

Forum:[Hack Forums](#)

Thread: [Hacker Test " C Linked List With Dynamic variables " Just Plain Ugly](#)

g4143 - June 18, 2009, 5:44 pm

I created this very short linked list program that supports dynamic variables to some extent. It will prompt the user to choose:

(0)exit (1)view (2)set (3)free it

where 0 = exit or terminate the program.

and 1 = view the linked list and its variables

and 2 = set the next linked list node

and 3 = free it, delete all the links and pointers (this doesn't work yet)

Now when you choose option 2 form above

you will again be presented with a list of options

enter type (0) INT (1) FLOAT (2) CHAR (3) STRING

where 0 = any valid integer such as - 12234

and 1 = any valid float such as - 89.987

and 2 = any valid character such as - G

and 3 = any valid string less than 100 characters such as - this is my string

Now when you execute this program you have to be very careful because I have no error checking in it right now and it will crash if you enter a character for an integer....

Note if this program doesn't scare you away from C/C++ then nothing will

Now the next few steps will walk you through using the program

First copy and compile the program.

Second run it.

At the main prompt " (0)exit (1)view (2)set (3)free it "?

type 2 and press enter

at the secondary prompt "enter type (0) INT (1) FLOAT (2) CHAR (3) STRING"?

type 0 and press enter

at the thirdly prompt "enter a integer->"

type 3456 and press enter

Now you just created a new linked list node with a variable integer

you should be back a the main prompt " (0)exit (1)view (2)set (3)free it "?

type 2 and press enter

at the secondary prompt "enter type (0) INT (1) FLOAT (2) CHAR (3) STRING"?

type 3 and press enter

at the thirdly prompt "enter a string of text->"

type " this is my string of text- and press enter

Now you just created a new linked list node with a variable cstring

and should be at the main prompt "œ(0)exit (1)view (2)set (3)free itâ€?"

now choose 1 and press enter

you should see your integer and string linked list nodes...

Note the first node in this linked list is static and its a integer node with the value 999 that's why your seeing an entry " integer value-> 999...

now enter 0 exit

If you follow the above rules for creating links in this program you should have no problems...

If your looking for a challenge " write the freeit() function for this code..G4143

Code #include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <string.h>

enum VARTYPE {INT, FLOAT, CHAR, STRING};

#define VTYPE unsigned char

struct slink

```
{
    VTYPE vtype;
    void *vptr;
    struct slink *slptr;
};
```

void showchoice(struct slink *sl)

```
{
    switch (sl->vtype)
    {
        case INT:
        {
            fprintf(stdout, "integer value->\t\t%d\n", *(unsigned int*)sl->vptr);
            break;
        }
        case FLOAT:
        {
            fprintf(stdout, "float value->\t\t%f\n", *(float*)sl->vptr);
            break;
        }
        case CHAR:
```

```

    {
        fprintf(stdout, "character value->\t%c\n", *(unsigned char*)sl->vptr);
        break;
    }
    case STRING:
    {
        fprintf(stdout, "cstring value->\t\t%s", (char*)sl->vptr);
        break;
    }
}
}

```

```

void showit(struct slink *sl)
{
    if (sl->slptr)
    {
        showit(sl->slptr);
        showchoice(sl);
    }
    else
    {
        showchoice(sl);
    }
}

```

```

void setit(struct slink *sl, VTYPE vt)
{
    if (sl->slptr)
    {
        setit(sl->slptr, vt);
    }
    else
    {
        sl->slptr = (struct slink*)malloc(sizeof(struct slink));
        sl->slptr->vtype = vt;
        sl->slptr->slptr = (struct slink*)NULL;
        switch (vt)
        {
            case CHAR:
            {
                unsigned char x;
                fputs("enter a character->", stdout);
                while ((x = getchar ()) != '\n');
                fscanf(stdin, "%c", &x);
                sl->slptr->vptr = (void*)malloc(sizeof(unsigned char));
                *(unsigned char*)sl->slptr->vptr = x;
                break;
            }
            case STRING:
            {
                char ch[100];
                int size = 0;
                int i = 0;

