

Respostas implementação Worst Fit em x86

Felipe Quaresma e Marcelo Schreiber

Novembro, 2023

Respostas

Pergunta (a)

Recomendaria fortemente a leitura de livros que passam, de forma didática, o conteúdo da matéria. Dessa forma, teríamos uma maior base de informação prévia e possíveis dúvidas que, ao longo das aulas de Software Básico, pudessem ser esclarecidas. Assim, os trabalhos e as provas não apresentariam tanta dificuldade e conseguiríamos lidar melhor com a questão da conciliação de várias matérias, estágio, entre outros.

Pergunta (b)

Recomendo ao professor da disciplina a inclusão de projetos práticos menores ao longo do próximo semestre remoto. Essa abordagem, em comparação com um único projeto extenso, tem se mostrado mais eficaz para manter o engajamento dos alunos e aprimorar a absorção do conteúdo. Projetos como a implementação de um simulador de memória virtual ou um leitor de arquivos ELF podem servir como ponte entre a teoria e a prática, proporcionando uma compreensão mais sólida e envolvente dos conceitos discutidos em sala de aula.

Pergunta (c)

Basicamente, no modelo first fit, nós fizemos um loop chamado `percorre_brk` que ia de alocação em alocação vendo se o bloco estava livre, depois se o tamanho era maior ou igual ou menor ao que queríamos alocar. Assim, seguíamos com o loop até achar um espaço alocável que condiz com os requisitos mencionados no trabalho ou até o `brk_inicial` que é colocado em um registrador e somado a cada iteração ficasse igual ao `brk_atual`. Nesse caso, criamos um bloco novo no fim da heap.

Para a mudança de first fit para worst fit, mudamos um pouco a dinâmica do loop. Esse fato decorre da alteração de, ao invés de parar o loop e alocar o bloco no primeiro bloco disponível com tamanho superior ou igual ao que passamos como argumento, o loop continua até ler todos

os blocos alocados e desalocados e guardando o endereço do bloco de maior valor de tamanho encontrado e o próprio valor de tamanho em um registrador. Com isso, podemos, depois de finalizar o loop (`brk_atual = brk_inicial`), voltar para aquele endereço e alocar o bloco novo ali. Caso não tenha nenhum bloco de tamanho suficiente para que o tamanho requisitado como argumento seja alocado, o programa cria um bloco novo no final da heap.