Avaliação Trabalho Prático de Software Básico

Caio Rufino (GRR 20224386), Frank Hannemann (GRR 20224758), Henrique Biehl (GRR 20221257)

(a - .5) Se você pudesse voltar no tempo, o que você (de hoje) recomendaria a você (do primeiro dia de aula de Software Básico) para minimizar o sofrimento do desenvolvimento deste trabalho?

Uma recomendação seria aprender direito como funciona a função cmp, porque em certos momentos ela é confusa e boa parte dos erros iniciais se deram pelo fato de estarmos utilizando ela de forma errada. Além disso, muito cuidado e atenção durante a implementação da pilha, no caso aperfeiçoando o conhecimento de como ela funciona e, durante a implementação do trabalho, visualizar por meio de desenho como está o comportamento da pilha. Isso pois erros associados ao uso da pilha causaram problemas durante a implementação. Outra recomendação seria fazer os exercícios que tem todo final do slide logo após toda aula, isso facilitaria muito no aprendizado.

(b - .5) O que você recomendaria ao professor da disciplina quando ele se preparar para o próximo semestre remoto a fim de aumentar o grau de absorção do conteúdo da disciplina por parte dos alunos?

Seria útil complementar a sumarização das aulas que o professor já estava fazendo, afinal, era uma maneira muito útil de sintetizar todo o conteúdo visto em apenas um lugar, e auxilia bastante na hora de fazer os trabalhos e estudar para as provas. Outra coisa seria que, na hora de falar sobre as bibliotecas, ensinar antes a diferença entre todas e como elas operam e só depois mostrar como usá-las. Creio que ficou meio confuso aprender como se usa antes mesmo de aprender direito como a biblioteca funciona.

(c - 2) Explique quais os trechos de código e quais as principais alterações que você fez para que a segunda parte funcionasse, ou indique o motivo de você não ter conseguido terminar a alteração. Indique, por exemplo, quais as linhas de código que você mudou e com qual objetivo.

As alterações foram feitas apenas na função memory_alloc, onde o que foi mudados foi:

1 - Uso de uma variável local extra (biggest_empty_block) que armazena a posição de memória (iniciando no registro de dados) do maior bloco livre encontrado na heap.

2 - Mudança no laço do while, onde esse laço, ao invés de procurar o first-fit e alocar a memória, busca o maior bloco livre.

3 - As verificações se o bloco tem tamanho para comportar a alocação ficaram após o laço while. Nesse processo, ao invés de usar da variável que possui o endereço da iteração na brk (na versão antiga chamada de tmp_brk) é usado o biggest_empty_block. Também é necessário verificar se biggest_empty_block é diferente de current_brk, para situações de primeira alocação, onde ainda não há nenhum bloco alocado. É necessário verificar, novamente, se o bloco apontado por biggest_empty_block está ou não sendo usado. É repetida essa verificação pois, se o maior bloco for o inicial, não há nada que garanta que ele não está sendo usado - pois essas verificações ocorrem apenas no laço while.

4 - Feitas essas verificações é analisado se o tamanho do bloco de biggest_empty_block menos os bytes requeridos são maior ou igual a 17, para poder criar um novo bloco. Isso foi feito dessa forma pois se for comparado o tamanho do bloco - bytes - 16 >= 1 há risco desse cálculo dar um valor negativo, o que altera substancialmente a lógica do programa.

5 - O resto do fluxo segue normalmente como na versão anterior com a criação desse novo bloco ou apenas a marcação do bloco atual (biggest_memory_block) como usado. A alocação de um bloco novo (aumento da heap) não sofreu alterações.