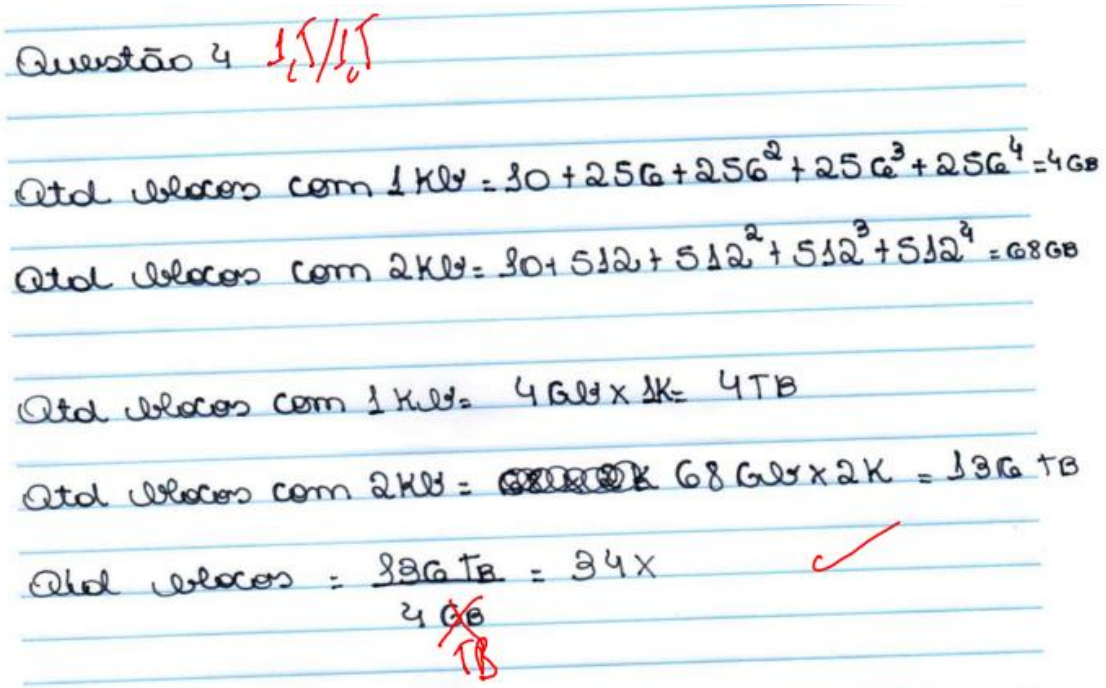
3) Qual a vantagem da lista de blocos livres em relação ao bitmap de blocos livres? Explique.

R: Ela requer menos espaço, e ela armazena apenas um bloco de controle no disco.

4) Em uma versão do Unix que utilize indireções sendo a última a indireção quadrupla, quantas vez o tamanho máximo de um arquivo será aumentado quando o tamanho do bloco for dobrado? Explique (ou calcule).



5) Um fabricante de discos magnéticos decidiu aumentar o desempenho de seus discos aumentando o número de setores por trilha (mantendo igual o número de bytes do setor). As demais características dos discos permaneceram inalteradas. Explique o que ocorreu nos três componentes do tempo de acesso ao disco (considerando tempos médios quando não houver informação adicional).

O tempo de acesso é a soma dos três tempos que são: tempo de seek, tempo de latência e tempo de transferência. Tempo de seek é o nome dado ao tempo do movimento da cabeça magnética ao longo do eixo até chegar na trilha onde está o setor que se quer acessar, setor que se quer ler ou escrever. Tempo de latência, é o tempo que a cabeça começa a passar sobre o setor que se quer acessar. Tempo de transferência: é o tempo que demora para cabeça passar ao longo do setor.
Como houve aumento do número de setores por trilha, o tempo de transferência nesse caso diminui, Já o seek e a latência permanecem iguais.

1) Dos cinco tipos de alocação de arquivo estudados no curso, ordene estes tipos (explicando a ordenação) segundo o menor custo computacional (em termos processamento de CPU necessário) para o acesso aleatório.

