

Re: Cannot return values of char variable

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-11/msg00540.html

- *From:* "Pedro Pinto" <kubic62@xxxxxxxxxx>
 - *Date:* 3 Nov 2006 01:07:22 -0800
-

Ok here it goes:

The problem is, when the function menu starts, i insert the information

```
CREATE TABLE [tab] col1,col2,col3
```

that starts the function `criaRespCreate` and the `buf` variable, when exported into the program, before is ok, i print it to the screen and appears well, but when returned it comes empty.....

I start the client socked, the result of the printf's:

```
sd075@lab1215-31:~/Desktop/teste/Cliente$ ./clisql 17500
```

Sintaxe do programa cliente:

```
clisql -s <endereço_servidor> <porto_servidor>
```

```
-endereço_servidor: (opcional) IP do servidor
```

```
-porto_servidor: porto do servidor
```

Insira comando:

```
CREATE TABLE [tab] col1,col2,col3
```

```
1 - buffer =
```

```
2 - buffer =
```

```
passei o primeiro divide com aux[0] = CREATE TABLE
```

```
passei o primeiro divide com aux[1] = tab] col1,col2,col3
```

```
passei o segundo divide com aux[0] = tab
```

```
passei o segundo divide com aux[1] = col1,col2,col3
```

```
dpx do memcpy aux[0] = tab
```

```
dpx do memcpy aux[1] = col1,col2,col3
```

Re: Cannot return values of char variable

buffer =

antes do strcpy – aux[1] =

argv[0] = (null)

No test to search.

Code -----

cliente_aux.c

```
#include "cli.h"
```

```
void syntax() {
```

```
printf("Sintaxe do programa cliente:\n");
```

```
printf(" clisql -s <endereço_servidor>  
<porto_servidor>\n");
```

```
printf(" -endereço_servidor: (opcional) IP do  
servidor\n");
```

```
printf(" -porto_servidor: porto do servidor\n");
```

```
}
```

```
void sintaxe(){
```

```
printf("Sintaxe do programa cliente:\n");
```

```
printf("CREATE TABLE [table_name] coluna1,coluna2,coluna3...\n");
```

```
printf("INSERT INTO [table_name] VALUES  
(coluna1_value,coluna2_value,coluna3_value...) \n");
```

```
printf("UPDATE [table_name] SET (Coluna) WHERE (expressao)\n");
```

```
printf("SELECT [Coluna] FROM (table_name) WHERE (expressao)\n");
```

```
}
```

```
char menu(){
```

Re: Cannot return values of char variable

```
char buf[BUFFSIZE];
char bufSaida[BUFFSIZE];

char *tmp = buf;
int cod = 0;
int id = random();
int tamanhoMsg;

printf("Insira comando:\n");

// le uma linha de input
if (fgets(buf,sizeof(buf),stdin) == NULL)
perror("fgets");

char comando[10];
memset(comando,0,sizeof(comando));
if (sscanf(buf,"%9s",comando) < 1)
perror("scanf");

if (strncmp(buf,"CREATE",6) == 0){
cod = 11;

tamanhoMsg = criaRespCreate(cod,id, buf,bufSaida);
}
else if(strncmp(buf,"INSERT",6) == 0){
cod = 12;
//buffer =criaRespInsert(cod,id, buf);
}
else if(strncmp(buf,"UPDATE",6) == 0){
cod = 13;
//buffer =criaRespUpdate(cod,id, buf);
}
else if(strncmp(buf,"SELECT",6) == 0){
cod = 14;
//buffer =criaRespSelect(cod,id, buf);
}
else if(strncmp(buf,"QUIT",4) == 0){
exit(0);
}
else { printf("Comando Desconhecido\n");
}

return bufSaida;
}

int criaRespCreate(int cod, int id, char *argv, char *buffer){

int tam = 0;
```

