

Teste de conhecimentos em programação Linux

1. O programa de exemplo abaixo deve inserir um cabeçalho de 12 caracteres em um pacote apontado por packet. Assume-se que a arquitetura é 32 bits e que o parâmetro length é o tamanho do pacote em bytes. *Aponte e comente os erros existentes no código. Entenda-se como problema: bug, código mal estruturado ou confuso. Sugestões de melhoria sempre são bem vindas.*

```
char *insert_header(int *packet, int *length)
{
    int i;
    char header[12] = { 0x80, 0x52, 0x66, 0x61, 0x77, 0x52 };
    /* os outros 6 são 0 */

    if (packet == NULL || length == NULL) {
        return NULL;
    }
    for (i=0; i < *length; i++)
        packet[i+3] = packet[i];

    for (i=0; i < 12; i++)
        packet[i] = header[i];

    *length += 12;

    return packet;
}
```

2. Considerando a função de exemplo abaixo responda as seguintes perguntas. A função funciona? É reentrante? (Porque?) O que a função faz? *Aponte e comente os erros existentes no código. Entenda-se como problema: bug, código mal estruturado ou confuso. Sugestões de melhoria sempre são bem vindas.*

```
void removeEntry (int key)
{
    int noPrev=1, noNext=1;
    list_entry *entry;

    for (entry = List_head; entry != NULL ; entry = entry->next) {
        if(entry->key != key)
            break;
    }

    if (entry == NULL)
        return;

    if (entry->previous != NULL) {
        entry->previous->next = entry->next;
        noPrev=0;
    }
    if (entry->next != NULL) {
        entry->next->previous = entry->previous;
        noNext=0;
    }

    free (entry);

    if ((noPrev) && (noNext)) {
        List_head = NULL;
    } else if (noPrev) {
        List_head = entry->next;
    }
}
```

3. Implemente, usando sua linguagem de preferência, a seguinte máquina de estados, ela representa um protocolo de conexão entre 2 equipamentos (mestre/escravo):

<i>Estado/Evento/Proximo estado</i>	<i>Interface nok</i>	<i>Interface ok</i>	<i>Pacote ok recebido</i>	<i>Pacote nok recebido</i>	<i>Não recebeu resposta</i>
1- Link Down	1	2	1	1	1
2 - Enviando Start	1	-	3	2	2
3 - Start recebido envia Configuração	1	-	4	3	2
4 - Link OK – manda keepalive	1	-	4	2	2

Deve ser implementada uma função que recebe o estado atual e o evento associado. Deverá retornar o próximo estado e realizar as chamadas a funções que realizam as ações referentes a cada estado.