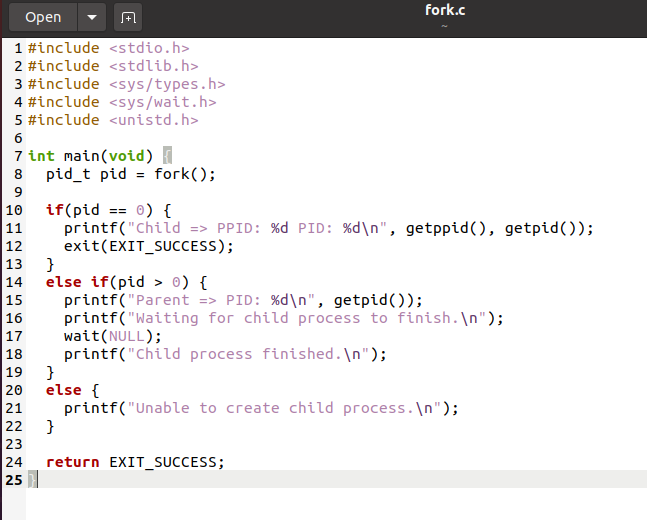
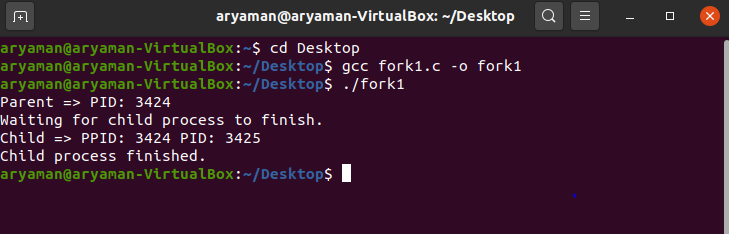
ARYAMAN MISHRA

