



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 'getservent_r.3'

\$ man getservent_r.3

getservent_r(3) Library Functions Manual getservent_r(3)

NAME

getservent_r, getservbyname_r, getservbyport_r - get service entry (reentrant)

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

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

```
int getservent_r(size_t size;
```

```
    struct servent *restrict result_buf,
```

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

```
    struct servent **restrict result);
```

```
int getservbyname_r(size_t size;
```

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

```
    const char *restrict proto,
```

```
    struct servent *restrict result_buf,
```

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

```
    struct servent **restrict result);
```

```
int getservbyport_r(size_t size;
```

```
    int port,
```

```
    const char *restrict proto,
```

```
    struct servent *restrict result_buf,
```

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

```
    struct servent **restrict result);
```



```

int size, erange_cnt, port, s;

struct servent result_buf;

struct servent *result;

char buf[MAX_BUF];

char *protop;

if (argc < 3) {
    printf("Usage: %s port-num proto-name [size]\n", argv[0]);
    exit(EXIT_FAILURE);
}

port = htons(atoi(argv[1]));

protop = (strcmp(argv[2], "null") == 0 ||
          strcmp(argv[2], "NULL") == 0) ? NULL : argv[2];

size = 1024;

if (argc > 3)
    size = atoi(argv[3]);

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

erange_cnt = 0;

do {
    s = getservbyport_r(port, protop, &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);

```

```

    }
}
} while (s == ERANGE);

printf("getservbyport_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("s_name=%s; s_proto=%s; s_port=%d; aliases=",
       result_buf.s_name, result_buf.s_proto,
       ntohs(result_buf.s_port));

for (char **p = result_buf.s_aliases; *p != NULL; p++)
    printf("%s ", *p);

printf("\n");

exit(EXIT_SUCCESS);
}

```

SEE ALSO

getservent(3), services(5)

Linux man-pages 6.17

2025-09-21

getservent_r(3)