

## 第十八章 终端I/O

习题：

18-1、执行命令前后，可以用stty -a来对比；

```
#include <stdio.h>
#include <unistd.h>           // For STDIN_FILENO;
#include <termios.h>
#include <errno.h>
static struct termios save_termios;
static int ttysavefd = -1;
static enum { RESET, RAW, CBREAK } ttystate = RESET;

int tty_raw(int fd) {
    int err;
    struct termios buf;
    if (RESET != ttystate) {
        errno = EINVAL;
        return -1;
    }
    if (0 > tcgetattr(fd, &buf)) {
        return -2;
    }
    save_termios = buf;
    buf.c_lflag &= ~(ECHO | ICANON | IEXTEN | ISIG);
    buf.c_iflag &= ~(BRKINT | ICRNL | INPCK | ISTRIP | IXON);
    buf.c_cflag &= ~(CSIZE | PARENB);
    buf.c_cflag |= CS8;
    buf.c_oflag &= ~(OPOST);
    buf.c_cc[VMIN] = 1;
    buf.c_cc[VTIME] = 0;
    if (0 > tcsetattr(fd, TCSAFLUSH, &buf)) {
        return -3;
    }
}
```

```

    if (0 > tcgetattr(fd, &buf)) {
        err = errno;
        tcsetattr(fd, TCSAFLUSH, &save_termios);
        errno = err;
        return -4;
    }
    if ((buf.c_lflag & (ECHO | ICANON | IEXTEN | ISIG)) ||
        (buf.c_iflag & (BRKINT | ICRNL | INPCK | ISTRIP | IXON)) ||
        (buf.c_cflag & (CSIZE | PARENB | CS8)) != CS8 ||
        (buf.c_oflag & OPOST) || buf.c_cc[VMIN] != 1 ||
        buf.c_cc[VTIME] != 0) {
        tcsetattr(fd, TCSAFLUSH, &save_termios);
        errno = EINVAL;
        return -5;
    }
    ttystate = RAW;
    ttysavefd = fd;
    return 0;
}

int main(int argc, const char *argv[]) {
    printf("return %d\n", tty_raw(STDIN_FILENO));
    return 0;
}

```

/opt/drill\_ground/aupe/18-1.c [FORMAT=unix:utf-8] [TYPE=C] [COL=001] [R

```

tongue aupe # gcc 18-1.c -o 18-1
tongue aupe # ./18-1
return 0
tongue aupe # ^dev/pts/1
tongue aupe #

```

回显关闭，  
这里的tty三  
个字符没显  
示出来

```

tongue aupe # tty
/dev/pts/1
tongue aupe #

```

执行reset  
之后再执行  
tty命令

18-2、对BSD不感兴趣；

18-3、两边差异用红框标出；

```

tongue ~ # ps aux|grep -v grep|grep vim
root      13115  0.0  0.1 43448 3432 pts/0    S+   16:18   0:00 vim
tongue ~ # stty -a -F /dev/pts/0
speed 38400 baud; rows 34; columns 122; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>; eol2 = <undef>; swtch = <undef>; start = ^Q;
stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr -icrnl ixon -ixoff -iuclic -ixany -imaxbel -iutf8
opost -olcuc -ocrnl -onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
-isig -icanon -iexten -echo -echoe echok -echonl -noflsh -xcase -tostop -echoprtr echoctl echoke
tongue ~ # stty -a
speed 38400 baud; rows 34; columns 122; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>; eol2 = <undef>; swtch = <undef>; start = ^Q;
stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk brkint ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff -iuclic -ixany imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon -iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprtr echoctl echoke

```

