

Advanced I/O

W4118 Operating Systems I

<https://cs4118.github.io/www/2024-1/>

Nonblocking I/O

Two ways to make “slow” system calls nonblocking:

- call `open()` with `O_NONBLOCK`
- call `fcntl()` to turn on `O_NONBLOCK` file status flag
 - file status flag is part of the file table entry

Nonblocking slow system call returns -1 with `errno` set to `EAGAIN` if it would have blocked

I/O Multiplexing

Network example: How can we monitor two connection simultaneously?

I/O Multiplexing

Network example: How can we monitor two connection simultaneously?

1. Nonblocking reads alternating between the two connection
2. Kernel I/O multiplexing

I/O Multiplexing

select () API for I/O multiplexing

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

int select(int maxfdp1, // max fd plus 1, or simply pass FD_SETSIZE
           fd_set *restrict readfds, // see if they're ready for reading
           fd_set *restrict writefds, // see if they're ready for writing
           fd_set *restrict exceptfds, // see if exceptional condition occurred
                                           // ex) urgent out-of-band data in TCP
           struct timeval *restrict tvptr); // timeout

// Returns: count of ready descriptors, 0 on timeout, -1 on error

int FD_ISSET(int fd, fd_set *fdset);

// Returns: nonzero if fd is in set, 0 otherwise

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