19BCE1027

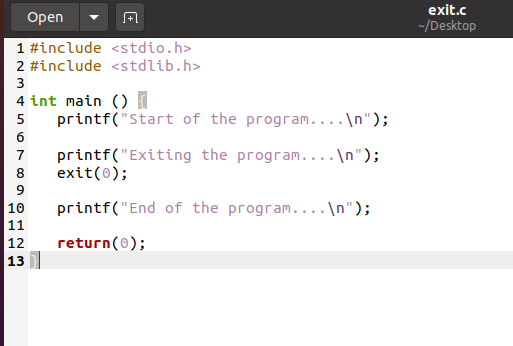
PROCESS CONTROL SYSTEM CALLS

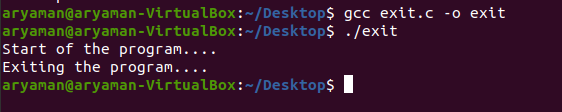
FORK



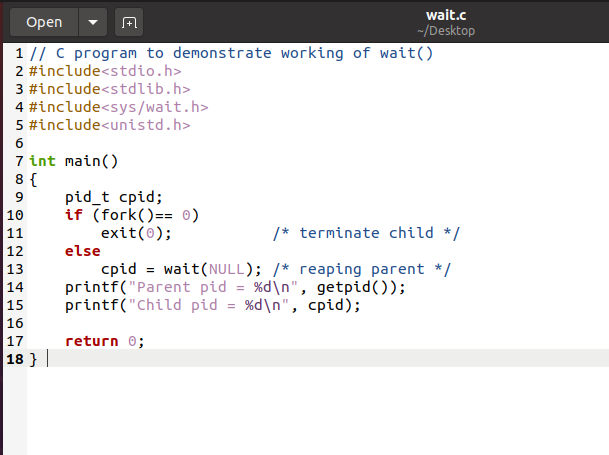


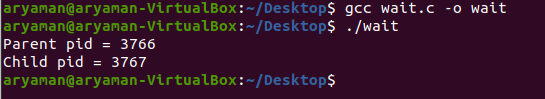
EXIT





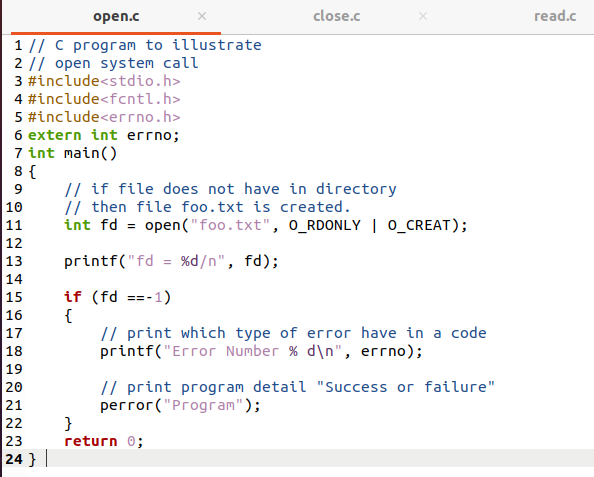
WAIT





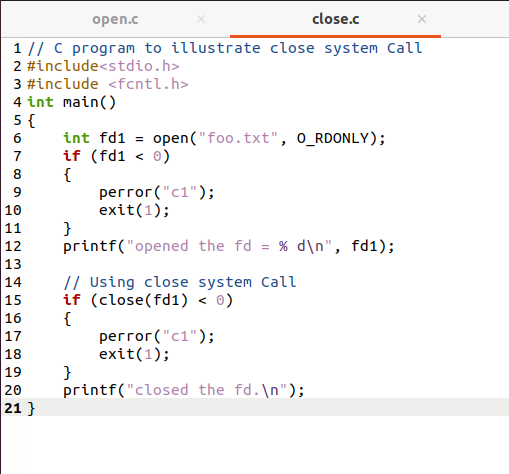
FILE MANIPULATION SYSTEM CALLS

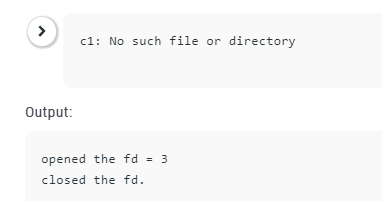
OPEN



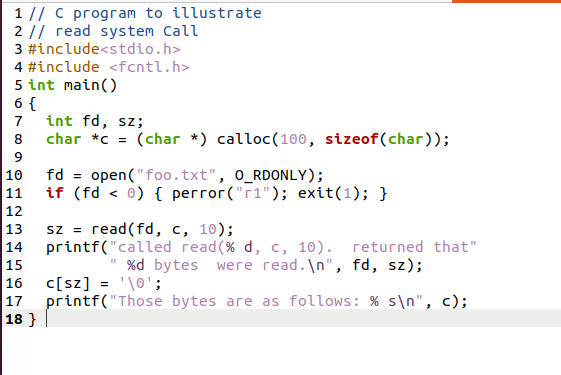


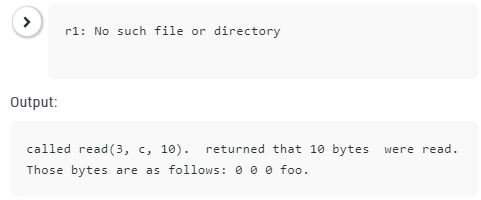
CLOSE

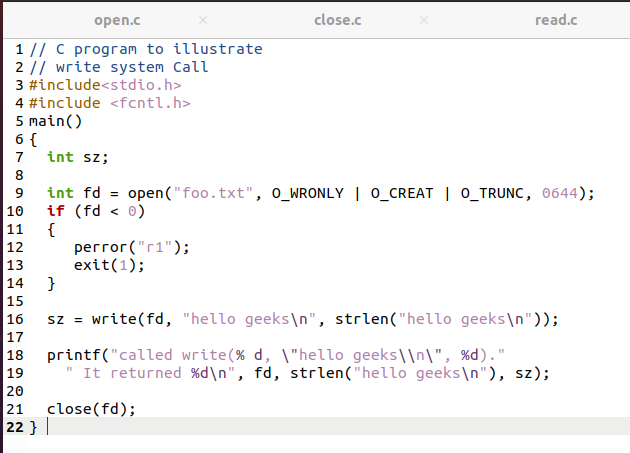




READ





WRITE



DEVICE MANIPULATION SYSTEM CALLS

ioctl()

ioctl() - Unix, Linux System Call

NAME

ioctl - control device

SYNOPSIS

**#include <sys/ioctl.h>**

**int ioctl(int***d***, int***request***, ...);**

DESCRIPTION

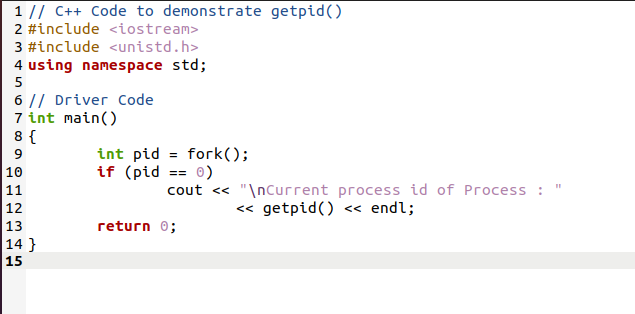
The **ioctl**() function manipulates the underlying device parameters of special files. In particular, many operating characteristics of character special files (e.g. terminals) may be controlled with **ioctl**() requests. The argument *d* must be an open file descriptor.

