select()

- A file descriptor represents a file.
- Two possible operations on a file:
    - Read
    - Write
    - There is a third one: Exceptions in protocol. Not of any concern to us.
- Select () monitors given files (represented via file descriptors) for the above three operations for a given amount of time.
- The syntax of select() is as follows:

#include <sys/select.h>

int select(int *nfds*, **fd_set** **readfds*, **fd_set** **writefds*,
fd_set **exceptfds*, **struct timeval** **utimeout***);**

Arguments:

1. *nfds*: Recall that a file descriptor is an integer value greater than or equal to zero.
   This argument is maximum of all the file descriptors specified in other arguments.
2. *readfds:* File descriptors that are to be monitored for reading.
3. *writefds:* File descriptors that are to be monitored for writing.
4. *exceptfds:* File descriptors to be monitored for exceptions in protocols. Ignore right now.
5. *utimeout:* Maximum time to wait if no file descriptor (for reading, writing or exceptions) is not ready.
   If specified as `NULL`, then select blocks until a file descriptor is ready for either of the three functions.
   If specified as 0, then select() checks the file descriptors and returns, without any wait.

Data structure for the *utimeout:*

```
struct timeval {
     time_t tv_sec;    /* seconds */
     long tv_usec;     /* microseconds */
 };
```

Both the arguments of `timeval` are integers i.e. can be assigned an integer.

The arguments *readfds, writefds and exceptfds* are abstract data types.
Though they are bit vectors, but they should not be treated as one.
How to assign values to these arguments?
Use the following macros:

```
void FD_CLR(int fd, fd_set *set);
 int  FD_ISSET(int fd, fd_set *set);
 void FD_SET(int fd, fd_set *set);
 void FD_ZERO(fd_set *set);
```

One can test if a file descriptor is still present in a set with the **FD_ISSET**() macro. **FD_ISSET**() returns nonzero if a specified file descriptor is present in a set and zero if it is not. **FD_CLR**() removes a file descriptor from a set.

FD_SET() adds the file descriptor specified by *fd* to the fd-set specified by *set.*
FD_ZERO() removes all the file descriptors for the fd-set specified by *set.*

RETURN VALUE:

On success, **select**() returns the total number of file descriptors still present in the file descriptor sets.

If **select**() timed out, then the return value will be zero. The file descriptors set should be all empty (but may not be on some systems).

A return value of -1 indicates an error, with *errno* being set appropriately. In the case of an error, the contents of the returned sets and the *struct timeout* contents are undefined and should not be used.

From executions of sample programs:

- When two messages are available at server before invocation of select each in a different *fd*, then select returns immediately returning both *fds.*
- Select waits only if none of the *fds* specified in the fd sets is *ready.*
- Select returns as soon as an *fd* becomes ready.

Sample codes have been added to the directory at https://goo.gl/eVLjaz.