Re: Cannot return values of char variable

Re: Cannot return values of char variable

```
int nRows = 0;
char *aux[strlen(argv)];
char str[]=" // ";
int h = 0;
//tamanho do caracter delimitador
int tcd = strlen(str);

// inserir codigo

memcpy(buffer,&cod,LENGTH);
tam += LENGTH;

printf("1 - buffer = %s\n", buffer);

// inserir caracter delimitador //
memcpy(buffer+tam,&str,strlen(str));
tam += tcd;

printf("2 - buffer = %s\n", buffer);
// inserir id da mensagem
memcpy(buffer+tam, &id, LENGTH);
tam += LENGTH;

// inserir caracter delimitador //
memcpy(buffer+tam,&str,strlen(str));
tam += tcd;

// dividir string e inserir nome tabela
divide(argv,aux,"");

printf("passei o primeiro divide com aux[0] = %s\n\n", aux[0]);
printf("passei o primeiro divide com aux[1] = %s\n\n", aux[1]);

memset(argv,0,strlen(aux));
strcpy(argv,aux[1]);
memset(aux,0,strlen(argv));
divide(argv,aux,"");

printf("passei o segundo divide com aux[0] = %s\n\n", aux[0]);
printf("passei o segundo divide com aux[1] = %s\n\n", aux[1]);
memcpy(buffer+tam,&aux[0],strlen(aux[0]));
tam += strlen(aux[0]);

printf("dpx do memcpy aux[0] = %s\n\n", aux[0]);
printf("dpx do memcpy aux[1] = %s\n\n", aux[1]);
// inserir caracter delimitador
memcpy(buffer+tam,&str,strlen(str));
tam += tcd;
printf("buffer = %s\n", buffer);
```

Re: Cannot return values of char variable

```
// inserir numero e nome colunas

memset(argv,0,strlen(aux[1]));
printf("antes do strcpy – aux[1] = %s\n\n", aux[1]);
strcpy(argv,aux[1]);
printf("argv[0] = %s\n", argv[0]);
memset(aux,0,strlen(argv));
divide(argv,aux,"");

printf("passei o terceiro divide com aux[0] = %s\n\n", aux[0]);
printf("passei o terceiro divide com aux[1] = %s\n\n", aux[1]);
printf("passei o terceiro divide com aux[2] = %s\n\n", aux[2]);

printf("str(aux) = %d\n", strlen(aux));
// inserir numero de colunas
for(h=0; aux[h]!=NULL;h++){
nRows++;
}

memcpy(buffer+tam, nRows, LENGTH);
tam += LENGTH;

// inserir caracter delimitador
memcpy(buffer+tam,&str,strlen(str));
tam += tcd;

// inserir colunas

for(h=0; aux[h]!=NULL;h++){
memcpy(buffer+tam,&aux[h],strlen(aux[h]));
tam += tcd;
// inserir caracter delimitador
memcpy(buffer,&str,strlen(str));
tam += tcd;

}
return buffer;
}
```

clisql.c-----

```
#include "cli.h"
```

Re: Cannot return values of char variable

```
/* Funcao Main */

int main (int argc, char *argv[]) {

syntax();
/* Funcao que apresenta sintaxe do programa */

int sock;

int broadcast = 1;

struct sockaddr_in server;

struct sockaddr_in cli;

int porto_servidor;
char buffer[BUFSIZE];
char *tmp = buffer;
tmp = buffer;

char input[1024];

/* Limpa a estrutura */

bzero((char *)&server, sizeof(server));

bzero((char *)&cli, sizeof(cli));

/* Limpar o buffer */

memset(buffer, 0, BUFSIZE);

input[0] = menu(buffer);
```

Re: Cannot return values of char variable

```
/* Criacao da socket */  
  
if ((sock = socket(AF_INET,SOCK_DGRAM,0)) <0) {  
  
perror("socket");  
  
exit(1);  
  
}  
  
else {  
  
printf("Cliente: socket criada \n");  
  
}  
  
  
  
/* Verificar ip */  
if(strcmp(argv[1], "-s") == 0) {  
  
printf("entrei n o verifica ip, estamos a funcionar com endereco ip  
inserido\n");  
  
inet_aton(argv[2], &server.sin_addr);  
porto_servidor = atoi(argv[3]);  
  
}  
  
else { //enviar em broadcast  
  
printf("Não foi fornecido o endereço IP, a enviar em broadcast\n");  
  
porto_servidor = atoi(argv[1]);  
server.sin_addr.s_addr = htonl(INADDR_BROADCAST);  
  
  
  
/* Activar broadcast na socket */  
  
if (setsockopt(sock, SOL_SOCKET, SO_BROADCAST, (char*)&broadcast,  
sizeof (int)) <0) {  
  
perror("setsockopt");  
  
exit(1);  
  
}  
  
}
```

