Trabalho - 2012/2

Descrição

O objetivo deste trabalho é implementar um gerenciador de memória da heap. Este gerenciador deverá ser inteiramente implementado em assembly e deverá conter três funções conforme descritas no arquivo-cabeçalho abaixo.

```
// Arquivo meuAlocador.h
//
//

void *meuAlocaMem ( int num_bytes );
void  meuLiberaMem ( void*  ptr );
void  imprMapa ( );
```

A função "meuAlocaMem" aloca um espaço de <num_bytes> na heap e retorna o endereço do primeiro dos <num_bytes> bytes usáveis. A função "meuLiberaMem" recebe como argumento um endereço retornado por "meuAlocaMem" e libera aquele espaço. Em ambas as funções deve ser utilizada a chamada de sistema "brk" quando necessário.

O livro "Programming from the ground up" apresenta um exemplo de gerenciador de memória semelhante, porém com duas funções a mais: iniciar e finalizar alocador. Estas funções não são utilizadas aqui.

A função "imprMapa" imprime o mapa do espaço ocupado e livre da heap. Como exemplo, assuma a seguinte seqüência de comandos:

```
...
a = meuAlocaMem ( 200 );
...
b = meuAlocaMem (100 );
...
meuLiberaMem ( a );
...
c = meuAlocaMem ( 50 );
...
imprMapa ( );
...
 meuLiveraMem ( b );
...
imprMapa ( );
```

Considerando a implementação do livro, a impressão seria algo como:

Inicio heap: 0x41000000
Segmento 1: 050 bytes ocupados
Segmento 2: 150 bytes livres
Segmento 3: 100 bytes ocupados
Segmentos Ocupados: 2 / 250 bytes
Segmentos Livres: 1 / 150 bytes

Inicio heap: 0x41000000
Segmento 1: 050 bytes ocupados
Segmentos Ocupados: 1 / 50 bytes
Segmentos Livres: 0 / 0 bytes

IMPORTANTE:

Este semestre (2012/2), os trabalhos devem ser implementados obrigatoriamente em assembly x86 e com uma lista duplamente encadeada.
Implemente também também "calloc" e "realloc" (procure nas man pages).

Será questionado como processa um malloc e um free. Traga uma folha para desenhar o que ocorre e o mapa de memória correspondente.

A função ImprimeMapa () tem como objetivo mostrar o funcionamento interno da implementação para avaliação. O exemplo acima é o resultado da implementação utilizando uma lista ligada.

O resultado apresentado acima da função "ImprimeMapa" contém informações que devem ser apresentado também na implementação deste semestre:

- Endereço de início do nodo na heap;
- Endereço de término do nodo na heap;
- Lista de cada segmento, contendo:
    - Número do segmento;
    - Quantos bytes alocados para aquele segmento;
    - Se é segmento livre ou ocupado;
- Total de segmentos livres, e a soma de tamanhos de todos os livres;
- Total de segmentos ocupados, e a soma de tamanhos de todos os ocupados;

Pontuação extra para quem gerar um mapa de memória. Para ter idéias, procure por imagens para

"memory fragmentation" no google. Minha sugestão: use símbolos como "*" para um determinado número de bytes ocupados e "-" para livres em imprima o mapa usando somente texto (não quero nada gráfico).

A idéia é conseguir visualizar os espaços alocados e os liberados com mais facilidade, o que facilita minha vida na hora da avaliação.

Sugestão:

Este programa pode ser de difícil implementação e de difícil depuração. Por isso, sugiro que primeiro seja implementada a função de alocar e de imprimir o mapa e testar estas duas operações exaustivamente. Somente depois deve ser implementada a função de liberar e novamente deve ser testada exaustivamente. Os testes devem começar com situações simples (alocar 50 bytes, depois 50+100, depois 100-50, etc) e quando estas estiverem estáveis, aumentar para um conjunto maior de elementos. Os resultados obtidos devem ser comparados com os resultados esperados, que devem ser gerados manualmente.

Para garantir a confiabilidade das soluções apresentadas, uma opção e utilizar um gerador automático de alocações/liberações como neste exemplo, onde basta retirar os comentários das linhas cercadas por "// ------ ".