

```

#include <stdio.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>

volatile int flag = 0;

// signal handler
void interrupted(int val) {
    flag = 1;
}

int main() {
    void (*old_sig_int_handler)(int);
    int res;

    // get the old handler
    old_sig_int_handler = signal(SIGINT, interrupted);
    if (old_sig_int_handler == SIG_ERR) {
        perror("could not change signal handler");
        return -1;
    }

    char buffer[100];
    flag = 0;
    ssize_t result = read(0, buffer, 100);

    // check for errors
    int error_val = errno;
    if (error_val != 0) {
        printf("\n");
        printf("error_val: %d\n", error_val);
        printf("read() was interrupted by a signal\n");
        printf("flag is: %d\n", flag);
        perror("hmm errno non zero ---> ");
    }

    fprintf(stderr, "managed to read: %d characters\n",
        result);

    printf("buffer contains: ");
    int i;
    for (i = 0; i < result; ++i)
        printf("_%c", buffer[i]);
    printf("\n");

```