, segmentos são grandes e demora a ser salvo em disco e para trazê-lo de volta e cada segmento tem um tamanho diferente.

Paginação sob demanda é quando a memória é dividida em unidades de tamanhos iguais e essas páginas são salvas em disco conforme a necessidade de espaço na memória.

Vantagens: a principal vantagem é em relação ao tamanho. Na paginação como todas páginas tem o mesmo tamanho se precisou liberar espaço na memória física pode escolher qualquer página física para ser salva em disco que está bom, pois tem o mesmo tamanho da página lógica.

Desvantagens: não se aplica em processos de tempo real, pois esses tipos de processos que tem a necessidade de executar muito rápido.

15.

A. Qual seria o tamanho da tabela de páginas se esta existisse de forma contínua em uma máquina de 64 bits? Considere tamanho da página = 4KB e tamanho da entrada na tabela = 4B. Justifique.

Memoria logica 64 bits = 2^{64} bytes

Quantidade de paginas logicas = Memoria logica / 4kb = $2^{64}\text{bytes}/4\text{Kb}$

Quantidade de entradas da tabela = Quantidade de Paginas Lógicas

Tamanho da Tabela de paginas = Quantidade de entradas da tabela * 4B(Tamanho da entrada) = $2^{64}\text{ bytes} * 4\text{B} / 4\text{Kb}$

B. Por que a tabela de página em dois níveis não é uma boa solução para o problema do tamanho?

Porque o tamanho será gigantesco, a paginação em níveis para 64 bits deverá conter 5 níveis, Causando mais acessos a TLB, caso a página não esteja mapeada, gerando lentidão.

C. Qual a melhor solução para este problema (Justifique)?

A tabela de páginas invertida, pois terá menos acesso a memória que a em níveis pois com 5 níveis terei sempre 5 acessos a memória, enquanto que na tabela de páginas invertida eu tenho 2 acessos (um na tabela hash e um tabela invertida e mais um para cada colisão) então só terei 5 acessos se tiver 3 colisões.

16. Em um SO que utilize paginação sob demanda, o que acontece quando a memória física está cheia e um dos processos precisa utilizar uma página de dados que está em disco? Nesta situação, quantos acessos ao disco precisam ser realizados no pior caso

(Justifique)? Em termos de número de acesso ao disco, quais são os casos em que pode ocorrer melhoria?

Uma página que está na memória é levada para o disco para a página que está no disco e que precisa ser usada poder voltar para a memória. No pior caso seriam feitos dois acessos ao disco, um para salvar a página vítima e outro para carregar a página utilizada pelo processo. Para melhorar esse sistema o hardware (ou o SO, simulando uma tabela com o bit) pode utilizar um adicional para cada página. Esse bit é o alterado, em páginas que sofreram alteração o hardware (ou o SO) alteram esse bit para 1. Quando uma página já foi vítima antes, o SO verifica se a página sofreu alteração, caso não tenha sofrido ela é descartada, pois já possui uma cópia idêntica em disco.

17. É possível utilizar paginação em um hardware que não tenha tabela de páginas de espécie alguma, mas tenha uma TLB com poucas dezenas de entradas? Explique por que sim ou por que não.

Sim, mas vai ser muito mais lento. A conversão é feita via software depois de gerar uma interrupção. Nos SOs que faziam isso, a TLB passava a ter centenas de registros pra aumentar a chance de encontrar o registro na TLB. (resposta baseada no que ele respondeu em sala sobre essa questão)

18. Descreva as novas transições de estado que são utilizadas pelo swap de processos.

Bloqueado em disco - Processo estava bloqueado em memória e foi levado para o disco

Pronto em disco - Processo que tava bloqueado no disco e aconteceu o que o processo estava esperando então ele vai para o estado de pronto em disco e agora está esperando para ir pro estado de pronto e voltar a executar.

19. Considere um SO que utilize paginação sob demanda. Neste SO o armazenamento das páginas vítimas é feito em um disco que tem exclusivamente este propósito. Em um certo momento a utilização da CPU é de 20%, a utilização do disco mencionado acima é de 98% e a utilização de outros dispositivos (como discos e a interface de rede) é de menos de 5%.

a. Qual é o fenômeno que deve estar acontecendo?

Thrashing, quando os processos geram paginação excessiva e utilizam muito o disco e impedindo o uso total da CPU.

Qual o efeito teríamos na utilização da CPU se:

b. Aumentássemos o tamanho do disco?

Nenhum efeito, pois os processos ficam muito tempo ociosos devido a demora do tempo de acesso ao disco.

c. Aumentássemos a velocidade da CPU?

Nenhum efeito, a CPU não trabalha em capacidade máxima quando acontece thrashing, isso porque ela precisa esperar os dados do processo que estão em disco.

d. Aumentássemos a quantidade de memória?

O uso da CPU aumentaria, os processos iriam gerar menos falhas de falta de página.

e. Abortássemos um processo?

O uso da CPU aumentaria, os processos restantes teriam mais espaço em memória para salvar suas páginas e gerariam menos falta de páginas.

20. Seja um computador que utiliza paginação sob demanda e a tabela de página está toda contida em registradores. O tempo necessário para tratar uma falta de página quando existe uma página física livre ou quando a página vítima não foi modificada é de 8 milissegundos. O tempo necessário para tratar uma falta de página quando a página vítima foi modificada é de 20 milissegundos. O tempo de acesso a memória física é de 100 nanossegundos. Assuma que a página vítima foi modificada 70% das vezes. Considerando que p é probabilidade de ocorrer falta de página, explicita a fórmula do tempo médio de acesso à memória lógica (em função de p).

$$p \cdot (0.7 \cdot (20 \cdot (10^{-3})) + 0.3 \cdot (8 \cdot (10^{-3})))$$

21. Considere os algoritmos de escolha de página vítima descritos abaixo, ordene estes algoritmos segundo o critério de geração de menor número de faltas de página. Justifique sua resposta (OBS.: É possível ter empates entre algoritmos).

- LRU
- FIFO
- Ótimo
- Segunda Chance
- Relógio

Ótimo - É o algoritmo com o melhor desempenho, pois é o único que possui as informações de todas as páginas que serão usadas pelos processos. Algoritmo teórico.