Em ordem crescente de desempenho:
Encadeada, FAT, indexada e contínua.
Tanto a encadeada quanto a FAT, utilizam a ideia de ligar os blocos por referências encadeadas entre eles. Portanto, em um acesso randômico, todo acesso será necessário percorrer a lista toda até se chegar no bloco desejado. Por esse motivo, elas têm o pior desempenho.

A encadeada é pior que a FAT pois, enquanto a FAT guarda o encadeamento completo separado, a encadeada guarda a informação do encadeamento nos próprios blocos de conteúdo, sendo necessário a varredura passando por vários blocos (muitas leituras) até chegar no bloco desejado.

Além disso, a FAT é guardada na memória, fazendo com que a busca pelo bloco desejado seja muito mais rápido que a busca utilizando a alocação encadeada.
A contínua, apesar de limitações no crescimento dos arquivos é a mais rápida ao acesso, pois seu controle é muito simples (nº bloco inicial, quantidade de blocos), sendo necessários apenas pequenos cálculos para saber onde achar o bloco a partir do primeiro.
A indexada também precisa de pequenos cálculos para saber qual o número do bloco do arquivo na visão lógica, porém ainda é preciso acessar uma estrutura de controle (ex. inode) que mapeie os blocos pelo índice na memória física. Por isso é mais rápido que a FAT e encadeada e menos rápida que a contínua.

2) Explique, citando um exemplo com FAT, como as inconsistências de arquivo ocorrem.

O bloco está marcado como livre na estrutura de blocos (bitmap, lista de blocos livres ou FAT), mas este bloco está presente na estrutura de controle de alocação de algum arquivo (indexada, extensão ou FAT). Nesse caso, o conteúdo desse arquivo corre o risco de ser sobrescrito se esse bloco for alocado para um outro propósito.

3) Explique como o conceito tradicional de setor circular foi modificado nos discos atuais, descrevendo como é feito atualmente. Explique a vantagem de fazer desta forma.

Antigamente se tinha o conceito de fatia de Pizza, na atualidade todos os setores possuem o mesmo tamanho, tendo como vantagem uma maior capacidade de armazenamento e uma maior velocidade de transferência nos setores mais externos.

4) A E/S através de verificação de status tem duas variações em termos do tempo que demora para ser feita cada verificação de status. Explique a desvantagem de cada uma dessas variações (obs. cada variação teve ter pelo menos uma desvantagem).

Polling Radical (chamada de Espera ocupada)
Nesse mecanismo A CPU não faz nada, exceto perguntar o status do dispositivo. A desvantagem desse mecanismo é que a CPU fica ocupada, o seu uso é contínuo até que seu status mude, o que faz com que a CPU seja usada para algo não muito útil.

Polling: Nesse mecanismo ocorre a verificação do status do dispositivo de vez em quando, e só faz o envio e recepção dos dados quando o status muda. Tem como desvantagem a perda de tempo em que o dispositivo já pode ter os dados sem o SO perceber.

5) Explique uma chamada ao sistema operacional no Unix, cuja existência é necessária em função do conceito de posição corrente do arquivo. Esta chamada pode provocar a existência de arquivos "pouco comuns", explique o que caracteriza esses arquivos.

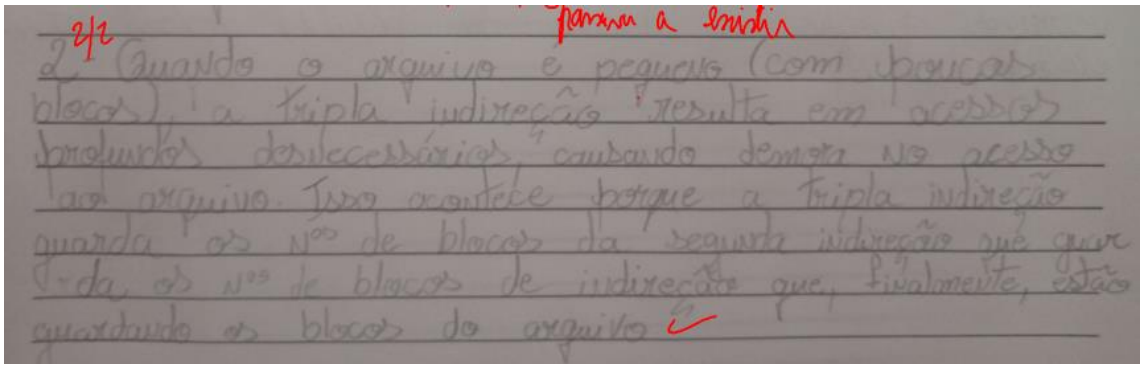
Exemplo de uso para arquivo esparso:
Matriz de inteiros 1024x1024, com 1 inteiro = 4 bytes (se a máquina for 32 bits)
Qtd_de_elementos = 1024.1024 = 1k.1k = 1M elementos
Memória gasta pela matriz = Qtd_de_elementos.tam_dos_elementos = 1M . 4 bytes
Memória gasta pela matriz = 4MB = 4096 bytes
Se essa matriz for do tipo que a maioria das linhas possua o valor 0, ela é chamada de matriz esparsa. Ela pode ser salva da maneira trivial como uma variável M no disco, ou como um arquivo esparso, que ocupará menos espaço no disco.

