

```

#define BUF_SIZE 10

int main(int argc, char *argv[]) /* Over-simplified! */
{
    int pfd[2]; /* Pipe file descriptors */
    char buf[BUF_SIZE];
    ssize_t numRead;

    pipe(pfd); /* Create the pipe */

    switch (fork()) {
    case 0: /* Child - reads from pipe */
        close(pfd[1]); /* Write end is unused */

        for(;;) { /* Read data from pipe, echo on stdout */
            if( (numRead = read(pfd[0], buf, BUF_SIZE)) == 0 )
                break; /* End-of-file */
            if( write(1, buf, numRead) != numRead ){}
        }
        puts(""); /* newline */

        close(pfd[0]); _exit(EXIT_SUCCESS);

    default: /* Parent - writes to pipe */
        close(pfd[0]); /* Read end is unused */

        if( (size_t)write(pfd[1], argv[1], strlen(argv[1])) != strlen(argv[1]) ){}

        close(pfd[1]); /* Child will see EOF */

        wait(NULL); /* Wait for child to finish */
        exit(EXIT_SUCCESS);
    }
}

```