LRU - mantém um histórico das páginas utilizadas pelos processos e elimina as que foram menos utilizadas. Ele se baseia na teoria de que

uma página que foi muito utilizada no passado, tende a ser utilizada novamente em um futuro próximo.

Segunda chance e Relógio - O segunda chance é uma modificação simples no algoritmo FIFO, utilizando um hardware extra que é o bit de referenciado, evitando assim descartar uma página intensamente usada. O relógio não precisa reinserir as páginas no final da lista pois ele mantém uma lista circular com todas as páginas e modifica somente os ponteiros. Difere do segunda chance somente na implementação.

FIFO - O mais simples de implementar e de baixo custo, porém ele pode descartar uma página intensamente utilizada.

22.

a. Explique, de forma sucinta, o que os estados “bloqueados em disco” (suspended blocked) e “pronto em disco” (suspended ready) de um processo e quais transições existem envolvendo estes estados.

Bloqueado em disco - Processo estava bloqueado em memória e foi levado para o disco

Pronto em disco - Processo que tava bloqueado no disco e aconteceu o que o processo estava esperando então ele vai para o estado de pronto em disco e agora está esperando para ir pro estado de pronto e voltar a executar.

Um processo bloqueado pode ir para o estado bloqueado em disco e de bloqueado em disco ele vai para pronto em disco para depois ir para o estado pronto e depois ser executado

b. O mecanismo associado a estes estados tem qual propósito?

Swap de processos, que leva um processo para o disco, um processo bloqueado, para que outro processo possa executar.

c. Além do propósito básico, este mecanismo resolve um problema da paginação sob demanda. Que problema é este? Como é resolvido?

Problema da memória física cheia. Na paginação quando não tem mais memória acontece o swap de páginas que é semelhante ao swap de processos.

23. O computador utiliza tabela de página invertida. Nesta máquina o endereço lógico tem 10 bits, as páginas têm 16 bytes, na tabela hash existem 4 entradas e existem 8 páginas físicas. A função hash utilizada é: $\text{numero_da_pagina_lógica} \text{ MOD } 4$. Existe uma lista de colisão. A situação atual da tabela hash e da tabela invertida está mostrada abaixo (todos os valores estão em binário).

Tabela Hash:

índice	valor
00	011
01	001
10	110
11	010

Tabela Invertida:

num_pag_logica	num_pag_fis	prox
000		1010
001		0010
	010	
	100	
011		0101
	(?)	
100		0100
101		1010
110		0010
	111	
111		1001
		000
		1110

Diga, explicando o que acontecerá na tradução dos seguintes endereços:

6 bits | 4 bits

a. 0010010011

001001|0011

$f = 001001 \% 4 = 9 \% 4 = 01$

01 001

001 0010

001001

001001 = 001001

encontrado

b. 1001100101

100110|0101

$100110 \% 4 = 10$

10 110

110 0010

100110 != 001010

prox: 111

100110 = 100110

Encontrado

c. 0011110011

001111|0011
001111%4=11
11 010
010 1110
001111 != 111000
prox 100
001111 != 010000
Não há prox, gera interrupção

24. Em uma certa máquina que utilize paginação, o máximo de memória física que ela pode ter é de 1GB, cada página tem 2KB. O tamanho lógico máximo de cada processo é de 128 MB. Com estes dados, determine explicando:

a. O número de bits no endereço físico;

$$2^{30}=1\text{gb}$$

30 bits

b. O número de bits para guardar o deslocamento da página;

$$2^{11}=2\text{kb } 11\text{bits}$$

c. O número de bits para guardar o número da página física;

$$1\text{gb}/2\text{kb} = 2^{30}/2^{11} = 2^{19} \Rightarrow 19 \text{ bits}$$

d. O número de entradas na tabela de páginas;

$$\text{Endereço lógico} = 128\text{mb}/2\text{kb} = 2^{27}/2^{11} = 2^{16} = 16 \text{ bits}$$

Cada entrada na tabela está relacionada a 1 página lógica do processo

e. Considerando que a única informação de controle de uma entrada na tabela de páginas diz respeito à validade da página lógica, qual o tamanho de uma entrada na tabela de páginas?

Tabela n pag física | validade

$$19 \text{ bits} + 1 \text{ bit} = 20\text{bits por entrada}$$

25. Suponha que exista um algoritmo de escolha de página vítima que se chame WS e que este algoritmo utilize a lógica de LRU, porém o número de páginas físicas alocadas ao processo seja igual ao tamanho do working set do processo. Considerando o intervalo de observação do working set como sendo de três acessos à memória lógica, mostre quando ocorre falta de página quando a seguinte sequência de acesso a páginas lógicas é utilizada:

24, 15, 18, 23, 24, 18, 17, 18, 24, 17, 17, 15, 24, 17, 24 e 18.

ATENÇÃO: O número de páginas físicas associado ao processo **NÃO** é constante, ele varia conforme o tamanho do working set (portanto, é diferente do que foi feito em sala de aula).

$ws(1)=\{24\}$ $ws(2)=24,15$ $ws(3)=24,15,18$ $ws(4)=15,18,23$ $ws(5) = 18,23,24$ $ws(6)=18,23,24$ $ws(7)= 24,18,17$ $ws(8)=18,17$ $ws(9)=17,18,24$
 $ws(10)=18,24,17$ $ws(11)=27,17$ $ws(12)=17,15$ $ws(13)=17,15,24$
 $ws(14)=15,24,17$ $ws(15)=24,17$ $ws(16)=17,24,18$

1	2	3	3	3	3	3	2	3	3	2	2	3	3	2	3
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
24	18	18	23	24	18	17	18	24	17	17	15	24	17	24	18
- 24	24	24	23	23	23	23	17	17	17	17	17	17	17	17	17
15	15	15	24	24	24	24	18	18	18	18	24	24	24	24	15
18	18	18	18	18	18	18	24	24	24	24	24	24	24	24	15
17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17
15	15	24	24	24	24	24	24	24	24	24	24	24	24	24	24
24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24

10 faltas de páginas

26. O tempo que demora a acessar uma grande estrutura de dados na memória lógica (em uma máquina com paginação) varia muito em função da ordem em que os elementos são acessados (isto é uma afirmativa). O acesso aos elementos na mesma ordem em que eles estão dispostos na memória lógica é, em muitos casos, bem mais rápido do que o acesso em outra ordem. Por exemplo, o código abaixo tem tempos de execução bastante distintos caso invertamos a ordem dos "for". Explique este fenômeno.

