Lab 10

Input / Output: A Buffered I/O Library

Purpose

To learn how buffered I/O operates and to compare the performance of buffered and unbuffered I/O.

Background Reading And Preparation

Read Chapters 13 through 15 to learn about I/O in general, and read Chapter 16 to learn about buffering.

Overview

Build three C procedures, *buf_in*, *buf_out*, and *buf_flush* that implement buffered I/O. On each call, procedure *buf_in* delivers the next byte of data from file descriptor zero. When additional input is needed from the device, *buf_in* reads sixteen kilobytes of data into a buffer, and allows successive calls to return values from the buffer. On each call, procedure *buf_out* writes one byte of data to a buffer. When the buffer is full or when the program invokes procedure *buf_flush*, data from the buffer is written to file descriptor one.

Procedure And Details (checkmark as each is completed)

. . . . . . 1. Implement procedure *buf_in*.

. . . . . . 2. Verify that *buf_in* operates correctly for input of less than sixteen kilobytes (i.e., less than one buffer of data).

. . . . . . 3. Redirect input to a large file, and verify that *buf_in* operates correctly for input that spans multiple buffers.

. . . . . . 4. Implement procedures *buf_out* and *buf_flush*.

. . . . . . 5. Verify that *buf_out* and *buf_flush* operate correctly for output of less than one buffer.

. . . . . . 6. Verify that *buf_out* and *buf_flush* operate correctly for output that spans multiple buffers.

Optional Extensions (checkmark as each is completed)

. . . . . . 7. Compare the performance of procedures *buf_in*, *buf_out*, and *buf_flush* to the performance of unbuffered I/O (i.e., *read* and *write* of one byte) for various size files. Plot the results.

. . . . . . 8. Measure the performance of *buf_in*, *buf_out*, and *buf_flush* for various size buffers when copying a large file. Use buffers that range in size from 4 bytes to 100 Kbytes, and plot the results.

Notes