INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CEARÁ

ENGENHARIA DE COMPUTAÇÃO
SISTEMAS OPERACIONAIS
SEMESTRE 2012.2 – 2ª Prova 2ª Etapa
PROF. FERNANDO PARENTE GARCIA

ALUNO: Denis Rainer

1) (2,0 Pontos) Sobre memória, responda:
   a) Que limitação teria um sistema operacional que não utilizasse relocação?
   b) Que limitação teria um sistema operacional que não utilizasse paginação (*swapping*)?

2) (2,0 Pontos) Imagine três processos que iniciem simultaneamente, sendo que cada um precisa de 5 minutos de tempo de execução de CPU. Suponha que cada processo é bloqueado durante 60% do tempo. Quanto tempo o último deles levará para executar até terminar se eles executarem sequencialmente? E se eles executarem em paralelo em uma máquina com um processador de apenas um núcleo? E se eles executarem em paralelo em uma máquina com um processador de quatro núcleos?

3) (2,0 Pontos) Suponha que a tabela de páginas para um processo atualmente executando no processador seja como mostrada na tabela abaixo, e que o processo possua 32K bytes de memória virtual e 16K bytes de memória real, com frames e páginas de 4K bytes de tamanho. O algoritmo de substituição de páginas utilizado pelo sistema operacional é o NUR. O processo executa uma operação de escrita no endereço 10000 e uma operação de leitura no endereço 30100, respectivamente. Quais os endereços físicos correspondentes a cada um destes endereços virtuais?

Página	Frame	Bit R	Bit M	Instante de Carga
4	0	1	0	140
6	1	1	0	130
5	2	1	1	110
2	3	0	1	100

4) (2,0 Pontos) Explique detalhadamente como funcionam os esquemas RAID 0, RAID 1 e RAID 3. Em seguida, responda:
   - Qual destas estruturas apresenta maior desempenho? Justifique.
   - Qual destas estruturas apresenta maior confiabilidade? Justifique.
   - Qual destas estruturas apresenta menor custo? Justifique.

5) (2,0 Pontos) Explique detalhadamente como funcionam os sistemas de arquivo que utilizam os métodos de alocação contígua e alocação com lista ligada. Qual destes métodos apresenta melhor desempenho quando se deseja gravar dados no final de um arquivo existente? Justifique.

BOA PROVA!

① a) Caso a relocação não fosse implementada os endereços deveriam ser setados no processo de escrita do código ou mesmo no processo de compilação, assim cada processo poderia ser alocado em apenas uma região da memória, sempre. 1,5

b) ~~Cada processo teria de ficar na memória por completo, tendo de ser executado até sua finalização.~~ 1,0

② O tempo de bloqueio de cada um é: 5 × 60% = 3 min

Executando sequencialmente teremos, 8 min de execução de cada processo, para 3 processos 24 min.

Em paralelo, o tempo de execução é dado por: $1 - (0,6)^3 = 1 - 0,49$

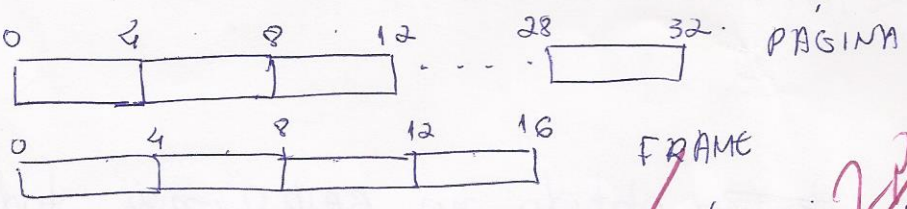
Assim, o tempo ~~total é dado por~~ de ~~execução~~ CPU de cada processo é

$\frac{51\%}{3} = 17\%$.

O tempo de execução é dado por = $\frac{5}{17\%} =$ 29,4 min

Como o número de ~~CPUs~~ cores é maior que o número de processos, cada processo é executado por um core. Então, os 3 processos terminariam em 8 min.

③



offset = 10000 − 8192 = 1808 (página 2).

endereço de memória = 14096. (frame 3) → bit R = 1.

end virtual = 30100. (página 7)

offset = 30100 − 28672 = 1428.

Como a página 7 não está mapeada na memória uma substituição deve ser feita. Pelo NUR, o ~~página~~ frame escolhido é o 1, por estar a mais tempo na memória, em relação ao frame 0. A escolha é feita desta forma pois os dois frames estão no caso 2, do NUR.

end. de mem = 4096 + 1428 = 5524

(7) RAID 0: não é exatamente um RAID, pois o conceito de ~~RAID~~ é de que haja redundância. Os dados ao invés de ~~copia~~ escritos ~~dos~~ todos em um ~~driver~~ disco (HD), são ~~dem~~ quebrados e ~~copi~~ ~~ados~~ escritos em mais de um ~~HD~~ disco. Isto aumenta a velocidade de escrita. Mas como não há redundância, a segurança é menor do que ~~em~~ na escrita em um SGBD.

RAID 1: Os dados são quebrados e escritos em mais discos, ~~cada parte em um disco,~~ mas neste caso há redundância, pois cada ~~disco que contenha uma parte~~ parte do arquivo é copiado em dois discos, ~~ou seja cada~~ caso haja a perda de 1 disco é feita a ~~substituição~~ deste.

Confiabilidade duas vezes melhor que o RAID 0.

VAI MAIS DISCOS

RAID 3: Cada bit é armazenado em um disco, ~~caso ha~~ com um bit de paridade, caso ocorra uma quebra em algum disco do conjunto dos bits, é feito um calculo para se encontrar o bit ausente.

AVOC SI TRANSFORMOS

O melhor desempenho é ~~visto no~~ obtido no RAID 0, pois todos os discos são usados no paralelismo.

Melhor confiabilidade, RAID 1, pois há uma espécie de backup de cada ~~RAID~~ disco.

(5) alocação contígua: o arquivo utiliza blocos que estão em sequência. A implementação é simples pois necessita-se apenas do endereço do bloco inicial e do número de blocos utilizados pelo arquivo. Tem como desvantagem o fato de que, ao ~~finalizar~~ fechar um arquivo, os blocos onde estes estavam armazenados ficam livres e se faz necessária uma compactação, que demanda um tempo muito longo. Uma leitura sequencial é rápida, mas uma leitura aleatória é lenta pois é preciso percorrer todos os blocos até encontrar o bloco onde inicia o arquivo.

alocação com lista ligada: neste método o ~~diretório~~ sistema de arquivos referencia apenas o bloco inicial e a partir deste os blocos são referenciados por ponteiros, que são a primeira palavra de cada bloco. ~~Neste processo os~~ Utilizando este método o arquivo pode ser armazenado em qualquer bloco independente da posição. Como desvantagem, tem o fato de os blocos não serem mais potências de 2, pois uma parte do bloco é ocupada pelo ponteiro. A leitura sequencial é rápida, mas a aleatória é lenta pois caso o arquivo começe no bloco n é preciso ler os n-1 blocos anteriores.

Caso houvesse ~~espaço~~ blocos livres a alocação contígua seria melhor. como em geral os discos são fragmentados, uma melhor opção é aquela onde todos os blocos podem ser usados. Então a melhor opção é a alocação com lista ligada

(6) cont.

Melhor custo, RAID 3, pois apresenta uma confiabilidade razoável ~~com~~ com o uso de quase todos os discos no paralelismo.