Dica: Imagine um pedaço do array (uma linha, por exemplo) localizado em uma página e a memória física é insuficiente para armazenar o array.

```

For (i=0; i<NUM_LINHAS;i++)
    For(j=0; j<NUM_COLUNAS; j++)
        Array[i,j] = func(x,y);

```

27. Seja a seguinte sequência de acesso a pags lógicas: A, B, C, D, A, B, E, A, B, C, D, E. Responda:

a. Mostre qual o número TOTAL de faltas de página se a memória física tiver 3 páginas e for utilizado o algoritmo da fila;

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
-	A	A	A	D	D	D	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
-	-	B	B	B	A	A	A	A	A	A	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
-	-	-	C	C	C	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
-	-	-	-	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
-	-	-	-	-	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
-	-	-	-	-	-	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
-	-	-	-	-	-	-	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
-	-	-	-	-	-	-	-	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
-	-	-	-	-	-	-	-	-	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
-	-	-	-	-	-	-	-	-	-	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J
-	-	-	-	-	-	-	-	-	-	-	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K
-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
-	-	-	-	-	-	-	-	-	-	-	-	-	M	M	M	M	M	M	M	M	M	M	M	M	M	M
-	-	-	-	-	-	-	-	-	-	-	-	-	-	N	N	N	N	N	N	N	N	N	N	N	N	N
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	O	O	O	O	O	O	O	O	O	O	O
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	P	P	P	P	P	P	P	P	P	P	P
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	R
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	S	S	S	S	S	S	S	S
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	T	T	T	T	T	T
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	U	U	U	U	U	U
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	V	V	V	V	V
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	W	W	W	W	W
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	X
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y	Y
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Z	Z
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

9 faltas de página

b. O mesmo do item A, porém agora com 4 páginas físicas;

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
-	A	A	A	A	A	A	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
-	-	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
-	-	-	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
-	-	-	-	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
-	-	-	-	-	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
-	-	-	-	-	-	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
-	-	-	-	-	-	-	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
-	-	-	-	-	-	-	-	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
-	-	-	-	-	-	-	-	-	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
-	-	-	-	-	-	-	-	-	-	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J	J
-	-	-	-	-	-	-	-	-	-	-	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K
-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
-	-	-	-	-	-	-	-	-	-	-	-	-	M	M	M	M	M	M	M	M	M	M	M	M	M	M
-	-	-	-	-	-	-	-	-	-	-	-	-	-	N	N	N	N	N	N	N	N	N	N	N	N	N
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	O	O	O	O	O	O	O	O	O	O	O
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	P	P	P	P	P	P	P	P	P	P	P
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	R
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	S	S	S	S	S	S	S	S
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	T	T	T	T	T	T
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	U	U	U	U	U	U
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	V	V	V	V	V
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	W	W	W	W	W
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	X
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y	Y
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Z	Z
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

10 faltas de página

c. Compare os resultados obtidos.

O número de faltas de A foi 9 e de B 10, aumentar o número de páginas na memória não necessariamente diminui o número de faltas de página (Anomalia de Bellady).

28. O thrashing ocorre quando a memória que está sendo ativamente usada é maior do que a máquina realmente possui, provocando a troca intensiva de páginas entre o disco e a memória física. Responda:

a. Explique o que é working set.

Conjunto de páginas utilizadas por um processo durante um certo intervalo de tempo observado, esse conjunto tem um tamanho variavel e permite ao SO saber o minimo de páginas que deve alocar para o processo para gerar o menor numero de faltas de páginas

b. Mostre uma condição para os working sets que terá como consequência o thrashing.

Numero de páginas alocadas ser menor que o tamanho do working set porque essa situação gera muitas faltas de páginas causando a troca intensiva de páginas entre o disco e a memória.

c. Qual a alteração que pode ser feita no hardware de forma a tentar reduzir o thrashing?

Utilizar um bit adicional na página/segmento, chamado alterado. Esse bit permite que o SO identifique páginas/segmentos que já foram pro disco e não foram alteradas, eliminando a necessidade de salvá-las novamente em disco.

29.

a. Explique a razão básica para um sistema operacional bloquear um processo.

Um processo usar uma página/segmento que ainda não foi carregada e precisa ser buscada no disco.

b. Seja a seguinte instrução, que está localizada no endereço lógico 4000: mov ah, [10000]. Esta instrução pode causar bloqueios no processo? Por quê? Caso possa, qual o número máximo de bloqueios que poderia existir?

Sim, o endereço que está procurando pode não estar carregado na memória. 2 bloqueios, quando a memória fica cheia e quando precisa trazer uma página/segmento do disco para a memória.

30. Diga, explicando quantos acessos à memória física são necessários para fazer a conversão do número página, via tabela de páginas, nos seguintes casos:

a. Tabela de página em 4 (quatro) níveis;

4 acessos, 1 no primeiro para pegar o endereço no segundo nível, 1 no segundo nível para pegar o endereço no nível 3 e um no nível 4 para pegar o endereço da página.

b. Melhor caso para a tabela de páginas invertida;

2 acessos, 1 na tabela hash e 1 na tabela invertida

c. Caso bastante ruim para a tabela de páginas invertidas;

No pior caso possível, seria o caso de ter vários conflitos então seria 2 acessos(um para a tabela e um para a tabela invertida) mais 1 para cada colisão. ex: se eu tiver 3 colisões serão $2+3=5$ acessos.

Tendo em vista estes casos, compare em termos de desempenho a tabela em 4 níveis com a tabela invertida.

A tabela invertida é melhor pois só vai ter 4 acessos se tiver 2 colisões enquanto que na tabela com 4 níveis eu sempre terei 4 acessos.

31. Suponha um hardware que utilize paginação e que a TLB seja gerenciada por software, isto é, todas as alterações na TLB sejam realizadas pelo sistema operacional. Descreva o que ocorre em relação à alteração da TLB nas seguintes situações:

a. Quando a conversão PODE ser feita via TLB;

Nada, o SO fará a consulta a TLB e a conversão sem alterar a TLB

b. Quando ocorre uma falta de página;

A TLB será modificada porque a página que está no disco tem seu bit inválido e não fica na TLB, essa página irá para a memória e seu bit vai ser alterado para 1 e ela entrará na TLB

c. Quando a conversão NÃO PODE ser feita via TLB.

