1. Explique a diferença entre endereço lógico e endereço físico.
R: ANSE: O endereço lógico é um endereço que é obtido em relação ao programa em execução, como o endereços lógicos iguais podem ter endereços físicos diferentes pois os programas podem estar em espaços de endereçamentos diferentes.
RDM: Endereço lógico é o endereço a nível de programa que é gerado na compilação, ele enxerga a memoria como sendo unicamente para o programa. Através da realocação dinâmica que consiste em utilizar um endereço base(endereço físico) e os endereços lógicos como offset, obtem-se o endereço físico para cada endereço lógico. Sendo o endereço físico um endereço que representa uma localização real e valida na memória.

2. Explique os seguintes algoritmos de alocação de memória:
a) First-fit:(o primeiro que couber) O dado é alocado na primeira área de memória que estiver disponível. O algorítimo é rápido porém fragmenta muito a memória.
b) Best-fit: (o que melhor couber) O dado é colocado na menor área em que servir na memória. Diminui o número de buracos mas o algorítimo é lento.
c) Worst-fit:(o que pior couber) O dado é colocado no maior espaço disponível da memória. Como conseqüência a memória fica muito fragmentada e é um algorítimo lento.

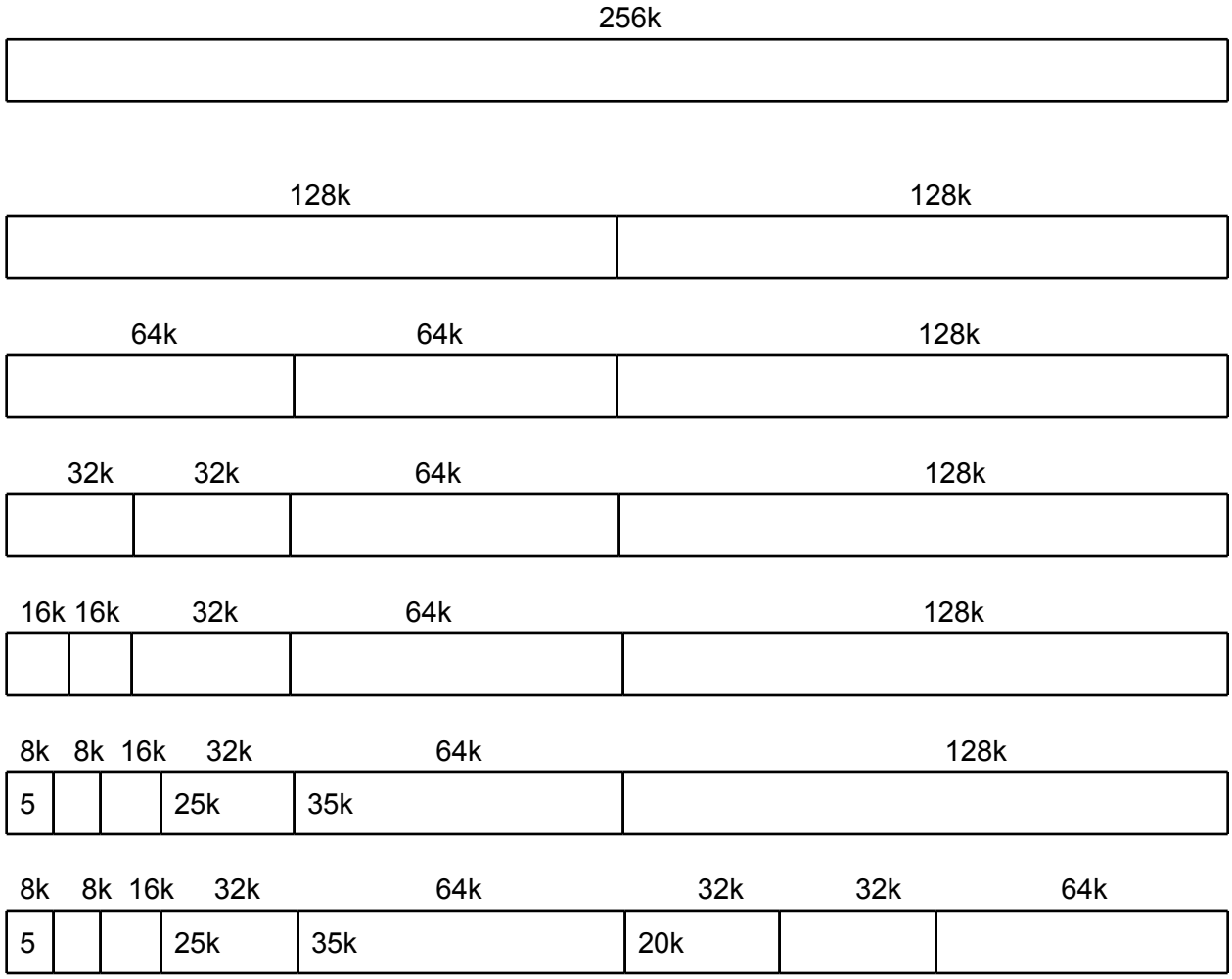
3. Considere as seguintes partições livres: 10K, 20K, 4K, 7K, 9K, 12K, 15K e 18K. Como os algoritmos First-fit, Best-fit, Worst-fit e Next-fit alocariam partições para as seguintes requisições:
a) 12K
b) 10K
c) 9K

