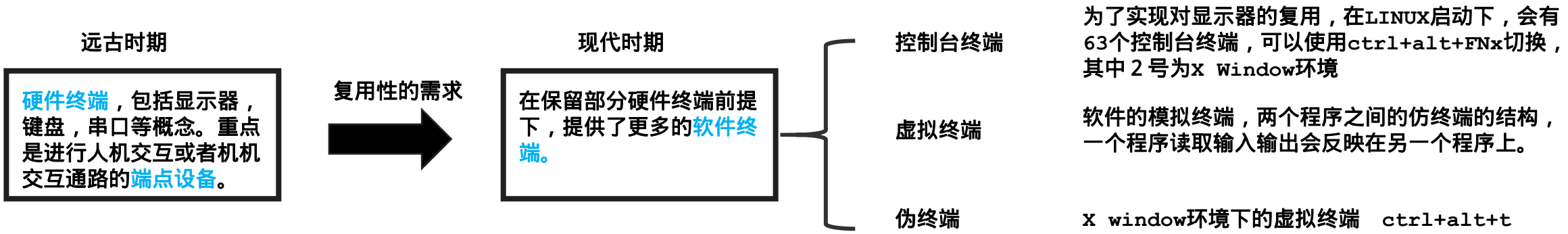
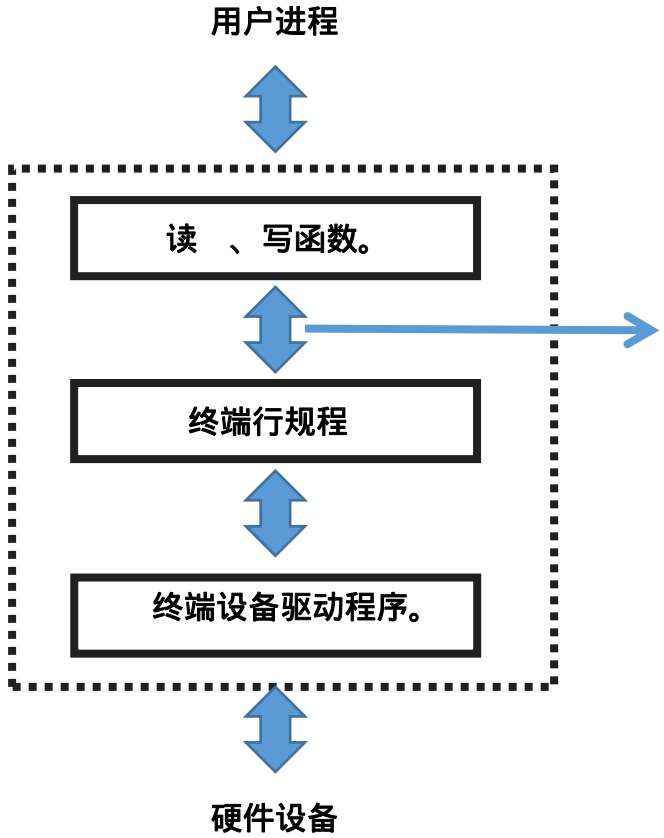
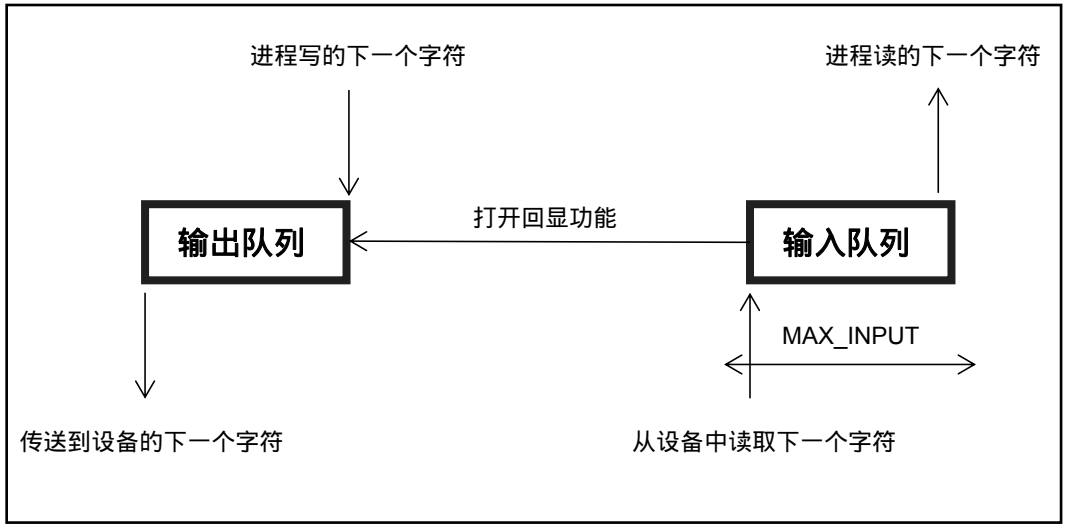
终端相关概念