```

```

        fputs("enter a string of text->", stdout);
        while ((ch[0] = getchar ()) != '\n');
        fgets(ch, 99, stdin);
        size = strlen(ch);
        sl->slptr->vptr = (void*)malloc(size + 1);
        for (i = 0; i < size; ++i)
            ((char*)(sl->slptr->vptr))[i] = ch[i];
        ((char*)(sl->slptr->vptr))[size] = '\0';
        break;
    }
    case INT:
    {
        int x;
        fputs("enter a integer->", stdout);
        fscanf(stdin, "%d", &x);
        sl->slptr->vptr = (void*)malloc(sizeof(int));
        *(unsigned int*)sl->slptr->vptr = x;
        break;
    }
    case FLOAT:
    {
        float x;
        fputs("enter a float->", stdout);
        fscanf(stdin, "%f", &x);
        sl->slptr->vptr = (void*)malloc(sizeof(float));
        *(float*)sl->slptr->vptr = x;
        break;
    }
}
}
}
}

```

```

int main(int argc, char**argv)
{
    int choice, val;

    struct slink mylink;

    mylink.vtype = INT;
    mylink.vptr = (void*)malloc(sizeof(int));
    *(unsigned int*)mylink.vptr = 999;
    mylink.slptr = (struct slink*)NULL;

    while (true)
    {
        fputs("(0)exit (1)view (2)set (3)freeit->", stdout);
        fscanf(stdin, "%d", &choice);

        if (choice <= 0) break;

        if (choice == 1) showit(&mylink);
    }
}

```

```

    if (choice == 2)
    {
        fputs("enter type (0) INT (1) FLOAT (2) CHAR (3) STRING->", stdout);
        fscanf(stdin, "%d", &val);
        setit(&mylink, val);
    }

    if (choice == 3)
    {
        //freeit(&mylink);
    }
}
exit(EXIT_SUCCESS);
}

```

an balker - December 28, 2019, 8:24 am

(Jun 18, 2009 05:44 PM)g4143 Wrote: [\[CITE\]](#) I created this very short linked list program that supports dynamic variables to some extent. It will prompt the user to choose:

(0)exit (1)view (2)set (3)free it

where 0 = exit or terminate the program.
 and 1 = view the linked list and its variables
 and 2 = set the next linked list node
 and 3 = free it, delete all the links and pointers (this doesn't work yet)

Now when you choose option 2 form above
 you will again be presented with a list of options

enter type (0) INT (1) FLOAT (2) CHAR (3) STRING

where 0 = any valid integer such as - 12234
 and 1 = any valid float such as - 89.987
 and 2 = any valid character such as - G
 and 3 = any valid string less than 100 characters such as - this is my string

Now when you execute this program you have to be very careful because I have noerror checking in it right now and it will crash if you enter a character for an integer....

Note if this program doesn't scare you away from C/C++ then nothing will

Now the next few steps will walk you through using the program

First copy and compile the program.
 Second run it.

At the main prompt "œ(0)exit (1)view (2)set (3)free itœ?
 type 2 and press enter

at the secondary prompt "œenter type (0) INT (1) FLOAT (2) CHAR (3) STRINGœ?
 type 0 and press enter

at the thirdly prompt "enter a integer->"

type 3456 and press enter

Now you just created a new linked list node with a variable integer

you should be back at the main prompt `œ(0)exit (1)view (2)set (3)free itœ?`
type 2 and press enter

at the secondary prompt `œenter type (0) INT (1) FLOAT (2) CHAR (3) STRINGœ?`
type 3 and press enter

at the thirdly prompt "enter a string of text->"
type `œ" this is my string of text-` and press enter

Now you just created a new linked list node with a variable cstring

and should be at the main prompt `œ(0)exit (1)view (2)set (3)free itœ?`

now choose 1 and press enter

you should see your integer and string linked list nodes...