	10k	20k	4k	7k	9k	12k	15k	18k
First-Fit	10k	12k			9k			
Best-Fit	10k				9k	12k		
Worst-Fit		12k					9k	10k
Next-Fit		12k				10k	9k	

4. Explique a diferença entre fragmentação externa e interna.
R: Na fragmentação externa o espaço requisitado existe mas não é contíguo, já na fragmentação interna o espaço para alocação do dado pode ser um pouco maior porém quando esse dado for alocado sobrará um pequeno espaço inutilizado.

5. Considere um computador usando Buddy System para o gerenciamento de memória. Inicialmente ele tem um bloco de 256K no endereço 0. Depois das requisições 5K, 25K, 35K e 20K acontecerem, quantos blocos existem e qual o tamanho e endereço?

R:



Existirão 8 blocos com tamanho e endereço mostrado na tabela abaixo:

Tamanho	Endereço
8k	0x0
8k	0x2000
16k	0x4000
32k	0x8000
64k	0x10000
32k	0x20000
32k	0x28000
64k	0x30000

(não tenho certeza se calculei direito os endereços... se alguem quiser pode calcular de novo)
(RDM: ew recalculei e tá certo)

6. Porque o tamanho das páginas é sempre potência de 2?
R: Pois a memória é dividida blocos de tamanho fixo chamados de frames-molduras de páginas e as páginas tem que ser construídas de forma a se encaixar melhor nessas molduras.
RDM(mesma coisa, só um complemento): A memória física é dividida em blocos de tamanho fixo(frames-molduras de páginas) sempre em tamanhos de potência de 2 e entre 512 ~ 8192 bytes. Por isso a memória lógica também deve ser dividida em blocos(páginas) de mesmo tamanho que a da física.

7. Considere um espaço de endereçamento lógico de 8 páginas de 1K cada, mapeados em uma memória física de 32 frames.
a) Quantos bits tem o endereço lógico?
R: Endereços lógicos (virtual) são formados por:
- o numero da página + o deslocamento (offset) dentro da página
numero da página: 8 = 2^3 $\Leftrightarrow$ 3 bits
deslocamento dentro da página: 1024 = 2^10 $\Leftrightarrow$ 10 bits
3 bits + 10 bits = 13

b) Quantos bits tem o endereço físico?
R: 32/8 = 4 = 2^2 = 2bits //numero total de páginas da memória física dividido pelo numero de paginas
2+10 = 12 bits

(a questão (a) eu tenho quase certeza que está certa, a (b) nem tanto...) pq 32/8? pq é o numero total de paginas da memória dividido pelo numero de paginas...