O SO vai consultar a tabela completa e colocar a página na TLB, ela será alterada

32. Seja uma máquina com um sistema operacional que disponibilize F páginas físicas para um certo processo. Dado seqüências de acesso a páginas lógicas que tenham N acessos e nestes acessos apenas M páginas distintas são acessadas (isto quer dizer que uma mesma página lógica pode ser acessada mais de uma vez dentro da seqüência). Supondo a memória física originalmente vazia, diga usando as variáveis F, M e N e explicando:

a. Qual o MENOR número de falta de páginas que poderá ocorrer?

M faltas de páginas

b. Qual o MAIOR número de falta de páginas que poderá ocorrer?

N faltas de páginas

33. Em uma certa máquina, a página física tem 1KB e tabela de página está assim:

Num pág lógica	Num pág física	Vál ido	Alter ado	Aces sado
0	3	1	1	1
1	2	1	0	1
2	4	0	0	0
3	1	1	1	0
4	2	0	0	0
5	0	1	0	0

a. Caso seja utilizado o endereço 3527, qual será o resultado da conversão de endereços?

num página lóg = $3527/1024=3$

deslocamento = $3527\%1024=455$

página lógica 3 = página física 1 (bit válido é 1)

deslocamento = 355

b. Caso seja utilizado o endereço 2352, qual será o resultado da conversão de endereços?

num página lógica = $2352/1024= 2$

deslocamento = $2352\%1024=304$

página lógica 2 tem o bit válido = 0 ento não será acessada, vai gerar uma interrupção

c. Caso precise escolher uma página como vitima, qual seria a melhor escolha? Por quê?

Página 5 porque seu bits alterado tem o valor 0 então não tem que salvar de novo em disco e seu bit acessado é 0 o que significa que ela não foi acessada recentemente.

d. O acesso a um endereço na página lógica 4 fará com que o processo seja abortado em qualquer situação? Por quê?

Não, vai gerar uma interrupção de falta de página pois seu bit válido é 0 e vai ver se essa página está no disco para levá-la para a memória

e. O que pode ter acontecido com a página lógica 3 para que os bits alterado e acessado estejam no estado que estão?

Ela foi alterada, seu conteúdo não é o mesmo que está no disco e não foi utilizada recentemente

34. Dada as informações seguintes a respeito das páginas físicas e em certo momento não existiam páginas físicas livres e o SO decida criar um 'pool' de páginas livres contendo duas páginas, o que ocorrerá se forem utilizados os seguintes algoritmos:

a. FIFO;

Vai retirar as páginas 2 e 5

b. LRU;

Vai retirar as páginas 5 e 7

Num pág. Física	Num pág. Lógica	Pág em uso	Tempo ocorreu carregamento página	Tempo ocorreu último uso da pág
0	77	S	123	270

1	48	S	231	250
2	102	S	101	278
3	95	S	145	279
4	37	S	123	253
5	25	S	109	167
6	88	S	132	198
7	6	S	143	185

35. Em uma máquina com sistema operacional que use segmentação sob demanda, o fabricante do SO deseja implementar um algoritmo do tipo NRU para a escolha do segmento vítima. Porém, a MMU não dispõe do bit 'acessado' (ou 'usado') na tabela de segmentos. O que pode ser feito para conseguir implementar o algoritmo?

Pode ser usado o bit válido para simular o bit acessado, ele coloca 0 em todas as páginas que estavam válidas e usa uma tabela auxiliar que tem um campo chamado realmente válido que contém o valor verdadeiro do campo válido. Quando o SO consultar e ver o bit válido 0 ele irá na tabela auxiliar consultar qual o valor verdadeiro e colocar no bit válido, de tempos em tempos esse bit vai receber o valor 0 novamente.

36. Em uma máquina que não use registrador de base, endereçamento relativo, segmentação ou paginação, responda:

a. Porque é necessário fazer a correção de endereço antes da execução do código de um processo?

Quando o compilador for gerar o código executável ele acha que começa no 0 então é preciso fazer a correção de endereços para que não tente acessar endereços inválidos.(correção de endereços em tempo de carga)

b. Quem é responsável por fazer a correção?

O SO

c. Em que parte da memória do processo estão os endereços corrigidos?

Estão em uma tabela de endereços que faz parte do código executável.

37.a. Quais são as áreas de memória típicas de um processo no Unix? O que é armazenado nelas?

Área de código - Guarda o código executável do programa

Área de Dados - Contém as variáveis globais e as Dinâmicas

Área de Pilha - Contém as variáveis locais

b. Além das mencionadas no item A, existem outras áreas que um processo pode ter?

Apenas áreas livres para alocação de outros processos.

c. Para as áreas do item A, na paginação, quantas áreas de cada tipo existem?

Existe uma área de código e uma área de dados e pode existir mais de uma área de pilha

d. E quando não é usada paginação, existe alguma vantagem de ter números diferentes do item C?

Sim, a pilha não pode ser dividida mas pode ter mais de uma área de pilha

38. Sejam três máquinas M1, M2 e M3 que utilizem paginação e possuam TLB. A máquina M1 tem uma TLB com tempo de acesso de 12ns e tem a probabilidade de conversão via TLB (taxa de acerto = hit ratio) de 60%. A máquina M2 tem uma TLB maior com uma maior taxa de acerto, 80%, porém um tempo de acesso pior, 18ns. Estas duas máquinas conseguem fazer a pesquisa na TLB simultaneamente ao acesso à memória física. A máquina M3 tem uma taxa de acerto 80% e um tempo de acesso à TLB de 12ns, porém o acesso à memória ocorre sempre depois do acesso à TLB. Em todos os casos o tempo de acesso à memória física é de 60 ns. Ordene as máquinas em relação ao tempo médio de conversão.

tempo médio de acesso a mem log = tempo médio de conversão + tempo de acesso a mem física

tempo acesso a lógica = temp de conversão end log em end fis+tempo de acesso a mem fis

$$M1 = (12 + 60) * 0,60 + ((12+60+60)*(0,4)) = 43,2 + 52,8 = 96ns$$

$$M2 = (18+60)*0,8 + ((18+60+60)*(0,2)) = 62,4 + 27,6 = 90ns$$

$$M3 = (12 + 60) * 0,8 + ((12 + 60 + 60) * (0,2)) = 57,6 + 26,4 = 84ns$$

$$M3 - M2 - M1$$

39. Seja uma máquina de 32 bits (para endereços lógico e físico) com páginas de 1KB.

a. Digamos que esta máquina utilize uma tabela de páginas de dois níveis em que a quantidade de bits referente a página lógica é igualmente dividida entre cada nível. A entrada no 1º nível tem 3

bytes de tamanho. No 2º nível, a entrada tem 10 bits de controle. Calcule, explicando o tamanho máximo desta tabela de páginas (para um processo).

