

Iniciado em Tuesday, 17 May 2022, 18:49

Estado Finalizada

Concluída em Tuesday, 17 May 2022, 18:56

Tempo empregado 7 minutos 2 segundos

Avaliar 10,0 de um máximo de 10,0(100%)

Questão 1

Correto

Atingiu 1,0 de 1,0

1. Tanto no EPOS quanto em sistemas Unix-like, a heap da aplicação fica localizada . Ela é inicializada tendo como base , que é incluído no arquivo ELF pelo .
2. A utilização de *named enums* com versão "placement" do operador *new* permite . O EPOS a utiliza para criar classes específicas de alocadores para propósitos específicos e eliminando métodos como *kmallocc* .

Questão 2

Correto

Atingiu 6,0 de 6,0

Considerando a versão atual do código no repositório GIT do EPOS, responda as seguintes perguntas:

1. Se o EPOS for configurado como *multiheap*, o operador *new* fará alocações de memória a partir da e o operador *new (SYSTEM)* fará alocações de memória a partir da . Por sua vez, caso seja configurado sem *multiheap*, as alocações se darão a partir da , a qual é inicializada em de forma a conter .
2. O atributo *char Application::_preheap[]* é utilizado para . Ele é utilizado para .
3. Quantas vezes a artimanha de reservar memória através do atributo *_preheap* é utilizada no EPOS? (apenas o número).
4. A função *memset* é chamada em várias regiões do código do EPOS, sua principal função é .
5. A política de alocação utilizada nesta versão do EPOS para o operador *new* é . Os blocos de memória alocados . Este comportamento é definido em qual método da classe ? (apenas o nome do método, sem classe, sem parâmetro e sem parêntesis).
6. A presente implementação de Heap armazena metadados sobre o bloco alocado . Os metadados incluem o tamanho do bloco alocado e .
7. O construtor da classe Heap opcionalmente invoca o método *free()* para .
8. O método *Heap::free()* executa uma chamada para o método *Grouping_List::insert_merging()*, que, por sua vez, busca dois elementos na lista, usando os métodos *search()* para e *search_left()* para .
9. Considerando-se uma heap do EPOS com o estado definido como [(a, O, 3) , (b, F, 5) , (c, O, 4) , (d, F, 6) , (e, O, 3)] onde cada tupla é formada por um ponteiro, um símbolo que determina se o bloco está ocupado (O) ou livre (F) e o tamanho do bloco, respectivamente. Considerando-se ainda que o ponteiro "b" aponte para o bloco de memória imediatamente após o bloco apontado por "a", que o ponteiro "c" aponte para o bloco de memória imediatamente após àquele apontado por "b" e que os ponteiros "d" e "e" tenham sido inicializados de forma similar, ao invocar-se *Heap::free(c)*:
 - O ponteiro é removido da lista de blocos livres e o(s) ponteiro(s) restante(s) nesta lista será(ão) (em caso de mais de um ponteiro, separe por vírgula e sem espaço);
 - O tamanho do bloco ao qual "c" foi acrescentado, após a operação de *Heap::free(c)* ser concluída, é ;
 - O número de elementos na lista de blocos livres da heap, após a operação de *Heap::free(c)* ser concluída, é (responda com um número)

Questão 3

Correto

Atingiu 3,0 de 3,0

Considerando a versão vigente do código n repositório GIT do EPOS, modifique a aplicação "Producer x Consumer" (app/producer_consumer.cc) para que a mesma seja executada sobre a plataforma **RV32/RISCV/SiFive_E SEM multiheap**, e responda as seguintes perguntas:

1. Endereço na memória do objeto Thread correspondente à thread main: ✓ .

2. Endereço na memória do objeto Thread correspondente à thread cons: ✓ .

Agora, modifique a aplicação para executar **COM** multiheap e responda as seguintes perguntas:

3. Endereço na memória do objeto Thread correspondente à thread main: ✓ .

4. Endereço na memória do objeto Thread correspondente à thread cons: ✓ .

5. Durante a inicialização da Heap da Aplicação identifique:

a) Endereço da Heap da Aplicação: ✓ .

b) Endereço no qual será feito a primeira alocação na Heap da Aplicação (endereço gerado por: endereço da head + tamanho da head): ✓ . Observe que a head da Grouping List deve estar como pública na Heap, note que o endereço retornado ao primeiro objeto alocado já considera o tamanho do objeto em si, a pergunta então faz referência a um endereço antes do decremento do tamanho do objeto.

◀ OSDI with EPOS: Memory Management

Seguir para...



E6b: OSDI with EPOS: Memory Management (coding) ▶