Computer Networks Lab

Instructors:
Ahmed Nawaz, Shehr Bano & Sara Afzal
Email id:
ahmed.nawaz@nu.edu.pk
shehr.bano@nu.edu.pk
sara.afzal@nu.edu.pk

Lab 7:
Socket Programming:
Select Function

Select() Function

- This function allows the process to instruct the kernel to for wait any one of multiple events to occur and to wake up the process only when one or more of these events has occurred or when a specified amount of time has passed.

- As an example, we can call **select** and tell the kernel to return only when:
  - Any of the descriptors in the set {1, 4, 5} are ready for reading
  - Any of the descriptors in the set {2, 7} are ready for writing
  - Any of the descriptors in the set {1, 4} have an exception condition pending
  - T seconds have elapsed

Select() Function

```
int select (int
maxfd1, fd_set *readset,
fd_set *writeset,
fd_set *exceptset
const struct timeval *timeout);
```

- maxfd1 - the max number of descriptors in the three sets
- three independent sets of descriptors are watched.
  - *readfds* - if a read is ready (one or more fd)
    - characters become available for reading
    - end of file reached
  - *writefds* - if a write is ready
  - *exceptfds* – if there are exceptions.
- timeout - time to wait
  - **NULL**, => block until one of the event occurs – wait forever
  - 0, => returns immediately after checking. (i.e. polling) – don't wait

Select Function

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

- If `timeout` is set to NULL, then `select` will wait until one of the event occurs
- If `timeout` is set to 0, then `timeout` returns immediately after checking. This is called polling
- Wait for a specified amount of time

Select() Function: Macros

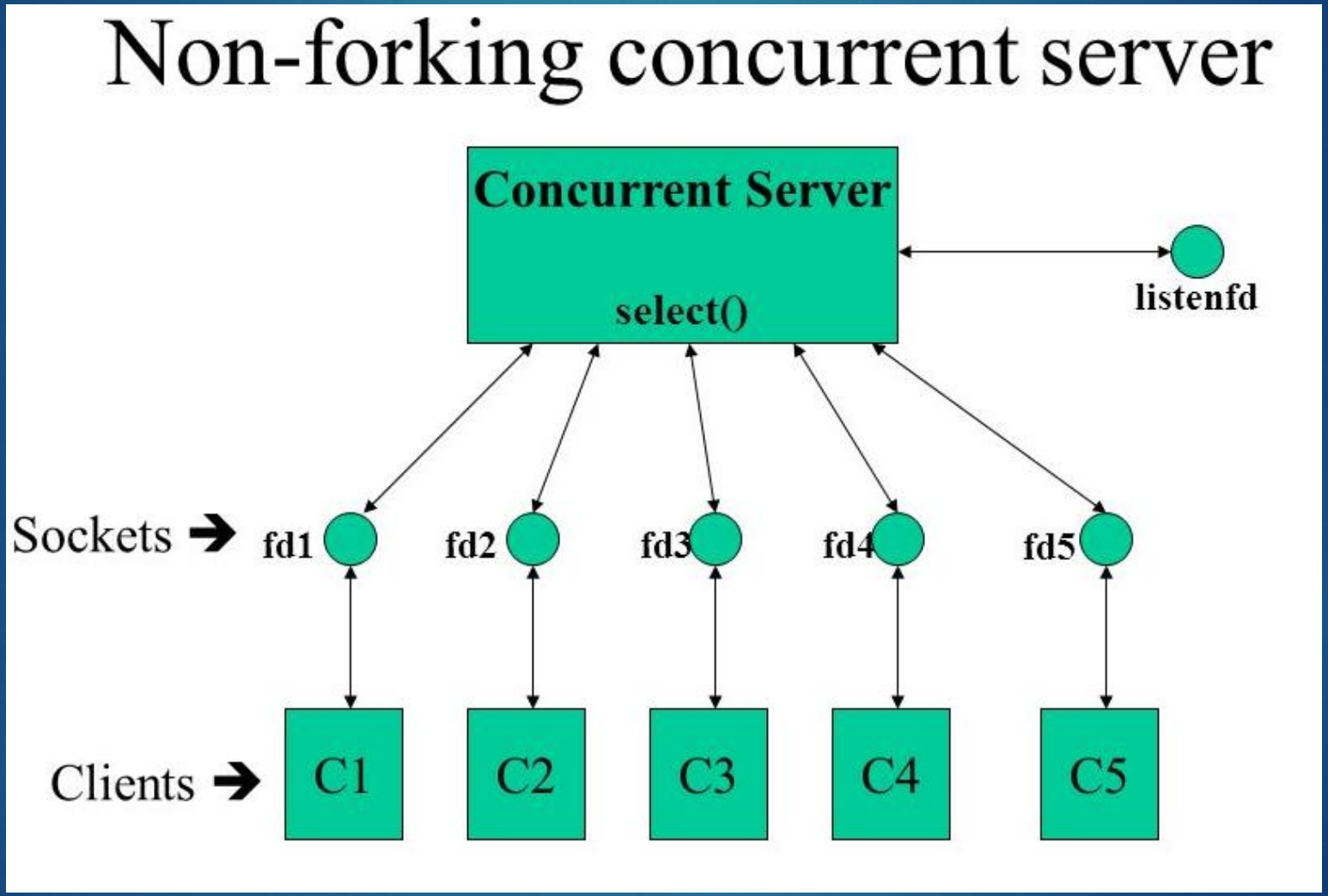
`FD_SET(int fd, fd_set *set);`	Add *fd* to the *set*.
`FD_CLR(int fd, fd_set *set);`	Remove *fd* from the *set*.
`FD_ISSET(int fd, fd_set *set);`	Return true if *fd* is in the *set*.
`FD_ZERO(fd_set *set);`	Clear all entries from the *set*.

For example, to define a variable of type fd_set and then turn on the bits for descriptors 1,4,5 we write

fd_set rset

```
FD_ZERO(&rset);          /* initialize set: all bits off */
FD_SET(1,&rset);         /*turn on bit for fd 1 */
FD_SET(4,&rset);         /* turn on bit for fd 4*/
FD_SET(5,rset);          /*turn on bit for fd 5*/
```

Exercise: Design a Non-forking concurrent server using select()



Lab Task

Question 1:

Write a program to create a non-forking concurrent Client-Server Chat Program using select ( ). The server must be able to handle at least three clients concurrently.

Thank You