

Iniciado em terça, 22 fev 2022, 15:12

Estado Finalizada

Concluída em sexta, 25 fev 2022, 20:13

Tempo empregado 3 dias 5 horas

Avaliar Ainda não avaliado

Questão 1

Completo

Vale 1,00 ponto(s).

1. Rode o comando free em um terminal, usando argumentos como -m (para valores em Megabytes). Quanta memória tem no seu sistema? Quanto dessa memória está livre? Esses são os valores que você esperava ver?

Como o linux foi recém instalado, a quantidade de memória disponível é de 3538, a qtd de memória é 7865 e achei que a quantidade de memória disponível seria menor.

O comando free é importante para entender melhor a gerência de espaço livre do sistema operacional.

Questão 2

Completo

Vale 1,00 ponto(s).

2. Crie um programinha em C chamado usaMemoria.c, que recebe como parâmetro um valor entre 0 e 4 bilhões e aloque um vetor com essa quantidade de bytes. Na sequência o seu programa deve ter um loop infinito onde uma variável auxiliar acessa sequencialmente todos os elementos deste vetor repetidamente (por exemplo aux = v[i]).

Rode o seu programa e, em outro terminal, rode o comando free. O quanto mudou o consumo de memória? Foi o esperado? E o que acontece quando você mata o usaMemoria? Repita isso para diferentes tamanhos de memória. O que acontece quando o tamanho alocado é muito grande (>1GB)?

Com 1 bilhao deu falha de segmentação, e a memoria usada variou entre 3243 pra 3244 ou 3245.

Questão 3

Completo

Vale 1,00 ponto(s).

3. Agora vamos utilizar o comando pmap: Ele mostra como está a memória virtual de um processo. Vamos ver como ele funciona.

Faça o comando ps aux para ver os processos que estão rodando na sua máquina. Se você achar um programa grande como um navegador Web, rode pmap <identificador do programa> (PID, que é mostrado pelo ps aux). O que ocorreu de diferente entre o espaço de endereçamento mostrado no vídeo e o do pmap, especialmente relacionado ao trio code/heap/stack?

em relação ao code/heap/stack, mantem a mesma ordem, o endereço aumenta, mas os valores alternam.

Questão 4

Completo

Vale 1,00 ponto(s).

4. Rode o seu programa usaMemoria com diferentes tamanhos de memória e cheque o que aconteceu com o pmap -X. Foi o que você esperava? O seu programa mostrou um [heap]?

Não mudou e não mostrou heap.

Questão 5

Não respondido

Vale 1,00 ponto(s).

5. O seu programa mostrou um [heap]? Se não mostrou, adicione uma alocação de memória pequena no meio do loop principal. O que aconteceu agora?