УПРАВЛЕНИЕ ТЕРМИНАЛЬНЫМ ВВОДОМ/ВЫВОДОМ

Системные вызовы и библиотеки Unix SVR4

ЦЕЛИ РАЗДЕЛА

По завершении этого раздела вы будете способны:

- описать аппаратный и программный интерфейс терминального ввода/вывода
- изменять характеристики терминального интерфейса ввода/вывода

ПРОГРАММНЫЙ ИНТЕРФЕЙС ВВОДА/ВЫВОДА

- open(2)
  - /dev/term/xx
  - /dev/pty
  - /dev/tty
  - возвращает дескриптор файла
- ioctl(2)
  - /usr/include/termio.h
- termios(2)
  - /usr/include/termios.h
- read(2)
- write(2)
- close(2)

ИСПОЛЬЗОВАНИЕ termios(2)

- Параметры RS232
- Отображение символов
- Задержки и табуляции
- Управление потоком
- Управляющие символы
- Эхо
- Немедленный ввод
- "Сырой" терминальный ввод/вывод

termios(2) - tcget/setattr

ИСПОЛЬЗОВАНИЕ

```
#include <termios.h>
int tcgetattr(int fildes,
  struct termios *termios_p);
int tcsetattr(int fildes,
  int optional_actions,
  const struct termios *termios_p);
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

успех - 0

неуспех - -1 и errno установлена

optional_actions

TCSANOW установить атрибуты немедленно

TCSADRAIN установить атрибуты после передачи содержимого буфера вывода

TCSAFLUSH установить атрибуты после передачи вывода и сброса ввода

ПОРЯДОК ИЗМЕНЕНИЯ ТЕРМИНАЛЬНЫХ АТРИБУТОВ

```
2 struct termios tty, savetty;
3 fd = open("/dev/tty", O_RDWR);
4 tcgetattr(fd, &tty); 5 savetty = tty;
6 /* modify struct termios tty members */
7 tcsetattr(fd, TCSANOW, &tty);
8 /* use terminal */
9 tcsetattr(fd, TCSAFLUSH, &savetty);
```

СТРУКТУРА termios

```
#define NCCS 19
struct termios {
  tcflag_t c_iflag; /* input modes */
  tcflag_t c_oflag; /* output modes */
  tcflag_t c_cflag; /* control modes*/
  tcflag_t c_lflag; /* local modes */
  cc_t c_cc[NCCS]; /* control chars */
};
```

Управляющие символы

индекс	c_cc[индекс]	
VINTR	ETX (CTRL-C)	в старых юниксах - DEL
VQUIT	FS (CTRL-|)	
VERASE	BS (CTRL-H)	в старых юниксах - #
VWERASE	ETB (CTRL-W)	
VKILL	NAK (CTRL-U)	в старых юниксах - @
VEOF	EOT (CTRL-D)	
VSTOP	DC1 (CTRL-S)	
VSTART	DC3 (CTRL-Q)	
VSUSP	EM (CTRL-Z)	
VDISCARD	SI (CTRL-O)	
VLNEXT	SYN (CTRL-V)	
VREPRINT	DC2 (CTRL-R)	

c_iflag

	IGNBRK	Игнорировать условие разрыва линии
	BRKINT	Посылать сигнал прерывания при разрыве линии
c_iflag	ISTRIP	Срезать старший бит у символов
	ICRNL	Преобразовывать CR в NL при вводе
	IXON	Разрешить старт/стоповое управление вводом
	IXANY	Любой символ возобновляет вывод

c_oflag

	OPOST	Постобработка вывода
c_oflag	ONLCR	Преобразовывать NL в CR-NL при выводе
	TAB3	Преобразует табуляцию в пробелы

c_cflag

	B1200	1200 бит/сек
	B2400	2400 бит/сек
	CS7	Семибитные символы
c_cflag	CS8	Восьмибитные символы
	CSTOPB	Посылать два стоповых бита (иначе - один)
	PARENB	Разрешить контроль четности
	PARODD	Проверять нечетность, иначе - четность

c_lflag

	ISIG	Разрешить сигналы
	ICANON	Канонический ввод (забой и стирание строки)
c_lflag	ECHO	Разрешить эхо
	ECHOE	Эхо для символа очистки BS-SP-BS
	ECHOK	Выдавать NL после символа стирания строки
	IEXTEN	Разрешить функции расширения

НЕКАНОНИЧЕСКИЙ ВВОД

- Сбросить флаг ICANON в c_lflag
- Установить MIN и TIME в соответствующие значения.
  - MIN ссылается на c_cc[VMIN]
  - TIME ссылается на c_cc[VTIME]

  MIN > 0, TIME > 0 MIN символов получено или истекло межсимвольное время

  MIN > 0, TIME = 0 MIN символов получено, TIME не играет роли

  MIN = 0, TIME > 0 один символ получен или истекло время с момента запроса

  MIN = 0, TIME = 0 возвращает управление немедленно, считываются только те символы, которые уже находятся в буфере

Управление заданиями

ИСПОЛЬЗОВАНИЕ

```
pgid_t tcgetpgrp ( int fd );
int tcsetpgrp ( int fd,
  pgid_t pgrpid );
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

tcgetpgrp – группа процессов первого плана

tcsetpgrp – успех/неуспех