Teste de conhecimentos em programação Linux

*O primeiro grupo de testes consiste de algumas questões teóricas e uma questão de desenvolvimento.*

1. Ponteiros são uma ferramenta de programação bastante importante em C/C++. Explique seu funcionamento e crie um exemplo de utilização.

**Ponteiros são um tipo de variável que armazenam endereços de memória e possuem operadores especiais que permitem que esses valores sejam acessados ou referenciados através dos operadores “\*” e “&”.**

**Podemos fazer com que funções recebam ponteiros de variáveis para uma melhor legibilidade do código e um melhor uso de memória em dispositivos que possuem pouca memória.**

**Exemplo: trabalhando com um sistema embarcado temos que usar ponteiros para as variáveis para trabalhar com objetos pois esses podem conter vários campos e ocupar vários bytes de memória então se um objeto é utilizado em vários lugares em vez de passar o objeto na subrotina o que criaria uma cópia e literalmente ocuparia o dobro do espaço podemos usar um ponteiro referenciando o objeto que é bem menor e que diminui o custo de memória em tempo de execução.**

1. Em Linux quais as diferenças e semelhanças entre processos e threads? Indique uma vantagem e uma desvantagem de usar cada um deles.
2. Implemente, usando sua linguagem de preferência, a seguinte máquina de estados, ela representa um protocolo de conexão entre 2 equipamentos (mestre/escravo):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Estado/Evento/Proximo estado*** | ***Interface nok*** | ***Interface ok*** | ***Pacote ok recebido*** | ***Pacote nok recebido*** | ***Não recebeu resposta*** |
| 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.

*O segundo grupo de testes apresenta programas com problemas que devem ser apontados e comentados. Entenda-se como problema: bug, código mal estruturado ou confuso. Sugestões de melhoria sempre são bem vindas.*

1. Este programa exemplo deve inserir um cabeçalho de 12 caracteres em um pacote apontado por packet. Assume­se que a arquitetura é 32 bits e que o parâmetro length é o tamanho do pacote em bytes.

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;

}

1. A função a seguir funciona? É reentrante? (Porque?) O que a função faz?

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;

}

}