



Atividades

1-Tendo por base as linhas abaixo, e supondo que estejam codificadas dentro da função `memory_init`, responda: a) em que região da memória (texto, dados, pilha, etc.) a string "É a vida vida..." está armazenada? b) O ponteiro `s2` aponta para um endereço que se encontra em qual região? Descreva como você procedeu para encontrar as respostas.

```
char* s = "É a vida...";  
char *s2 = alloc(1);
```

Baseando se no algoritmo disposto neste capítulo tive a conclusão que sendo “s” uma variável do tipo `char` porém contendo uma string, dentro da função `memory init`, a string será copiada para uma página disponível dentro do, `heap`, assim como `*s2` está recebendo o endereço de memória de alocação disposto da função `alloc()`.

figura 1 – trecho da função “`memory_init()`” arquivo `memory.c`

```
124 void  
125 memory_init() {  
126     pages_init();  
127     char* p = alloc(1);  
128     printf("Ptr de alloc(1):%p\n", p);  
129     *p = '0';  
130     *(p+1) = 'i';  
131     *(p+2) = '\n';  
132     char *p2 = alloc(2);  
133     ///!! string literal. p aponta para a seção de texto ou a string é co  
134     ///!! heap e então seu endereço atribuído a p?  
135     p = "0la\n";  
136     printf("Ptr (strg literal):%p\n", p);
```

figura 2 – trecho da função “`alloc()`” arquivo `memory.c`

```
68 void *  
69 alloc(int pages) {  
70     // início da lista dos descritores de página  
71     uint8 * ptr = (uint8 *) HEAP_START;  
72     int i;  
73     int count = 0;  
74     uint8 * desc = 0; // 0 descritor da primeira página do bloco alocado  
75  
76     if (pages == 0)  
77         return 0;  
78     for(i = 0; i < total_pages; i++) {  
79         ptr = (uint8 *) HEAP_START + i;  
80         if(free_page(*ptr)) {  
81             if (count == 0)  
82                 desc = ptr;
```