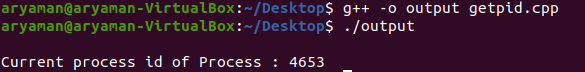
The second argument is a device-dependent request code. The third argument is an untyped pointer to memory. It’s traditionally **char \****argp* (from the days before **void \*** was valid C), and will be so named for this discussion.

An **ioctl**() *request* has encoded in it whether the argument is an *in* parameter or *out* parameter, and the size of the argument *argp* in bytes. Macros and defines used in specifying an **ioctl**() *request* are located in the file *<sys/ioctl.h>*.

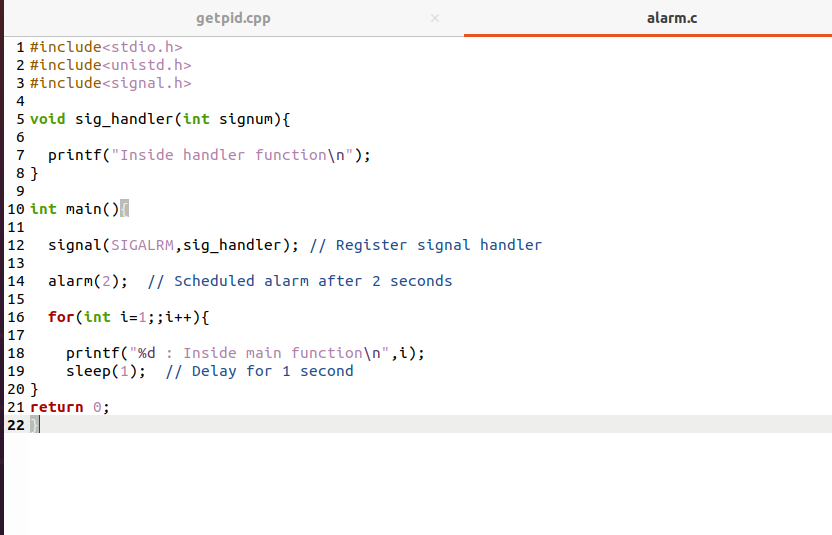
INFORMATION MAINTENANCE SYSTEM CALLS

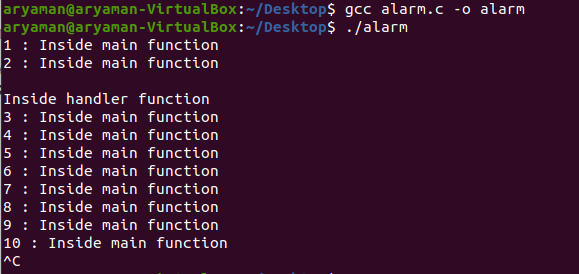
GETPID



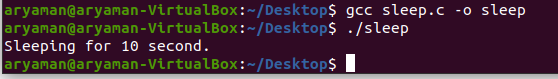


ALARM



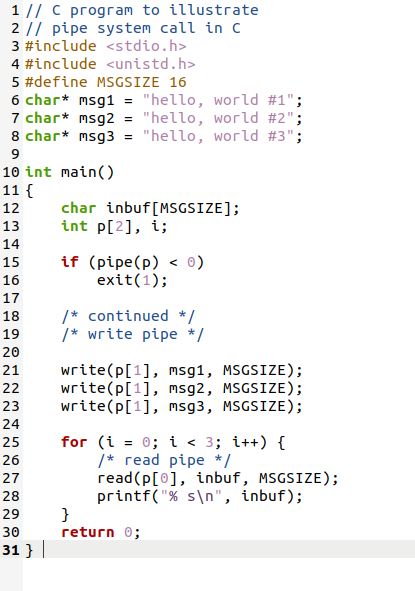


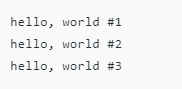
SLEEP

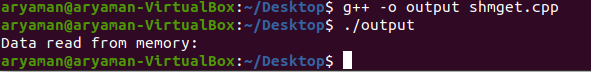


COMMUNICATION SYSTEM CALLS

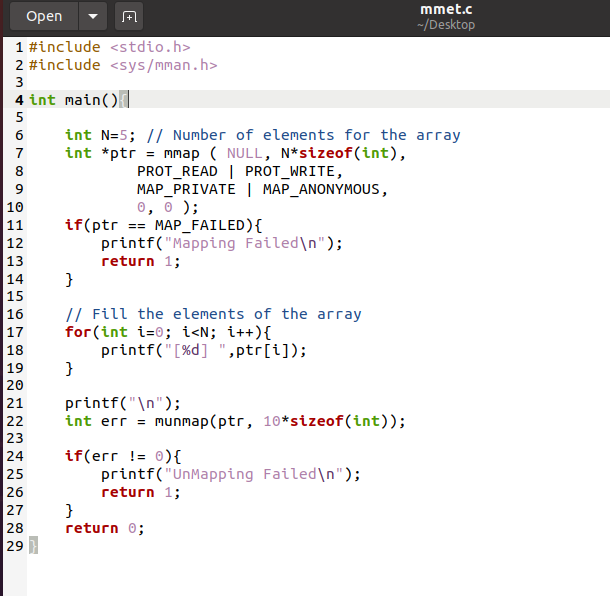
PIPE SYSTEM CALL





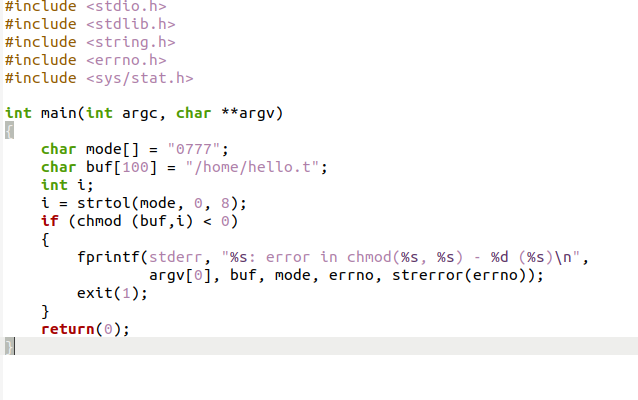
SHMGET

MMET SYSTEM CALL





PROTECTION SYSTEM CALLS

CHMOD



## UASK

## NAME         [top](https://man7.org/linux/man-pages/man2/umask.2.html#top_of_page)

umask - set file mode creation mask

## SYNOPSIS         [top](https://man7.org/linux/man-pages/man2/umask.2.html#top_of_page)

**#include <sys/types.h>**

**#include <sys/stat.h>**

**mode\_t umask(mode\_t** *mask***);**

## DESCRIPTION         [top](https://man7.org/linux/man-pages/man2/umask.2.html#top_of_page)

**umask**() sets the calling process's file mode creation mask

(umask) to *mask* & 0777 (i.e., only the file permission bits of

*mask* are used), and returns the previous value of the mask.

## CHOWN

## NAME         [top](https://man7.org/linux/man-pages/man2/lchown.2.html#top_of_page)

chown, fchown, lchown, fchownat - change ownership of a file

