filewrite.c

#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>

int main(int argc, char ** argv)
{
    char* fname;
    int file_descriptor;

    if (argc != 2) {
        fprintf(stderr, "usage: filewrite filename\n");
        exit(1);
    }

    /* this next line takes the user's argument and stores it */
    fname = argv[1];

    /* a system call */
    file_descriptor = open(fname, O_WRONLY | O_CREAT);

    /* an error check */
    if (file_descriptor < 0) {
        fprintf(stderr, "Major problem, dude\n");
        exit(1);
    }

    /* this is the interesting line */
    write(file_descriptor, "this text says nothing at all", 29);

    /* cleanup (why do we need this?) */
    close(file_descriptor);
    exit(0);
}

our_yes.c

#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <stdio.h>

int main(int argc, char ** argv)
{
    char* repeated;
    int len;
    char* newline = "\n";

    /* check to make sure the user gave us one argument */
    if (argc != 2) {
        fprintf(stderr, "usage: our_yes string_to_repeat\n");
        exit(1);
    }

    /* put the user's argument in a variable */
    repeated = argv[1];

    /* find out how long the argument is */
    len = strlen(repeated);

    /* loop forever */
    while (1) {
        /* what is going on here? */
        write(1, repeated, len);

        /* what about here? */
        write(1, newline, 1);

        sleep(1);
    }
}
our_head.c

/* our_head.c -- A C program that prints the first L lines of its input,
   where L defaults to 10 but can be specified by the caller of the
   program.
*/

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>

int main(int argc, char** argv)
{
    int i = 0;
    int nlines;
    char ch;
    int ret;

    /* next lines say:
    * if user supplied an argument, i.e., the # of lines, put that
    * argument into the 'nlines' variable.
    * if user did not supply an argument, set the # of lines to be 10
    */
    if (argc == 2)
        nlines = atoi(argv[1]);
    else if (argc == 1)
        nlines = 10;
    else
        fprintf(stderr, ": usage: our_head [nlines]\n";
        exit(1);

    /* we'll loop for a maximum of 'nlines'. */
    for (i = 0; i < nlines; i++) {
        do {
            /* read in the first character from file_descriptor 0 */
            ret = read(0, &ch, 1);
            /* if there are no more characters to read, then exit */
            if (ret == 0) exit(0);
            /* what does this line do? .... */
            write(1, &ch, 1);
            /* .... what about this next line? */
            while (ch != '\n');
        } while (ch != '\n');

        exit(0);
    }
}