1. 什么是终端？只是一个粗略的概念



进行终端操作的模块耦合作用过程





终端设备属性控制块

```
struct termios
{
    tcflag_t c_iflag; /*input flags*/
    tcflag_t c_oflag; /*output flags*/
    tcflag_t c_cflag; /*control flags*/
    tcflag_t c_lflag; /*local flags*/
    cc_t c_cc[NCCS];  /*control characters, 共11个，回车和换行不可被修改*/
}
```

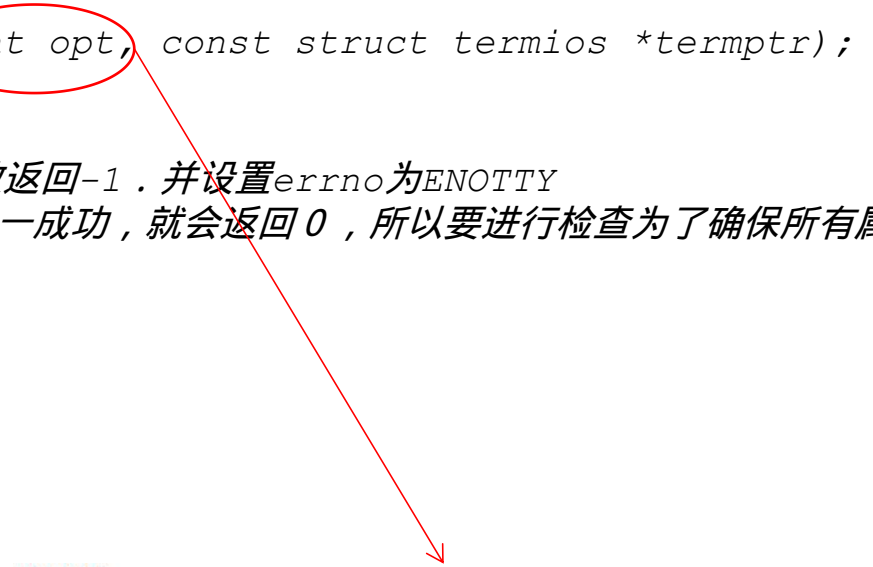
stty -a 命令来查看终端的属性

终端设备属性__获取和修改

```
int tcgetattr(int fd, struct ternios *termptr);

int tcsetattr(int fd, int opt, const struct termios *termptr);

//成功，返回0，失败，返回-1
//fd没有引用终端设备，则失败返回-1．并设置errno为ENOTTY
//只要对要设置的属性集合中之一成功，就会返回0，所以要进行检查为了确保所有属性集合中的属性都被成功设置
```



TCSANOW
the change occurs immediately.

TCSADRAIN
the change occurs after all output written to fd has been transmitted. This option should be used when changing parameters that affect output.

TCSAFLUSH
the change occurs after all output written to the object referred by fd has been transmitted, and all input that has been received but not read will be discarded before the change is made.