Isso pode ser feito usando a seguinte chamada:

```
int m[1024][1024];
fd = create(“matriz”,...); //chamada create também retorna um inteiro
for (i=0;i<1024;i++){
if (linha_so_tem_zeros(m[i]))
lseek(fd,tam_linha,seek.atual)//tam_linha = 4096 nesse exemplo
else
write(fd,m[i],4096)
}
close(fd);
```

O que caracteriza isso é o Arquivo esparso que é quando o arquivo tem pedaços que não existem, e esse espaço é preenchido com números 0 binários. O arquivo esparso tem um tamanho lógico que ele é maior que o espaço ocupado no disco, só que isso normalmente não é verdade.

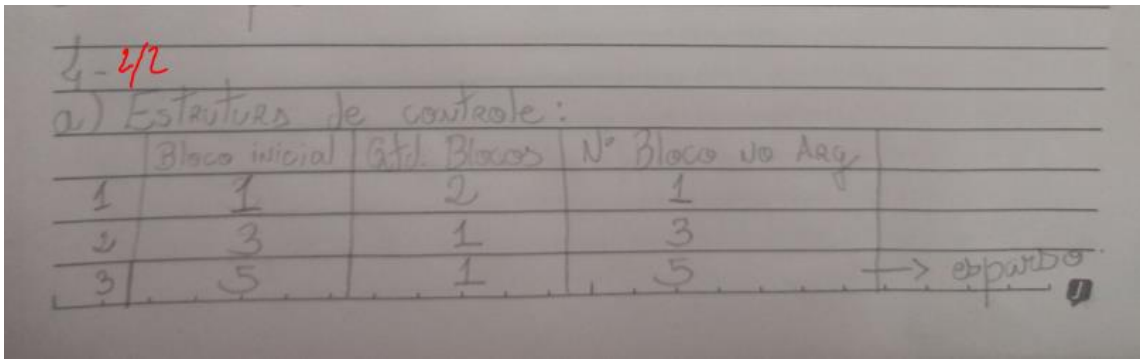
2) Por que a alocação indexada no Unix não usa apenas indireção tripla (sem possuir indireção simples e dupla)? Dica: arquivos de tamanho pequeno



3) Quando o SO se comunica com o disco considerando-o como sendo um vetor de setores, qual é o problema (pequeno) que pode ocorrer na leitura de um arquivo contínuo todo para a mémória? Explique

Você explica que pra ler um bloco tem que carregar todo pra memoria, explica a questão do espaço ocupado

4) Mostre exemplo (utilizando a estrutura de controle de alocação) de arquivo esparso, em dois casos: A) Alocação por extensão e b) Alocação encadeada



5) Se um SO similar ao Unix não dispor do conceito de posição atual (corrente) de um arquivo, quais as modificações que precisarão ser feitas nas chamadas do Unix? Qual chamada do Unix poderá ser suprimida?

2/2

5 - As modificações necessárias são nas instruções de read e write que precisam passar a receber a posição do arquivo como parâmetro. A chamada de lseek pode ser suprimida porque ela altera a posição atual do arquivo e, já que o SO não tem esse conceito, essa chamada perde a necessidade. ✓