

Linux Software Handshaking Hints

Abstract:

This document explains the procedures used for efficient, robust software handshaking. The information contained in this document is geared more towards the developer.

Setting the LOW LATENCY flag

The default serial port configuration (both PC and IPm) may allow up to 40 characters to be sent out the serial port, after XOFF has been received. Typically, you will see 32 characters or less. However, because Linux has a default delay of 1 jiffy (10ms), before it sends characters out to the line discipline, it is possible to have pending characters sent out, even though the XOFF has been sent. This is because the line discipline has not been given the XOFF yet. This results in the approximately 32 characters being sent even after the XOFF has gone into the port. The serial port hardware itself uses a hardware 8-character FIFO on transmit and receive, so the FIFO depth can also introduce some delay in response to an XOFF character as well.

Now it is possible to reduce the size of the serial buffers and/or the FIFO sizes. However, the size is fixed and needs to be compiled into the kernel. By further reducing the default buffer sizes, we also decrease serial/system performance because the kernel is interrupting much more frequently.

There will be devices, using XON/XOFF, with limited buffer space (and that 40 characters may be excessive). In these situations there is a way to bypass the 1-jiffy delay when sending characters to the line discipline. This is known as “low latency” mode. Those applications that need better performance should set the serial port for 'low latency'. Setting the low latency flag will immediately pass characters off to the line discipline, bypassing the character queue and 10ms delay. Of course, the performance increase comes with higher CPU utilization.

The FIFO size still can introduce some delay in response, but it is comparatively small when compared to the 10ms delay, especially at higher baud rates.

setserial

setserial is a program designed to set and/or report the configuration information associated with a serial port. setserial resides in /bin and is included in most standard Linux distributions.

#setserial /dev/ttyS1 Displays current settings.

#setserial /dev/ttyS1 low_latency Sets the low_latency flag.

#setserial /dev/ttyS1 ^low_latency Unsets the low_latency flag.

For more information on the setserial command, consult the setserial man pages.

***ioctl()* – TIOCGSERIAL / TIOCSSERIAL**

The program below illustrates how to enable the `low_latency` flag for a given serial port (`/dev/ttyS1`), using `ioctl()` calls.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <linux/serial.h>
#include <fcntl.h>
#include <unistd.h>
#include <errno.h>
extern int errno;
int main(int argc, char **argv)
{
    int fd;
    struct serial_struct serial_info;
    /* Open the port */
    if ((fd = open("/dev/ttyS1", O_RDWR | O_NOCTTY |
O_NONBLOCK)) == -1) {
        fprintf(stderr, "open() failed: %s\n",
strerror(errno));
        exit (1);
    }
    /* Get the current serial port settings */
    if (ioctl(fd, TIOCGSERIAL, &serial_info) != 0) {
        fprintf(stderr, "TIOCGSERIAL ioctl() failed: %s\n",
strerror(errno));
        exit (1);
    }
    /* Enable the low_latency flag */
    serial_info.flags |= ASYNC_LOW_LATENCY;
    /* Update the current settings */
    if (ioctl(fd, TIOCSSERIAL, &serial_info) != 0) {
        fprintf(stderr, "TIOCSSERIAL ioctl() failed: %s\n",
strerror(errno));
        exit (1);
    }
    close(fd);
    exit (0);
}
```

Controlling XON/XOFF

There may be a time when a developer needs to force out an XON/XOFF character, or suspend/restart output. Although these issues could be address via ioctl() calls, the following POSIX wrapper function exists:

```
#include <termios.h>
#include <unistd.h>
```

```
int tcflow(int fd, int action);
```

tcflow() suspends transmission or reception of data on the object referred to by fd, depending on the value of action:

TCOOFF	suspends output.
TCOON	restarts suspended output.
TCIOFF	transmits a STOP character, which stops the terminal device from transmitting data to the system.
TCION	transmits a START character, which starts the terminal device transmitting data to the system.

The default on open of a terminal file is that neither its input nor its output is suspended. For more information on tcflow() or termios, consult the termios man pages.