11bits | 11 bits | 10 bits

qtd pág lógica = $2^{32}/2^{10}=2^{22} = 4G$

2G no primeiro nível e 2 GB no segundo nível

Tam da Tab de páginas = $2G*3 + 2G*1k$

b. Se esta máquina utilizar tabela invertida, considere que o número máximo de processos é 65536 (16), que existam 4bits de controle e que o número de entradas na tabela hash é igual ao número de entradas da tabela invertida. Calcule, explicando o tamanho da estrutura da tabela invertida (incluindo a tabela hash). Considere que o tamanho da entrada em qualquer das tabelas ocupam um número inteiro de bytes.**

40. Em uma máquina sem paginação, segmentação, registrador de base ou endereçamento relativo, diga:

a. Como é possível executar dois programas quaisquer simultaneamente sem a necessidade de recompilá-los? Dica: lembre-se dos endereços em linguagem de máquina.

Usando correção de endereços em tempo de carga, vai mudar no código dos processos para os endereços corretos

b. Neste caso, quem é o responsável por implementar a solução para este problema?

O SO é o responsável pela correção dos endereços

41. Quando a memória física está cheia e ocorre uma tentativa de acesso a uma página lógica que está em disco, explique: Qual(is) situação(ões) na(s) qual(is) NÃO é necessário um duplo acesso ao disco?

Quando a página vítima já foi vítima anteriormente e está com o bit alterado com o valor 0 então não será necessário salvar ela novamente no disco ou quando a página é uma página de código pois não são alteráveis e já está no disco dentro do arquivo exe.

42. Suponha um computador com endereço lógico de 38 bits, página de 16KB e 4 bytes para cada entrada na tabela de página. A tabela de página tem dois níveis e cada parte da tabela de página cabe uma página física. Responda, explicando:

a. Quantas páginas lógicas existem?

$2^{38}/16kb=17$ mil páginas

b. Diga a divisão de bits entre deslocamento, diretório da tabelas de página e 2º nível da tabela.

12 bits|12 bits|14 bits

c. Dado um processo em que todas suas páginas estejam na memória física, diga quantas páginas físicas são usadas por este processo, inclua no cálculo as páginas físicas usadas pela tabela de página deste processo.

$17m * 4B = 68 \text{ mil}$

43. Dadas as suposições:

a. O sistema operacional utiliza paginação sob demanda

b. Em um certo momento da execução do processo algumas páginas estão na memória física e outras estão no disco

c. Um resumo da tabela de página está na TLB

d. O tempo de acesso a TLB é de 10 nanossegundos

e. O tempo de acesso à memória física é de 100 nanossegundos

f. O tempo de acesso ao disco é de 100000 nanossegundos

g. Para cada acesso à memória lógica, primeiro tenta-se fazer a conversão via TLB

h. 90% das vezes a conversão é feita via TLB e a paginação lógica acessada está na memória física

i. 10% das vezes a conversão não está na TLB

j. Neste caso, é necessário fazer acesso à tabela de páginas para fazer a conversão

k. 80% destes 10% dos casos (8% do total, portanto) a página lógica está na memória física e, sendo assim, a memória física é acessada.

l. Nos 20% dos 10% dos casos (2% do total) a página lógica está no disco e é necessário um acesso ao disco para trazê-la para a memória física

m. Em 15% dos casos descritos no item l, antes de podermos trazer a página lógica para a memória física é necessário salvar uma página vítima no disco, gastando, portanto, mais um acesso ao disco

n. Uma vez que a página lógica volta para a memória física esta sofre o acesso que era desejado

Perguntas:

i. Qual é o nome do evento descrito na suposição 1?

ii. Na suposição m, por que em alguns casos é necessário salvar a página vítima no disco?

Porque a página pode ter sido alterada ou nunca ter sido vítima então será necessário salvá-la em disco

iii. Na suposição 'm', mesmo quando é escolhida uma página vítima, porque não temos sempre que salvá-la no disco?

Porque ela já pode ter sido vítima anteriormente e seu bit alterado ser 0 então já existe uma cópia dessa página salva em disco

iv. Na média, quantos nanossegundos são necessários para fazer um acesso à memória lógica? Mostre os cálculos, (Dica: pode ajudar fazer as contas calculando 1.000 acessos a memória lógica, depois é só dividir por mil).

tempo médio de acesso a mem log = tempo médio de conversão + tempo de acesso a mem física

tempo consulta a TLB = 10ns

tempo consulta a mem fis = 100ns

tempo acesso a lógica = temp de conversão end log em end fis + tempo de acesso a mem fis

tempo médio de conversão: $P \cdot 10\text{ns} + (1-P) \cdot 110\text{ns} = -100P + 210 = -100 \cdot 0,9 + 210 = 120\text{ns}$

44. O uso de variáveis (normalmente ponteiros) que contém endereços absolutos (aqueles que são a partir do início da memória) cria impedimentos para a implementação de um mecanismo importante para a gerência de memória. Que mecanismo é este? Por que ocorre o impedimento?

Correção de endereços relativo a instrução atual. Porque ele não sabe exatamente onde na memória que está o endereço.

45. O que é linkagem dinâmica? Ela tem utilidade para economia de memória? Explique.

A linkagem dinâmica é um mecanismo para carregar uma biblioteca de processos. Ela tem utilidade de memória porque através de uma pequena parte do código o SO verifica se as bibliotecas necessárias a ele já estão na memória evitando a duplicação de código na memória.