8. Considere um sistema com paginação onde a tabela de páginas está na memória:
a) Quanto tempo leva uma referência a memória paginada se uma referência a memória leva 150ns?
R: 150ns para buscar na tabela + 150ns para buscar a referencia = 300ns , no melhor caso.
(eu não faço idéia se isso está certo)

b) Qual o tempo de uma referência a memória se tivermos uma memória associativa que tem sucesso em 75% das vezes? (o tempo da memória associativa é 15% do tempo normal)
R:

9. Considere um computador com 32-bits de endereço usando tabela de páginas em 2-níveis. Os endereços virtuais são compostos por um campo de 9-bits para o nível 1, 11-bits para o nível 2 e um deslocamento. Qual o tamanho das páginas e quantas páginas

podem existir no endereço lógico?

R:

p1	p2	offset
9bits	11bits	12bits

Já que o offset é 12 bits o tamanho das páginas são de 2^12 = 4kb (isso é de certeza)
O número de páginas que podem existir é (2^9)*(2^11) = 2^20 = 1mb
(referência http://wiki.icmc.usp.br/images/5/5a/Aula11.pdf)

10. Uma máquina tem um endereço lógico de 48-bits, endereço físico de 32-bits e páginas
de 8K. Quantas entradas são necessárias para uma tabela de páginas convencional? E para uma tabela invertida?

R:

11. Porque segmentação e paginação são combinadas em um novo esquema?

R: Se os segmentos são grandes, talvez seja inconveniente (ou mesmo impossível) mantê-los em memória na sua totalidade. Isso gera a ideia de paginação dos segmentos, de modo que somente as páginas realmente necessárias terão de estar na memória. Cada segmento é um espaço de endereçamento comum, sendo também paginado do mesmo modo que a memória paginada não segmentada.

12. Quando usamos segmentação e paginação, primeiro consultamos o descritor de segmentos e depois o descritor de páginas. No caso da memória associativa, ela funciona da mesma forma? Com dois níveis de consulta?

R: Não, pois parte da tabela está em hardware. (não faço a minima idéia do que falei.
Referencia: http://www.dimap.ufrn.br/~ivan/SO/SO-aula10.pdf)

13. Quando ocorre uma falta de página (page fault)? Descreva as ações do SO em uma page fault?

R: - O sistema operacional descobre a ocorrência de uma falta de página e tenta descobrir qual página virtual é necessária.
- Uma vez conhecido o endereço virtual que causou a falta da página, o sistema verifica se esse endereço é válido e se a proteção é consistente com o acesso.
-Se existe algum frame livre aloca. Senão seleciona um frame para realizar a troca de página.
- Se o frame da página selecionada estiver sujo, a página é escalonada para ser transferida para o disco e será realizado um chaveamento de contexto.
- Quando o frame da página estiver limpo, o sistema operacional buscará o endereço em disco onde está a página virtual solicitada e escalonará uma operação para trazê-la.
- O processo em falta é escalonado, o sistema operacional retorna para a rotina, em linguagem de máquina, que o chamou.

14. Considere uma sequência de referências a páginas para um processo com m frames (vazios inicialmente). A sequência tem tamanho p com n páginas distintas ocorrendo. Para um algoritmo de troca de páginas:
a) o que é um limite baixo de page fault?
R: Um limite baixo neste caso é m page faults, pois os frames estarão vazios inicialmente e para ocupar cada frame ocorrerá obrigatoriamente uma falta, portanto no melhor dos casos ocorrerá m faltas e todas as outras requisições já se encontraram em algum dos frames.

b) o que é um limite alto de page fault?
R:Um limite alto é p page faults, pois no pior dos casos toda requisição da sequência resultará numa falta.

15. Se usarmos um algoritmo de troca de páginas FIFO com 4 frames e 8 páginas, quantos page faults vão acontecer na seguinte sequência de referências: 0 1 7 2 3 2 7 1 0 3 se os 4 frames estão inicialmente vazios.
R:

0	0	0	0	**3**	3	3	3	3	**3**
-	**1**	1	1	1	1	1	**1**	**0**	0
-	-	**7**	7	7	7	**7**	7	7	7
-	-	-	**2**	2	**2**	2	2	2	2

Ocorreram 6 page faults.

16. Repita o problema anterior usando LRU.
R:

0	0	0	0	**3**	3	3	3	**0**	**3**
-	**1**	1	1	1	1	1	**1**	1	1
-	-	**7**	7	7	7	**7**	7	7	7
-	-	-	**2**	2	**2**	2	2	2	2

Ocorreram 7 page faults.