## SYNOPSIS         [top](https://man7.org/linux/man-pages/man2/lchown.2.html#top_of_page)

**#include <unistd.h>**

**int chown(const char \****pathname***, uid\_t** *owner***, gid\_t** *group***);**

**int fchown(int** *fd***, uid\_t** *owner***, gid\_t** *group***);**

**int lchown(const char \****pathname***, uid\_t** *owner***, gid\_t** *group***);**

**#include <fcntl.h>** /\* Definition of AT\_\* constants \*/

**#include <unistd.h>**

**int fchownat(int** *dirfd***, const char \****pathname***,**

**uid\_t** *owner***, gid\_t** *group***, int** *flags***);**

Feature Test Macro Requirements for glibc (see

[feature\_test\_macros(7)](https://man7.org/linux/man-pages/man7/feature_test_macros.7.html)):

**fchown**(), **lchown**():

/\* Since glibc 2.12: \*/ \_POSIX\_C\_SOURCE >= 200809L

|| \_XOPEN\_SOURCE >= 500

|| /\* Glibc versions <= 2.19: \*/ \_BSD\_SOURCE

**fchownat**():

Since glibc 2.10:

\_POSIX\_C\_SOURCE >= 200809L

Before glibc 2.10:

\_ATFILE\_SOURCE

## DESCRIPTION         [top](https://man7.org/linux/man-pages/man2/lchown.2.html#top_of_page)

These system calls change the owner and group of a file. The

**chown**(), **fchown**(), and **lchown**() system calls differ only in how

the file is specified:

\* **chown**() changes the ownership of the file specified by

*pathname*, which is dereferenced if it is a symbolic link.

\* **fchown**() changes the ownership of the file referred to by the

open file descriptor *fd*.

\* **lchown**() is like **chown**(), but does not dereference symbolic

links.

Only a privileged process (Linux: one with the **CAP\_CHOWN**

capability) may change the owner of a file. The owner of a file

may change the group of the file to any group of which that owner

is a member. A privileged process (Linux: with **CAP\_CHOWN**) may

change the group arbitrarily.