Re: Cannot return values of char variable

```
/* verifica se a porta esta entre os valores pretendidos,  
  
* visto o número de grupo ser o 75, port vai variar entre  
  
* 17500 e 17599 */  
void checkPort(porto_servidor) {  
  
if(porto_servidor < 17500 || porto_servidor > 17599) {  
printf("Cliente: O porto tem de ser entre 17500 e 17599!\n");  
exit(-1);  
}  
}
```

```
/* Definir a familia de protocolo e porto */  
server.sin_family = AF_INET;  
  
server.sin_port = htons(porto_servidor);
```

```
/* Inicio da comunicacao com o servidor */
```

```
/* Enviar a mensagem para o servidor*/
```

Re: Cannot return values of char variable

```
if(sendto(sock, tmp, strlen(buffer), 0, (struct sockaddr *)&server,  
sizeof(server)) <0) {
```

```
    perror("sendto\n");
```

```
    exit(1);
```

```
}
```

```
else {
```

```
    printf("Cliente: Mensagem enviada com sucesso\n");
```

```
}
```

```
while(1){  
    printf("entrei no while\n");
```

```
    /* timeout de 10 segundos */
```

```
    struct timeval timeout;
```

```
    timeout.tv_sec = 10; //segundos
```

```
    timeout.tv_usec = 0; //microsegundos
```

```
    fd_set readfds;
```

```
    int sel;
```

```
    // char buf1[64];
```

```
    /* limpar a fd_set */
```

```
    FD_ZERO(&readfds);
```

```
    /* colocar o file descriptor no fd_set */
```

```
    FD_SET(sock, &readfds);
```

```
    printf("vou ver agora o select\n");
```

```
    if((sel = select(sock+1, &readfds, NULL, NULL, &timeout)) <0)
```

Re: Cannot return values of char variable

Re: Cannot return values of char variable

```
perror("select\n");

else if (sel == 0){

printf("Ocorreu um timeout! Nao foi recebida nenhuma mensagem em
10s.\n");

exit(0);
}

memset(buffer,0,BUFFSIZE);

printf("vou agora ler a resposta do servidor\n");

/* ler a resposta do servidor */

if(recvfrom(sock, buffer, BUFFSIZE, 0, NULL, NULL) < 0) {
printf("entrei no recv por ser < 0\n");

perror("recvfrom\n");

exit(1);

}

else {

printf("Cliente: Mensagem recebida com sucesso\n");

printf("buffer = %s\n", buffer);
exit(0);
}

}

/* fecho da socket */

close(sock);

}
```

Re: Cannot return values of char variable

-----cli.h-----

```
#ifndef CLI_H
#define CLI_H
```

```
#include "projecto.h"
```

```
/* Definicao das funcoes definidas em cliente_aux.c */
```

```
void syntax();
```

```
char menu();
```

```
int criaRespCreate(int cod, int id, char *argv, char *buffer);
```

```
#endif
```

-----projecto.h-----

```
#ifndef PROJECTO_H
```

```
#define PROJECTO_H
```

```
#define BUFFSIZE 6804
```

```
#define LENGTH 4
```

```
/* Bibliotecas necessarias */
```

```
#include <sys/socket.h>
```

```
#include <sys/types.h>
```

Re: Cannot return values of char variable

Re: Cannot return values of char variable

```
#include <stdio.h>

#include <errno.h>

#include <sys/time.h>

#include <stdlib.h>
#include <arpa/inet.h>

#include <netinet/in.h>

#include <netdb.h>
#include <string.h>

#endif
```

.