NAME

getsubopt – parse suboptions from a string

SYNOPSIS

#include <stdlib.h>

int getsubopt(char **optionp*, **const char * const** **tokens*, **char** **valuep*);**

MT-LEVEL

MT-Safe

DESCRIPTION

getsubopt( ) parses suboptions in a flag argument that was initially parsed by **getopt**(3C). These suboptions are separated by commas and may consist of either a single token or a token-value pair separated by an equal sign. Since commas delimit suboptions in the option string, they are not allowed to be part of the suboption or the value of a suboption. A command that uses this syntax is **mount**(1M), which allows the user to specify mount parameters with the **-o** option as follows:

mount –o rw,hard,bg,wsize=1024 speed:/usr /usr

In this example there are four suboptions: **rw**, **hard**, **bg**, and **wsize**, the last of which has an associated value of 1024.

getsubopt( ) takes the address of a pointer to the option string, a vector of possible tokens, and the address of a value string pointer. It returns the index of the token that matched the suboption in the input string or −1 if there was no match. If the option string at *optionp* contains only one subobtion, **getsubopt( )** updates *optionp* to point to the null character at the end of the string; otherwise it isolates the suboption by replacing the comma separator with a null character, and updates *optionp* to point to the start of the next suboption. If the suboption has an associated value, **getsubopt( )** updates *valuep* to point to the value's first character. Otherwise it sets *valuep* to **NULL**.

The token vector is organized as a series of pointers to null strings. The end of the token vector is identified by a null pointer.

When **getsubopt( )** returns, if *valuep* is not **NULL**, then the suboption processed included a value. The calling program may use this information to determine if the presence or lack of a value for this subobtion is an error.

Additionally, when **getsubopt( )** fails to match the suboption with the tokens in the *tokens* array, the calling program should decide if this is an error, or if the unrecognized option should be passed to another program.

RETURN VALUES

getsubopt( ) returns −1 when the token it is scanning is not in the token vector. The variable addressed by *valuep* contains a pointer to the first character of the token that was not recognized rather than a pointer to a value for that token.

The variable addressed by *optionp* points to the next option to be parsed, or a null character if there are no more options.

EXAMPLE

The following code fragment shows how to process options to the **mount**(1M) command using **getsubopt( )**.

```
#include <stdlib.h>

char *myopts[] = {
#define READONLY      0
                      "ro",
#define READWRITE     1
                      "rw",
#define WRITESIZE     2
                      "wsize",
#define READSIZE      3
                      "rsize",
                      NULL};

main(argc, argv)
        int  argc;
        char **argv;
{
        int sc, c, errflag;
        char *options, *value;
        extern char *optarg;
        extern int optind;
        .
        .
        .
        while((c = getopt(argc, argv, "abf:o:")) != -1) {
                switch (c) {
                case 'a': /* process a option */
                        break;
                case 'b': /* process b option */
                        break;
                case 'f':
                        ofile = optarg;
                        break;
                case '?':
                        errflag++;
                        break;
                case 'o':
                        options = optarg;
                        while (*options != '\0') {
                                switch(getsubopt(&options,myopts,&value) {
                                case READONLY : /* process ro option */
                                        break;
                                case READWRITE : /* process rw option */
                                        break;
```

```
                    case WRITESIZE : /* process wsize option */
                        if (value == NULL) {
                            error_no_arg( );
                            errflag++;
                        } else
                            write_size = atoi(value);
                        break;
                    case READSIZE : /* process rsize option */
                        if (value == NULL) {
                            error_no_arg( );
                            errflag++;
                        } else
                            read_size = atoi(value);
                        break;
                    default :
                        /* process unknown token */
                        error_bad_token(value);
                        errflag++;
                        break;
                    }
                }
                break;
            }
        }
        if (errflag) {
            /* print usage instructions etc. */
        }
        for (; optind<argc; optind++) {
            /* process remaining arguments */
        }
        .
        .
        .
}
```

SEE ALSO

mount(1M), **getopt**(3C)

NOTES

During parsing, commas in the option input string are changed to null characters. White space in tokens or token-value pairs must be protected from the shell by quotes.