46. Considerando um hardware de paginação (similar ao de uma antiga SUN SPARC Station) no qual a tabela de página é composta por três níveis. Nesse hardware um endereço lógico é dividido em 3 partes: a 1ª tem 8 bits e serve para identificar uma entrada no 1º nível da tabela de páginas; a 2ª tem 6 bits e serve para identificar uma entrada no 2º nível da tabela de páginas; a 4ª tem 6 bits e serve para identificar uma entrada no 3º nível da tabela de páginas;

a 4ª tem 12 bits e serve para identificar o deslocamento dentro de uma página.

8 bits | 6 bits | 6 bits | 12 bits

a. Calcule o tamanho do espaço lógico de endereçamento (memória lógica).

$$2^{32}=4\text{GB}$$

b. Calcule o tamanho de cada página.

$$2^{12}=4\text{kb}$$

c. Considerando que cada entrada na tabela de páginas (em qualquer um dos níveis) tenham 64 bits, qual o menor tamanho possível que uma tabela pode ocupar.

$$2^{32}/2^{12}=2^{20}$$

$$2^{20} \times 64$$

d. Considerando esse arranjo multinível, calcule os números mínimo e máximo de acessos à memória eventualmente necessários para conseguir converter um endereço lógico para físico.

Como a tabela tem 4 níveis, serão sempre 4 acessos

47. Compare fragmentação interna com fragmentação externa. Qual a relação destas com paginação e segmentação? Explique.

Fragmentação interna ocorre quando há perda de espaço em área de tamanhos fixos e a fragmentação externa ocorre em uma área com tamanhos variados. Na paginação só é possível acontecer fragmentação interna já que todas as páginas têm tamanhos iguais e na segmentação acontece a fragmentação externa porque os segmentos têm tamanhos variados.

48. Em um processo tradicional com uma única thread diga:

a. O que é armazenado na pilha?

Variáveis Locais

b. O que é armazenado na heap?

Variáveis Dinâmicas e Globais

c. Qual a diferença quando o processo tem outra thread?

Cada Thread tem as suas variáveis mas elas podem compartilhar memória.

d. É possível inverter o que é armazenado? Isto é, colocar na heap o que normalmente é armazenado na pilha e colocar na pilha o que é normalmente armazenado na heap? Explique.

49.

a. É possível que duas entradas na TLB tenham o mesmo número de página física? Explique.

Não, uma página física só pode conter uma página lógica então as entradas da TLB não podem apontar para páginas físicas 2 vezes já que na TLB só é guardada as páginas lógicas válidas

b. É possível que duas entradas na TLB tenham o mesmo número de página lógica ? Explique. (Lembre que existem diferentes tipos de TLB)

Sim, podem ser páginas de dois processos diferentes. Esse tipo de TLB tem um campo a mais chamado número do processo para indicar de qual processo pertence a página

50. Para que serve o modelo working set? Explique

Working set é um conjunto de páginas que o processo usou em certo intervalo de tempo observado, serve para saber o mínimo de páginas que devem ser alocadas ao processo de acordo com o tamanho do working set

51. Por que o tamanho da página é sempre potência de 2?

Porque o sistema numérico dos computadores é binário

52. Um computador tem um endereço lógico de 64 bits e páginas de 2Kb. Uma entrada na tabela de páginas ocupa 4 bytes. Uma tabela de páginas em níveis é utilizada e cada pedaço (em qualquer nível da tabela) está contido em uma página. Quantos níveis são necessários? Explique. (Dica: o tamanho da página limita o número de bits de cada nível)

$$2^{11}/2^2=2^9$$

$$64-11=53$$

$$53/9=5,8=6 \text{ níveis}$$

8 bits | 8 bits | 8 bits | 8 bits | 10 bits | 11 bits | 11 bits

53. Descreva a vantagem da "linkagem" dinâmica sobre a "linkagem" estática.

A linkagem estática identifica todos os objetos que serão usados durante o código e compila em diversos objetos de código, ela insere este código dentro do programa final e transforma em um unico programa, já a dinamica ao inves de inserir no codigo final ela adiciona ponteiros para estes arquivos que serão chamados durante o tempo de execução. A vantagem é que ela não copia o código então na linkagem estática por copiar para o código este código fica bem maior.

54. Para cada um dos esquemas de gerenciamento de memória diga se existem as seguintes coisas/situações:

a. TLB

b. Fragmentação Interna

c. Fragmentação Externa

Esquemas de gerenciamento de memória:

i. Registrador de base;

a) TLB - Não existe pois é uma tabela resumo da tabela de páginas

b) Fragmentação Interna - Não existe, guardam um endereço da base

c) Fragmentação Externa - Não existe, guardam um endereço da base

ii. Segmentação;

d) TLB - Não existe, é uma tabela resumo da tabela de páginas

e) Fragmentação Interna - Não existe pois é a fragmentação de áreas com tamanhos iguais e os segmentos possuem tamanhos diferentes

f) Fragmentação Externa - Existe pois é a fragmentação de áreas com tamanhos diferentes e os segmentos possuem tamanhos diferentes

iii. Paginação;

g) TLB - Existe pois é uma tabela resumo da tabela de páginas

h) Fragmentação Interna - Existe pois é a fragmentação de áreas com tamanhos iguais e as páginas possuem tamanhos iguais

i) Fragmentação Externa - Não Existe pois é a fragmentação de áreas com tamanhos diferentes e as páginas possuem tamanhos iguais

Justificar cada caso (são nove no total)

55. Quais das características estão presentes em uma máquina que está em thrashing:

a. O disco utilizado para armazenar as páginas vitimas esta quase 100% em uso.

Existe, alto indice de falta de páginas

b. O disco utilizado para armazenar as paginas vitimas está quase 100% cheio.

Não, o thrashing não necessariamente precisa ocupar quase todo o disco, basta que os processos em uso gerem muitas faltas de paginas/segmentos.

c. Alguns processos estão bloqueados quase todo o tempo.

Sim, uso excessivo de páginas/segmentos em disco

d. A CPU está quase 100% ocupada.

Não, o thrashing é caracterizado pelo baixo uso de CPU.

e. A CPU está quase 100% livre.

