530128-17-4E AID:83578 | 03/18/2019

**Program Plan:**

Removing the compile-time limit on the size(MAXARGC) of the **argv** variable and use dynamic memory allocation.

**Program:**

**#include** "apue.h"

**#define** WHITE " \t\n" /\* white space for tokenizing arguments \*/

/\*

\* buf[] contains white-space-separated arguments. We convert it to an

\* argv-style array of pointers, and call the user’s function (optfunc)

\* to process the array. We return -1 if there’s a problem parsing buf,

\* else we return whatever optfunc() returns. Note that user’s buf[]

\* array is modified (nulls placed after each token).

\*/

**int** **buf\_args**(**char** \*buf, **int** (\*optfunc)(**int**, **char** \*\*))

{

**char** \*ptr, \*\*argv;

**int** argc = 0, length;

**int** i, j;

length = **strlen**(buf);

**if** (**strtok**(buf, WHITE) == NULL) /\* an argv[0] is required \*/

**return** (-1);

/\* to determine number of tokenized strings \*/

**while** (**strtok**(NULL, WHITE) != NULL)

++argc;

/\* dynamic memory allocation \*/

argv = (**char**\*\*) **malloc**(argc \* **sizeof**(**char**\*));

argv[j = 0] = buf;

**for** (i = 0; i < length; i++)

**if** (buf[i] == '\0' && j < argc)

argv[++j] = &buf[++i];

/\*

\* Since argv[] pointers point into the user’s buf[],

\* user’s function can just copy the pointers, even

\* though argv[] array will disappear on return.

\*/

**return** ((\*optfunc)(argc, argv));

}