17. Considere o seguinte array:
var A: array[1..100] of array[1..100] of integer;
onde A[1][1] está na posição 200 em um sistema com memória paginada com páginas de tamanho 200. O processo que manipula a matriz está na página 0 (0-199), desta forma as

instruções são buscadas na página 0.
Considerando 3 frames, quantas page fault são geradas pelos seguintes códigos de inicialização do array, usando LRU, e sabendo que o frame 1 tem o código de inicialização e os outros dois frames estão vazios inicialmente.

a) for j:= 1 to 100 do
    for i:= 1 to 100 do
        A[i][j]:=0;

b) for i:= 1 to 100 do
    for j:= 1 to 100 do
        A[i][j]:=0;

18. Considere a utilização de uma política de troca de páginas que regularmente examina cada página e descarta aquela que não foi usada desde a última vez que foi feita o exame. O que voce ganha e o que voce perde usando esta política ao invés de LRU ou segunda
chance?
R:

19. Considere um sistema de computação com paginação por demanda onde o grau de multiprogramação é fixado em 4. O sistema foi recentemente medido para determinar a utilização da CPU e do disco de paginação. Os resultados podem estar entre os listados abaixo. Para cada caso, o que está acontecendo? O grau de multiprogramação pode ser aumentado para aumentar a utilização da CPU?
a) utilização da CPU 13%; utilização do disco 97%
R: Esse caso está usando muito o disco provavelmente por buscar muitas instruções que não estão na memória, consumindo muito tempo para busca e pouco para processamento.
b) utilização da CPU 87%; utilização do disco 3%
R: Esse caso parece ideal pois a CPU está trabalhando maior parte do tempo sem a necessidade de muitas buscas ao disco.
c) utilização da CPU 13%; utilização do disco 3%
R: Nesse caso a multiprogramação pode ser aumentada para aproveitar melhor a CPU.

20. Qual é a causa do thrashing ? Como o sistema detecta o thrashing ? Uma vez detectado o que o sistema pode fazer para eliminar o problema?
R: A causa do thrashing é um programa que frequente e continuamente gera falta de página. O sistema pode detectar esse problema monitorando as faltas dos processos. Esse problema pode ser resolvido mudando o algoritmo de substituição de pagina. Uma solução é adicionar mais memória (referência: http://en.wikipedia.org/wiki/Thrashing_(computer_science)

21. Um processo tem 4 frames alocados. O quadro a seguir mostra as informações referentes aos mesmos.

No. Pag. Virt.	Frame	Tempo carga	Tempo referenc.	Bit R	Bit M
2	0	60	159	1	1
1	1	130	160	0	1

0	2	26	162	1	0
3	3	30	163	0	0

Um page fault para a página virtual 4 ocorre. Qual frame será trocado para os seguintes algoritmos de troca:

a) FIFO (first-in-first-out) __1__.
b) Clock (segunda chance) _1_
c) LRU (least recently used) _1_
d) Algoritmo Ótimo _3_

22. Um SO observa o conjunto de páginas lógicas em uso pelos processos para decidir se pode ou não iniciar a execução de um novo processo. Esse sistema define, como conjunto de trabalho, o conjunto das páginas acessadas no último intervalo completo do timer. Osistema dispõe de 20 páginas físicas. A história do sistema no último intervalo completo do timer é mostrada abaixo. Quantas páginas físicas o sistema poderá fornecer a um novo processo, mantendo ainda o conjunto de trabalho dos processos atuais?

P0 (4,5,3,6,3,4,5,3,3,6)
P1(1,7,8,8,1,3,1,7,8,9)
P2(6,7,8,5,8,6,6,7,5,5)

23. Qual problema pode ocorrer quando um periférico pode acessar diretamente a memória e utiliza-se memória virtual? Como resolve-lo?

R: A memória pode conter um dado que não é referente a aplicação que o periférico está se referindo, pode-se resolver este problema alocando uma área de memória para o periférico ou fazendo-o verificar se o dado é válido. (sem embasamento nenhum)