

Re: Reading a string of unknown size

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-11/msg03663.html

- *From:* "Sundar" <sunder.svit@xxxxxxxxxx>
 - *Date:* 27 Nov 2006 20:46:32 -0800
-

santosh wrote:

Santosh wrote:

I have to read characters from stdin and save them in a string.
The
problem is that I don't know how much characters will be
read.

First include necessary headers: stdio.h, stdlib.h

```
int main()
```

Better yet, replace above with `int main(void)`

```
{  
char *str = NULL, ch ;  
int i = 0 ;  
str = (char*) malloc (2*sizeof(char)) ;
```

Don't cast return value of `malloc()` in C. It can hide the non-inclusion of its prototype, (by way of failure to include `stdlib.h`), and, on some implementations, can result in nasty crashes during runtime.

Since `sizeof(char)` is by definition 1, you can omit that and instead do `'2 * sizeof *str'`. This has the advantage of becoming automatically updated when you later on happen to change the type of `*str`.

```
*str = '\0' ;
```

Re: Reading a string of unknown size

And now you're possibly writing to a random area of memory, since you failed to check the return value of `malloc()` above for failure.

```
while( (ch=getchar()) != '\n' )
```

Check for EOF not newline. Moreover `getchar()` returns an `int` value which you're storing in a `char` variable, probably getting a spurious garbage return value when end-of-file is encountered.

```
{
*(str+i) = ch ;
```

You've overwritten your earlier nul character.

```
i++ ;
str = (char*) realloc(str, (2*sizeof(char)) + i ) ;
```

Again, `_don't_` cast the return value of `XXalloc()` functions in C, and check the call for failure before proceeding further. Also change `sizeof(char)` to `sizeof *str`.

Anyway, your allocation strategy is very inefficient. Your calling `realloc()` once every iteration of the loop. This could result in fragmentation of the C library's memory pool. Why not allocate in terms of fixed sized or dynamically growing blocks, say 128 bytes or so to start with?

```
}
*(str+i) = '\0' ;

printf("\n\n %s ", str) ;
```

Unless you terminate the output with a newline character, it's not guaranteed to show up on screen, or wherever `stdout` happens to be directed to.

```
getch() ;
```

Non-standard, unportable and unnecessary function. Just get rid of it.

Re: Reading a string of unknown size

Can u please further explain the following statement....

"Don't cast return value of malloc() in C. It can hide the non-inclusion of it's prototype, (by way of failure to include stdlib.h), and, on some implementations, can result in nasty crashes during runtime."

I am unable to understand the intricacies of the above statement.

.