Sim, está aguardando dados em disco

Justifique suas respostas para cada item

56. Relativamente ao thrashing, responda:

a. Explique o que é working set.

Conjunto de páginas lógicas que um processo utiliza em certo intervalo de tempo observado

b. Caso o SO utilize alocação global, mostre uma condição para os working sets que terá como consequência o thrashing.

Working set ser maior que o número de páginas alocadas para o processo, vai ter muitas faltas de páginas fazendo com que o disco seja altamente utilizado e causando interrupção de falta de página

57. Supondo um SO que suporte apenas um único processo, seja a MMU que utilize tabela invertida com a função hash igual a $\text{num_pág_lógica} \text{ MOD } 3$. O computador possua 6 páginas físicas. As páginas lógicas utilizadas pelo processo são: 0, 1, 2, 3, 8 e 9. Mostre desenhando, como ficam as tabelas hash e invertida.

$$f(0) = 0$$

$$f(3) = 0$$

$$f(1) = 1$$

$$f(8) = 2$$

$$f(2) = 2$$

$$f(9) = 0$$

nº Pág. lóg. próx. reg. de colisão Tab Hash

5	0	4	5	
4	9	2	4	
3	8	1	3	
2	3		2	3
1	2		1	0
0	1		0	5

58. Considere as seguintes ações. Diga, explicando, quais são realizadas pelo núcleo do SO e não podem ser realizada por um processo comum e quais podem ser feitas por esse processo:

a. Alterar o valor do registrador que aponta para a tabela de páginas

SO, somente ele tem acesso a tabela de pagina lógica

Alterar o valor do registrador que aponta para o topo da pilha (Stack Pointer - SP)

So, ele realoca novas áreas de memória

c. Acrescentar um segmento novo na tabela de segmentos

so, só ele pode alterar a tabela de segmentos

d. Alterar um byte dentro de um segmento compartilhado por múltiplos processos

Processos podem alterar valores dentro de segmentos compartilhados (bibliotecas).

e. Alterar o bit válido de uma entrada da tabela de páginas.

so, só ele pode alterar a tabela de páginas

59. Em um computador que tenha um endereço lógico de 38 bits, tamanho da página de 16KB e 4 bytes para cada entrada na tabela de páginas (em qualquer dos níveis), responda, explicando:

a. Quantas páginas existem no espaço de endereçamento lógico (memória lógica)?

qtd pág lógicas = $2^{38}/2^{14}=16M$

b. Mostre como poderia(m) ser a(s) quebra(s) em bits do endereço lógico dado que a tabela de páginas tem dois níveis.

12 bits | 12 bits | 14 bits

c. Suponha que exista um processo que tenha 32MB e que todo processo esteja na memória física. Qual a quantidade de memória física utilizada pelo processo (incluindo a memória gasta com a tabela de páginas).

qtd página = $32mb/16kb=2^{11}$ páginas

$2^{1284}+2^{1284}+32mb$

60. Considere o mecanismo de tradução de endereços utilizado pela intel para 64 bits: endereço lógico de 48 bits usando uma tabela de páginas de 4 níveis para traduzir para um endereço físico de 52 bits, tamanho de pagina de 4096 bytes. Cada entrada na tabela de página (em qualquer nível) ocupa 8 bytes. Considerando que cada nível da tabela de página ocupa apenas uma página. Diga qual deve ser o tamanho da página para que a tabela de páginas tenha apenas dois níveis.

$2^{12} = 4096$

tam página/8 = qtd registros = 2^y qual deve ser o tam da pág?

$2^x/2^3=2^y \rightarrow x-3=y$

tam página = 2^x

$2^y+x=48$

$2^{*(x-3)} + x = 48 \rightarrow x=18$

2^{18} =tam página para ter 2 níveis
15 bits|15 bits|18 bits

61. Seja um computador que utiliza paginação sob demanda e a tabela de páginas está toda contida em registradores (sendo assim, o gasto de tempo consultando a tabela de páginas ficaria desprezível). O tempo necessário para tratar uma falta de página quando existe uma página física livre ou quando a página vítima não foi modificada é de 10 milissegundos. O tempo necessário para tratar uma falta de página quando a página vítima foi modificada é de 20 milissegundos. O tempo de acesso a memória física é de 100 nanossegundos.

A) Qual a explicação para o segundo tempo (20ms) ser o dobro do primeiro (10ms)?

Porque no 1º não houve modificação da página então não é necessário salvar a página novamente em disco, já no segundo houve a modificação na página então é necessário salvar ela no disco fazendo assim 2 acessos ao disco.

B) Assuma que a página vítima foi modificada 70% das vezes. Considerando que p é a probabilidade de não ocorrer falta de página, explicita a formula do tempo médio de acesso a memória logica (em função de p).

$$p*100+(1-p)*(0,7*20+(1-0,7)*10)$$

62. A) o que aconteceria se não existisse TLB? Por quê?

Afetaria o desempenho do processo, já que teria que consultar sempre a tabela de páginas na memória que demora mais tempo que na TLB.

B) Dado que a segmentação usa a tabela de segmentos fale se a TLB é necessária na segmentação

Não, a segmentação apresenta registradores de segmento que armazenam o endereço do segmento tornando mais rápida a execução por não necessitar acessar sempre a tabela de segmentos. Trazem o mesmo beneficio que a TLB em relação a desempenho tornando a TLB desnecessária.

63. Explique uma vantagem do swap de processos em relação ao overlay. Explique uma vantagem do overlay em relação ao swap de processos.

Uma vantagem do swap de processos em relação ao overlay é que no swap de processos é retirado a responsabilidade do programador de organizar a entrada de partes de programa na memória, o SO passa a ser o responsavel o que diminui a ocorrencia de erros.

Uma vantagem do overlay em relação ao swap de processos é o fato do overlay poder executar programas com tamanho maior que o da memória física já que não necessita armazenar todo o conteúdo do programa pra ele. Permite a execução de programas grandes mesmo que o tamanho da memória física seja inferior.

64. Explique uma razão que pode provocar a existência de uma certa página lógica na memória física, porém a sua entrada na tabela de página não estar válida.