终端设备属性控制块__特殊字符值设置

```
struct termios
{
    tcflag_t c_iflag; /*input flags*/
    tcflag_t c_oflag; /*output flags*/
    tcflag_t c_cflag; /*control flags*/
    tcflag_t c_lflag; /*local flags*/
    cc_t c_cc[NCCS];  /*control characters, 共11个，回车和换行不可被修改*/
}
```

```
struct termios term;
修改：
        term.c_cc[VXX] = 目标值；
禁用：
        value = fpathconf(设备的文件描述符, _PC_VDISABLE)
        term.c_cc[VXX] = value;
```

终端设备属性控制块__终端选项标志

```
struct termios
{
    tcflag_t c_iflag; /*input flags*/
    tcflag_t c_oflag; /*output flags*/
    tcflag_t c_cflag; /*control flags*/
    tcflag_t c_lflag; /*local flags*/
    cc_t c_cc[NCCS];  /*control characters, 共11个，回车和换行不可被修改*/
}
```

太多了，不要记了，用的时候百度

终端设备属性控制设置__波特率设置

```
speed_t cfgetispeed(const struct termios *termios_p);

speed_t cfgetospeed(const struct termios *termios_p);

int cfsetispeed(struct termios *termios_p, speed_t speed);

int cfsetospeed(struct termios *termios_p, speed_t speed);
```

```
B0
B50
B75
B110
B134
B150
B200
B300
B600
B1200
B1800
B2400
B4800
B9600
B19200
B38400
B57600
B115200
B230400
```

终端设备行控制能力设置

```
int tcsendbreak(int fd, int duration);

int tcdrain(int fd);

int tcflush(int fd, int queue_selector);

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

//成功，返回0，出错，返回-1,fd必须引用终端设备
//drain等待所有的输出都被传递
//tcflow对输入和输出流进行控制
//tcflush冲洗缓冲区，该缓冲区和flush冲洗的不一样，这个更底层
//tcsendbreak在一个指定的时间区间内部发送连续的0值位流，duration为0，
//则传递持续0.25-0.5秒，非零则取决于实现

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.

TCIFLUSH
flushes data received but not read.

TCOFLUSH
flushes data written but not transmitted.

TCIOFLUSH
flushes both data received but not read, and data written but not transmitted.

终端设备标识获取

```
char *ctermid(char *ptr);
//成功，返回指向控制终端的指针，出错，返回NULL
//返回一个指针，该指针指向字符串”/dev/tty”

int isatty(int fd);
//若为终端设备，返回 1 ，否则，返回0

char *ttyname(int fd);
//成功，指向终端路径名的指针，出错，返回NULL
```

终端设备工作模式__规范模式

规范模式：发一个读请求，当一行输入后，终端驱动程序就返回

- *所请求的字节数满足*
- *读到行定界符*
- *接收到信号，且没有设置重启*

终端设备工作模式__非规范模式

规范模式：通过关闭termios结构中的c_lflag字段的ICANON标志来指定非规范模式， 非规范模式下，输入数据不装配成行，不处理下列特殊字符：ERASE, KILL, EOF, NL, EOL, EOL2, CR, REPRINT, STATUS, WERASE.

非规范模式下如何知道什么时候给我们返回数据呢？

VMIN, VTIME两个变量指定， 当已经读了VMIN个字节数据，或者等待超过VTIME的时间[分秒单位，等于1/10s]的时候，就通知系统返回。

终端窗口大小

```
struct winsize
{
    unsigned short ws_row;        /*rows, in characters*/
    unsigned short ws_col;        /*columns, in characters*/
    unsigned short ws_xpixel;     /*horizontal size*/
    unsigned short ws_ypixel;     /*vertical size*/
}
```

1. 使用ioctl的TIOCGWINSZ命令取得此结构的当前值
2. 使用ioctl的TIOCSWINSZ命令设置此结构的值，如果此值是新的，那么前台进程组将会收到SIGWINCH，此信号默认忽略