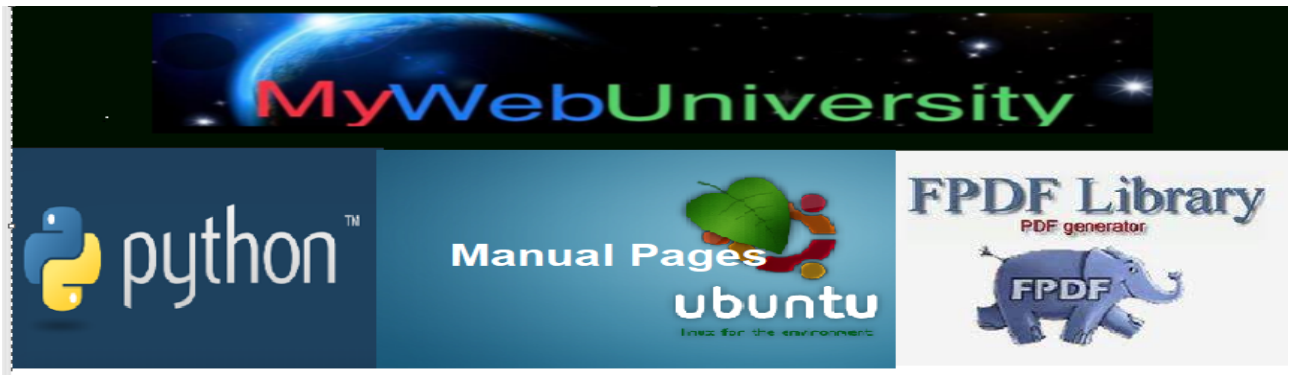

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'getopt_long.3'

$ man getopt_long.3

getopt_long(3) Library Functions Manual getopt_long(3)

NAME

getopt_long - parse command-line options

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#define _GNU_SOURCE
#include <getopt.h>
int getopt_long(int argc, char *argv[],
        const char *optstring,
        const struct option *longopts, int *longindex);
```

DESCRIPTION

The getopt_long() function works like getopt(3) except that it also accepts long options, started with two dashes. (If the program accepts only long options, then optstring should be specified as an empty string (""), not NULL.) Long option names may be abbreviated if the abbreviation is unique or is an exact match for some defined option. A long option may take a parameter, of the form --arg=param or --arg param.

longopts is a pointer to the first element of an array of struct option declared in <getopt.h> as

```
struct option {
    const char *name;
    int         has_arg;
    int        *flag;
```

```
        int         val;
    };
```

The meanings of the different fields are:

name is the name of the long option.

has_arg

is: no_argument (or 0) if the option does not take an argument; required_argument (or 1) if the option requires an argument; or optional_argument (or 2) if the option takes an optional argument.

flag specifies how results are returned for a long option. If flag is NULL, then getopt_long() returns val. (For example, the calling program may set val to the equivalent short option character.) Otherwise, getopt_long() returns 0, and flag points to a variable which is set to val if the option is found, but left unchanged if the option is not found.

val is the value to return, or to load into the variable pointed to by flag.

The last element of the array has to be filled with zeros.

If longindex is not NULL, it points to a variable which is set to the index of the long op? tion relative to longopts.

Its analogue to getopt(3)'s optopt is ?argv[(optind - 1)]?.

RETURN VALUE

See getopt(3).

getopt_long() also returns the option character when a short option is recognized. For a long option, it returns val if flag is NULL, and 0 otherwise. Error and -1 returns are the same as for getopt(3), plus '?' for an ambiguous match or an extraneous parameter.

ENVIRONMENT

See getopt(3).

ATTRIBUTES

For an explanation of the terms used in this section, see attributes(7).

```
????????????????????????????????????????????????????????????????????????????????????????????
? Interface     ? Attribute     ? Value                                   ?
????????????????????????????????????????????????????????????????????????????????????????????
? getopt_long() ? Thread safety ? MT-Unsafe race:getopt env               ?
????????????????????????????????????????????????????????????????????????????????????????????
```

STANDARDS

GNU.

EXAMPLES

The following example program illustrates the use of getopt_long() with most of its fea?

tures.

```
#include <getopt.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int
main(int argc, char *argv[])
{
    int c;
    int digit_optind = 0;
    while (1) {
        int this_option_optind = optind ? optind : 1;
        int option_index = 0;
        static struct option long_options[] = {
            {"add",     required_argument, 0,  0 },
            {"append",  no_argument,       0,  0 },
            {"delete",  required_argument, 0,  0 },
            {"verbose", no_argument,       0,  0 },
            {"create",  required_argument, 0, 'c'},
            {"file",    required_argument, 0,  0 },
            {0,         0,                 0,  0 }
        };
        c = getopt_long(argc, argv, "abc:d:012",
                    long_options, &option_index);
        if (c == -1)
            break;
        switch (c) {
        case 0:
            printf("option %s", long_options[option_index].name);
            if (optarg)
```

```
            printf(" with arg %s", optarg);
        printf("\n");
        break;
    case '0':
    case '1':
    case '2':
        if (digit_optind != 0 && digit_optind != this_option_optind)
          printf("digits occur in two different argv-elements.\n");
        digit_optind = this_option_optind;
        printf("option %c\n", c);
        break;
    case 'a':
        printf("option a\n");
        break;
    case 'b':
        printf("option b\n");
        break;
    case 'c':
        printf("option c with value '%s'\n", optarg);
        break;
    case 'd':
        printf("option d with value '%s'\n", optarg);
        break;
    case '?':
        break;
    default:
        printf("?? getopt returned character code 0%o ??\n", c);
    }
}
if (optind < argc) {
    printf("non-option ARGV-elements: ");
    while (optind < argc)
        printf("%s ", argv[optind++]);
```

```
            printf("\n");
        }
        exit(EXIT_SUCCESS);
    }
```

SEE ALSO

getopt(1), getopt(3), getopt_long_only(3), getsubopt(3)