O bit válido pode estar sendo usado pra simular o bit acessado, o bit acessado serve para informar se a página foi acessada recentemente ou não. Quando o hardware não tem o bit acessado ele pode simular usando o bit válido, ele zera o valor de todas as páginas válidas e coloca em uma tabela auxiliar que contém o bit chamado realmente válido o valor verdadeiro deste bit válido. Ao tentar acessar a página e o bit válido estiver zerado vai gerar uma interrupção e o SO vai na tabela auxiliar checar o valor verdadeiro.

65. Considere um gerenciamento de memória que utilize overlays. Seja um processo P que tenha sua memória restrita a 512KB. Este processo está estruturado em dois overlays cada um com código e dados exclusivos. O código do overlay A tem 180KB, os dados exclusivos do overlay B somam 40KB, os dados usados por ambos overlays tem 80KB. Seja X o tamanho do código comum e y o tamanho do código do overlay B e Z o tamanho dos dados exclusivos do overlay A. Qual das seguintes valores (X,Y,Z) permitem que o processo execute. Justifique

- (a)(220KB, 140KB, 40KB)
- (b)(220KB, 180KB, 10KB)
- (c)(160KB,240KB,60KB)
- (d)(160KB, 200KB, 100KB)
- (e)(160KB, 220KB, 80KB)

4)	exclusivos		comum		
	Código	Dados	Código	Dados	
A	180KB	Z	X	80KB	(I)
B	Y	40KB	X	80KB	(II)

Para os valores de X, Y, Z serem válidos a soma da linha A e da linha B têm que ser menores que 512KB. Com isso, será possível executar tanto o overlay A quanto o overlay B.

I) Linha A: $180 + 80 + X + Z < 512 \Rightarrow X + Z < 252$

II) Linha B: $40 + 80 + X + Y < 512 \Rightarrow X + Y < 392$

a) $X = 220$ $Y = 140$ $Z = 40$

I) $X + Z = 260 > 252$ X A ultrapassa o limite de 512KB

II) $X + Y = 360 < 392$ ✓

Não permite execução

b) $X = 220$ $Y = 180$ $Z = 10$

I) $220 + 10 = 230 < 252$ ✓

II) $220 + 180 = 400 > 392$ X B ultrapassa o limite

Não permite execução

c) $X = 160$ $Y = 240$ $Z = 60$

I) $160 + 60 = 220 < 252$ ✓

II) $160 + 240 = 400 > 392$ X B ultrapassa o limite

Não permite execução

d) $X = 160$ $Y = 200$ $Z = 100$

I) $160 + 100 = 260 > 252$ X A ultrapassa o limite

II) $160 + 200 = 360 < 392$ ✓

Não permite execução

e) $X = 160$ $Y = 220$ $Z = 80$

I) $160 + 80 = 240 < 252$ ✓

II) $160 + 220 = 380 < 392$ ✓

Permite execução

66. O que acontece com a TLB em uma troca do processo de execução? Responda para todos os tipos de TLB. Explique a razão destes comportamentos nas TLBs

Para a TLB simples a cada troca de processo a TLB será esvaziada porque ela armazena as páginas de um único processo então quando ocorre a troca de processo a TLB vai passar a armazenar só as páginas

do processo em execução no momento. Antes de fazer o armazenamento das páginas do novo processo a TLB é esvaziada para não conter as páginas do processo anterior.

Para a TLB estendida será incluído novos registros nela referentes as páginas do novo processo e os anteriores não serão apagados. A TLB estendida armazena as páginas de processos distintos e os diferencia pela coluna nº do processo.

67. Caso o tamanho da página seja aumentado (mantendo todas as demais características do computador), o que fica melhor ou pior em termos de:

a) desperdício de memória

Pior. Ao Alocar algo que era menor do que o tamanho da página ira sobrar mais memória sendo desperdiçada.

b) tempo medio de conversão de endereços

Melhora, pois ao aumentar o tamanho da página eu diminuo o número de páginas usadas diminuindo assim a quantidade de registros na tabela e aumentando a probabilidade da página ser encontrada na TLB. Ao aumentar esse número eu consigo diminuir o tempo de conversão já que o tempo médio de conversão é

$(\text{Tempo acesso TLB}) \cdot p + (\text{Tempo acesso TLB} + \text{Tempo acesso Mem.}) \cdot (1-p)$

ex: tam página = 4kb

tempo acesso a tlb = 10ns tempo acesso a mem = 100ns

prob = 0,5

tempo de conversão = $10 \cdot 0,5 + 110 \cdot 0,5 = 60\text{ns}$

Aumento a página para 5kb e supondo que agora a probabilidade é:

prob = 0,67

tempo de conversão = $10 \cdot 0,67 + 110 \cdot 0,33 = 43\text{ns}$

c) espaço gasto para controle de paginação

Melhor pois ao aumentar o tamanho da página iria diminuir a quantidade de registros dentro da tabela

68.

3) Considere um computador com as seguintes características: endereços de 32 bits, páginas de 256 bytes, uma entrada na tabela de páginas ocupando 4 bytes, não usa TLB. Este computador pode usar tabelas de página com um único nível ou tabelas de página de 3 níveis com um número igual de entradas em cada nível. Responda, explicando: A) Se todas as páginas lógicas estão alocadas, qual o espaço gasto com a tabela de páginas em um único nível? B) Se todas as páginas lógicas estão alocadas, qual o espaço gasto com a tabela de páginas em um três níveis? C) Se somente os 4KB iniciais e os 4KB finais da memória lógica estão alocados, qual o espaço gasto com a tabela de páginas em um três níveis? D) Para processos que alocam sempre uma fração da sua memória lógica há desvantagem em usar tabela de páginas em três níveis?

69. a) O número de páginas físicas pode ser menor que o número de páginas lógicas? Se sim, qual a utilidade desta situação?

Sim. As páginas lógicas ficam na memória lógica e como a memória lógica é falsa ela pode ser bem grande pois só existem de fato as páginas com conteúdo. A vantagem de se ter mais páginas lógicas do que físicas é que isso permite a execução de processos com tamanho maior que a memória física.

b) O número de páginas físicas pode ser maior que o número de páginas lógicas? Se sim, qual a utilidade desta situação?

Sim. Tendo uma quantidade de páginas físicas maior do que de páginas lógicas terá mais memória física para alocar para outros processos.