BiT
Bahir Dar University
Bahir Dar Institute of Technology

Faculty of Computing

Department of Software Engineering

Shutdown - System call

Student info:

Name : HENOK ABRHAM

ID : BDU1307206

DEPT: SOFTWARE ENGINEERIMG

Submitted to:lecturer wendimu Baye

Submission date: 16/11/2014 e.c

Shutdown- System call
shutdown - shut down part of a full-duplex connection on the socket associated with the file descriptor socket to be shut down.
The shutdown system call is used to halt the system. It can be invoked only from shell of the root user. It terminates all the running processes,and halts the system. Or it used on all types of sockets to disable reception, transmission, or both.

code implementation in C language

```
#include <sys/socket.h>

       int shutdown(int sockfd, int
how);
```

describtion list of parameters and flags

The constants SHUT_RD, SHUT_WR, SHUT_RDWR have the value 0, 1, 2, respectively, and are defined in <sys/socket.h> .

The shutdown() call causes all or part of a full-duplex
connection on the socket associated with *sockfd* to be shut down.
sockfd − It is a socket descriptor returned by the socket function.

socket() creates an endpoint for communication and returns a descriptor.
Return value:
On success, a file descriptor for the new socket is returned. On error, -1 is returned, and errno is set appropriately.

If *how* is SHUT_RD, further receptions will be disallowed.
If *how* is SHUT_WR, further transmissions will be disallowed.
If *how* is SHUT_RDWR, further receptions and transmissions will be disallowed.

Return value
On success, zero is returned.
On error, -1 is returned, and
errno is set to indicate the error.

Errors

EBADF *sockfd* is not a valid file descriptor.
EINVAL An invalid value was specified in *how* (but see BUGS).
ENOTCONN The specified socket is not connected.
ENOTSOCK The file descriptor *sockfd* does not refer to a socket.

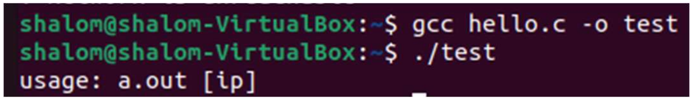
Example:

1. If *how* is SHUT_RD,

```
   #include <sys/socket.h>
   int main(){
  int shutdown(int sockfd, int SHUT_RD);
 return 0;
 }
```

   // sockfd = file descriptor numbers & **SHUT_RD = 0;**

   output: **// further receptions will be disallowed.**



```
shalom@shalom-VirtualBox:~$ gcc hello.c -o test
shalom@shalom-VirtualBox:~$ ./test
usage: a.out [ip]
```

2. If *how* is SHUT_WR,

```
   #include <sys/socket.h>
   Int main() {
 int shutdown(int sockfd, int SHUT_WR);

 return 0;
 }
```

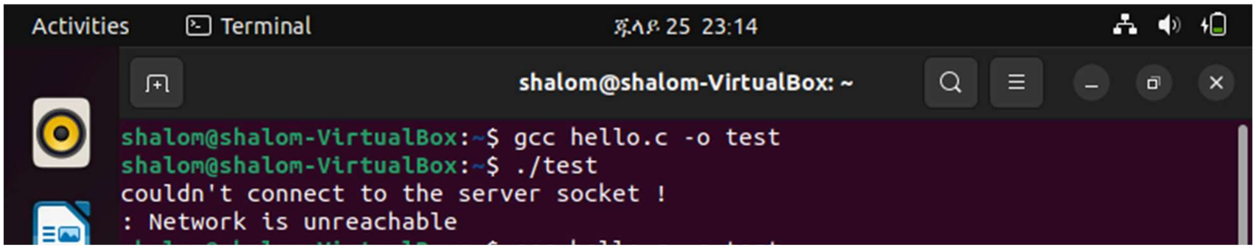
   // sockfd = file descriptor numbers & **SHUT_WR = 1;**

   output: **// further transmissions will be disallowed.**

3. If *how* is SHUT_RDWR,

```
   #include <sys/socket.h>

  int shutdown(int sockfd, int SHUT_RDWR);
```

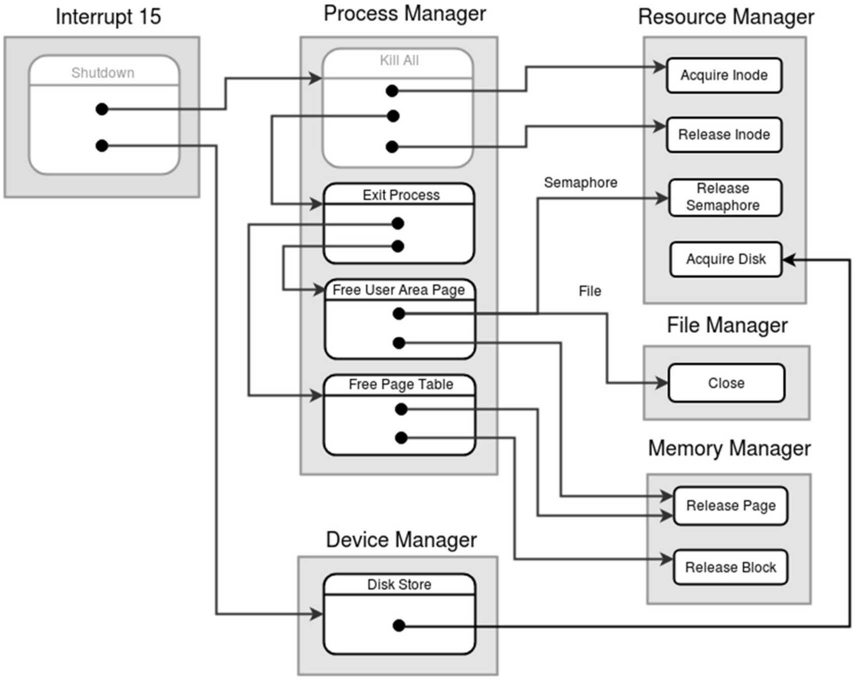


```
shalom@shalom-VirtualBox:~$ gcc hello.c -o test
shalom@shalom-VirtualBox:~$ ./test
couldn't connect to the server socket !
: Network is unreachable
```

// sockfd = file descriptor numbers & **SHUT_RDWR = 2;** *the same as close()*

output: **// further receptions and transmissions will be disallowed.but in my call the ip address is unreachable**

Algorithm of shutdown system call in OS



Process manager is part of OS, it in involves the creation, scheduling, termination of the process in the operating system.
Resource manager allows multiple programs to be in memory and run at the same time.
Device manager an operating system means controlling the Input/Output devices like disk, microphone, keyboard, printer, magnetic tape, USB port s …
File manager is used for file maintenance (or management) operations.
Memory manager is part of an operating system which handles or manages primary memory and moves processes back and forth between main memory and disk during execution.