```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/types.h>
#include <unistd.h>

int
main(void)
{
    fd_set allfds;

    // open files of interest
    int livefd1 = open("livefile1.txt", O_RDONLY);
    if (-1 == livefd1) {
        perror("could not open file");
        return 1;
    }
    int livefd2 = open("livefile2.txt", O_RDONLY);
    if (-1 == livefd2) {
        perror("could not open file");
        return 1;
    }
    char buffer[10];

    // nfds is the highest-numbered file descriptor in any
     of
    // the three sets, plus 1.
    int nfds = livefd2 > livefd1 ? (livefd2+1) :
     (livefd1+1);

    printf("set fd 0\n");
    printf("set fd %d\n", livefd1);
    printf("set fd %d\n", livefd2);
    printf("nfds: %d\n", nfds);

    // how often to check?
    long delay_sec = 3;
    struct timeval tv;

    int data_available = 1;
    while (1)
```

```
{
    // look at file for changes for reading
    FD_ZERO(&allfds);
    FD_SET(0, &allfds); // stdin
    FD_SET(livefd1, &allfds);
    FD_SET(livefd2, &allfds);

    // set delay
    tv.tv_sec = delay_sec;
    tv.tv_usec = 0;

    // select
    int ret = select(nfds, &allfds, NULL, NULL, &tv);
    if (-1 == ret) {
        perror("select() failed");
    } else if (ret) {
        data_available = 1;

        // which file descriptor?
        int is_stdin = FD_ISSET(0, &allfds);
        int is_live1 = FD_ISSET(livefd1, &allfds);
        int is_live2 = FD_ISSET(livefd2, &allfds);

        printf("Data is available for %d files [%d %d
         %d]\n",
                ret, is_stdin, is_live1, is_live2);

        ssize_t nread;
        if (is_stdin) {
            printf("data entered on cmd line: ");
            nread = read(0, buffer, 10);
            buffer[nread]= '\0';
            printf("%s\n", buffer);
            if (strncmp(buffer, "quit", 4) == 0)
                break;
        }
        if (is_live1) {
            errno = 0;
            nread = read(livefd1, buffer, 10);
            if (-1 == nread) {
                perror("could not read further");
            }
            if (0 == nread) {
                printf("live1fd end of file reached\n");
                //close(livefd1);
```

```
                    //livefd1 = -1;
                    // remove livefd1 from our fd sets
                }
                buffer[nread]= '\0';
                printf("data entered in live1fd: ");
                printf("%s\n", buffer);
            }
            if (is_live2) {
                errno = 0;
                nread = read(livefd2, buffer, 10);
                if (-1 == nread) {
                    perror("could not read further");
                }
                if (0 == nread) {
                    printf("live2fd end of file reached\n");
                    //close(livefd2);
                    //livefd2 = -1;
                    // remove livefd2 from our fd sets
                }
                buffer[nread]= '\0';
                printf("data entered in live2fd: ");
                printf("%s\n", buffer);
            }
        } else {
            if (data_available == 1)
                printf("No data within %ld seconds\n",
                 delay_sec);
            else
                printf(".");
            data_available = 0;
            usleep(100);
        }
    }

    // free your resources!
    FD_ZERO(&allfds); // clear all fds
    close(livefd1);
    close(livefd2);

    return 0;
}
```