2D character array initialization in C

I am trying to build a list of strings that I need to pass to a function expecting `char **`

How do I build this array? I want to pass in two options, each with less than 100 characters.

```
char **options[2][100];

options[0][0] = 'test1';
options[1][0] = 'test2';
```

This does not compile. What am I doing wrong exactly? How do I create a 2D character array in C?

c | arrays | multidimensional-array | char

edited Sep 30 '13 at 20:59

Eric Leschinski
16.7k 9 90 98

asked Jul 6 '11 at 16:20

Derek
2,895 5 46 135

2 `char **options[2][100]` is an array [2][100] of pointers to pointers of chars. – sidyll Jul 6 '11 at 16:24

add a comment

4 Answers

C strings are enclosed in double quotes:

```
const char *options[2][100];

options[0][0] = "test1";
options[1][0] = "test2";
```

Re-reading your question and comments though I'm guessing that what you *really* want to do is this:

```
const char *options[2] = { "test1", "test2" };
```

edited Jul 6 '11 at 16:53

answered Jul 6 '11 at 16:22

Paul R

110k 11 141 253

This is the same as larsmans answer. However, it seems that my function is still complaining that i am not passing char** – Derek Jul 6 '11 at 16:32

1 Edit your question so that it includes the function to which you are trying to pass this array. – Paul R Jul 6 '11 at 16:34

If I wish to allocate memory dynamically for this above 2D array. How can I do this..? @PaulR – ranaarjun Feb 12 at 14:15

char (*str)[5] = malloc(sizeof(*str)*2); str[0][1] = "abcd"; I do this in this way. But compiler throws an error.!!!! @PaulR – ranaarjun Feb 12 at 14:20

@ranaarjun: you should post a new question describing your problem in detail – Paul R Feb 12 at 14:27

show **1** more comment

```
char **options[2][100];
```

declares a size-2 array of size-100 arrays of pointers to pointers to `char`. You'll want to remove one `*`. You'll also want to put your string literals in double quotes.

answered Jul 6 '11 at 16:23

larsmans
176k 18 275 444

Do I need to dereference this or something when passing to a function looking for char**? Right now it is complaining that char*[2][100] is not a valid argument – Derek Jul 6 '11 at 16:27

@Derek: you can't pass `options` to such a function, but you can pass `options[i]` for `i` in `{0,1}` . – larsmans Jul 6 '11 at 16:31

add a comment

How to create an array size 5 containing pointers to characters:

```
char *array_of_pointers[ 5 ];        //array size 5 containing pointers to char
char m = 'm';                        //character value holding the value 'm'
array_of_pointers[0] = &m;           //assign m ptr into the array position 0.
printf("%c", *array_of_pointers[0]); //get the value of the pointer to m
```

How to create a pointer to an array of characters:

```
char (*pointer_to_array)[ 5 ];        //A pointer to an array containing 5 chars
char m = 'm';                         //character value holding the value 'm'
*pointer_to_array[0] = m;             //dereference array and put m in position 0
printf("%c", (*pointer_to_array)[0]); //dereference array and get position 0
```

How to create an 2D array containing pointers to characters:

```
char *array_of_pointers[5][2];
//An array size 5 containing arrays size 2 containing pointers to char

char m = 'm';
//character value holding the value 'm'

array_of_pointers[4][1] = &m;
//Get position 4 of array, then get position 1, then put m ptr in there.

printf("%c", *array_of_pointers[4][1]);
//Get position 4 of array, then get position 1 and dereference it.
```

How to create a pointer to an 2D array of characters:

```
char (*pointer_to_array)[5][2];
//A pointer to an array size 5 each containing arrays size 2 which hold chars

char m = 'm';
//character value holding the value 'm'

(*pointer_to_array)[4][1] = m;
//dereference array, Get position 4, get position 1, put m there.

printf("%c", (*pointer_to_array)[4][1]);
//dereference array, Get position 4, get position 1
```

To help you out with understanding how humans should read complex C/C++ declarations read this:
http://www.programmerinterview.com/index.php/c-cplusplus/c-declarations/

edited Oct 1 '13 at 16:56

answered Sep 30 '13 at 20:58

Eric Leschinski
16.7k 9 90 98

add a comment

I think what you originally meant to do was make an array only of characters, not of pointers:

```
char options[2][100];
```

```
options[0][0]='t';
options[0][1]='e';
options[0][2]='s';
options[0][3]='t';
options[0][4]='1';
options[0][5]='\0';  /* NUL termination of C string */

/* A standard C library function which copies strings. */
strcpy(options[1], "test2");
```

The code above shows two distinct methods of setting the character values in memory you have set aside to contain characters.

answered Jul 6 '11 at 16:28

Heath Hunnicutt

10k 17 45

add a comment

Not the answer you're looking for? Browse other questions tagged c | arrays | multidimensional-array | char **or ask your own question.**