



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 'getprotobyname_r.3'

\$ man getprotobyname_r.3

getprotoent_r(3) Library Functions Manual getprotoent_r(3)

NAME

getprotoent_r, getprotobyname_r, getprotobyname_r - get protocol entry (reentrant)

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <netdb.h>
```

```
int getprotoent_r(size_t size;
```

```
    struct protoent *restrict result_buf,
```

```
    char buf[restrict size], size_t size,
```

```
    struct protoent **restrict result);
```

```
int getprotobyname_r(size_t size;
```

```
    const char *restrict name,
```

```
    struct protoent *restrict result_buf,
```

```
    char buf[restrict size], size_t size,
```

```
    struct protoent **restrict result);
```

```
int getprotobyname_r(size_t size;
```

```
    int proto,
```

```
    struct protoent *restrict result_buf,
```

```
    char buf[restrict size], size_t size,
```

```
    struct protoent **restrict result);
```

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

```
getprotoent_r(), getprotobyname_r(), getprotobyname_r():
```


[illegible]

```

int size, erange_cnt, s;

struct protoent result_buf;

struct protoent *result;

char buf[MAX_BUF];

if (argc < 2) {
    printf("Usage: %s proto-name [size]\n", argv[0]);
    exit(EXIT_FAILURE);
}

size = 1024;

if (argc > 2)
    size = atoi(argv[2]);

if (size > MAX_BUF) {
    printf("Exceeded buffer limit (%d)\n", MAX_BUF);
    exit(EXIT_FAILURE);
}

erange_cnt = 0;

do {
    s = getprotobyname_r(argv[1], &result_buf,
                        buf, size, &result);

    if (s == ERANGE) {
        if (erange_cnt == 0)
            printf("ERANGE! Retrying with larger buffer\n");

        erange_cnt++;

        /* Increment a byte at a time so we can see exactly
           what size buffer was required. */
        size++;

        if (size > MAX_BUF) {
            printf("Exceeded buffer limit (%d)\n", MAX_BUF);
            exit(EXIT_FAILURE);
        }
    }
} while (s == ERANGE);

printf("getprotobyname_r() returned: %s (size=%d)\n",

```

```

    (s == 0) ? "0 (success)" : (s == ENOENT) ? "ENOENT" :
    strerror(s), size);
if (s != 0 || result == NULL) {
    printf("Call failed/record not found\n");
    exit(EXIT_FAILURE);
}
printf("p_name=%s; p_proto=%d; aliases=",
    result_buf.p_name, result_buf.p_proto);
for (char **p = result_buf.p_aliases; *p != NULL; p++)
    printf("%s ", *p);
printf("\n");
exit(EXIT_SUCCESS);
}

```

SEE ALSO

[getprotoent\(3\)](#), [protocols\(5\)](#)

Linux man-pages 6.17

2025-09-21

[getprotoent_r\(3\)](#)