Note the first node in this linked list is static and its a integer node with the value 999 that's why your seeing an entry `œ" integer value-> 999...`

now enter 0 exit

If you follow the above rules for creating links in this program you should have no problems...

If your looking for a challenge `œ" write the freeit() function for this code..G4143`

```
Code #include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
```

```
enum VARTYPE {INT, FLOAT, CHAR, STRING};
```

```
#define VTYPE unsigned char
```

```
struct slink
{
    VTYPE vtype;
    void *vptr;
    struct slink *slptr;
};
```

```
void showchoice(struct slink *sl)
{
    switch (sl->vtype)
    {
        case INT:
        {
            printf(stdout, "integer value->\t\t%d\n", *(unsigned int*)sl->vptr);
```

```

break;
}
case FLOAT:
{
fprintf(stdout, "float value->\t\t%f\n", *(float*)sl->vptr);
break;
}
case CHAR:
{
fprintf(stdout, "character value->\t\t%c\n", *(unsigned char*)sl->vptr);
break;
}
case STRING:
{
fprintf(stdout, "cstring value->\t\t%s", (char*)sl->vptr);
break;
}
}
}

```

```

void showit(struct slink *sl)
{
if (sl->slptr)
{
showit(sl->slptr);
showchoice(sl);
}
else
{
showchoice(sl);
}
}

```

```

void setit(struct slink *sl, VTYPE vt)
{
if (sl->slptr)
{
setit(sl->slptr, vt);
}
else
{
sl->slptr = (struct slink*)malloc(sizeof(struct slink));
sl->slptr->vtype = vt;
sl->slptr->slptr = (struct slink*)NULL;
switch (vt)
{
case CHAR:
{
unsigned char x;
fputs("enter a character->", stdout);
while ((x = getchar ()) != '\n');
fscanf(stdin, "%c", &x);
sl->slptr->vptr = (void*)malloc(sizeof(unsigned char));

```

```

*(unsigned char*)sl->slptr->vptr = x;
break;
}
case STRING:
{
char ch[100];
int size = 0;
int i = 0;
fputs("enter a string of text->", stdout);
while ((ch[0] = getchar ()) != '\n');
fgets(ch, 99, stdin);
size = strlen(ch);
sl->slptr->vptr = (void*)malloc(size + 1);
for (i = 0; i < size; ++i)
((char*)(sl->slptr->vptr))[i] = ch[i];
((char*)(sl->slptr->vptr))[size] = '\0';
break;
}
case INT:
{
int x;
fputs("enter a integer->", stdout);
fscanf(stdin, "%d", &x);
sl->slptr->vptr = (void*)malloc(sizeof(int));
*(unsigned int*)sl->slptr->vptr = x;
break;
}
case FLOAT:
{
float x;
fputs("enter a float->", stdout);
fscanf(stdin, "%f", &x);
sl->slptr->vptr = (void*)malloc(sizeof(float));
*(float*)sl->slptr->vptr = x;
break;
}

}
}
}

```

```

int main(int argc, char**argv)
{
int choice, val;

```

```

struct slink mylink;

```

```

mylink.vtype = INT;
mylink.vptr = (void*)malloc(sizeof(int));
*(unsigned int*)mylink.vptr = 999;
mylink.slptr = (struct slink*)NULL;

```

```

while (true)

```

```
{
fputs("(0)exit (1)view (2)set (3)freeit->", stdout);
fscanf(stdin, "%d", &choice);

if (choice <= 0) break;

if (choice == 1) showit(&mylink);

if (choice == 2)
{
fputs("enter type (0) INT (1) FLOAT (2) CHAR (3) STRING->",stdout);
fscanf(stdin, "%d", &val);
setit(&mylink, val);
}

if (choice == 3)
{
//freeit(&mylink);
}
}
exit(EXIT_SUCCESS);
}
```

good feed your code has been mentioned in my feedlog, once again great position