

```
printf("SIGWINCH received\n");
pr_winsize(STDIN_FILENO);
return;
}
```

ウィンドウ端末でプログラム 11.12 を実行するとつぎの結果を得る。

\$ a.out	
35 rows, 80 columns	はじめのサイズ
SIGWINCH received	ウィンドウサイズを変更、シグナルを捕捉
40 rows, 123 columns	
SIGWINCH received	再度シグナルを捕捉
42 rows, 33 columns	
^? \$	割り込みキーをタイプして終了する

11.13 termcap, terminfo, curses

termcap は、“terminal capability(端末機能)”の略であり、テキストファイル/etc/termcap とこのファイルを読み取るルーティン群を指す。termcap の技法は、vi エディタのためにパークレーで開発された。termcap ファイルにはさまざまな端末の記述が収められている。つまり、(行レーで開発された。termcap ファイルにはさまざまな端末の記述が収められている。つまり、(行数やコラム数、バックスペースを使えるかなどの) 端末の機能、(画面消去、カーソル移動などの) 操作を行う方法である。これらの情報をコンパイルしたプログラムから分離し、容易に編集できるテキストファイルに収めたことで、vi エディタは多くの異なる端末で動作できた。

その後、termcap ファイルのためのルーティン群は vi エディタから分離され、別の curses ライブラリに収められた。このライブラリには多くの機能が追加され、画面を扱うプログラムには便利である。

termcap の技法は完全ではない。より多くの端末をデータファイルに追加するにつれ、特定の端末を探すためのファイルの走査に時間がかかるようになった。端末の異なる属性を識別するために、データファイルでは 2 文字の名称を用いる。このような欠陥のために、terminfo の技法と関連した curses ライブラリが開発された。terminfo における端末の記述は、基本的にはテキスト記述をコンパイルしたものであり、実行時には高速に走査できる。terminfo は SVR2 で採用され、以後、すべてのシステム V リリースにある。

SVR4 は terminfo を用いているが、4.3+BSD は termcap を用いている。

terminfo と curses ライブラリに関する記述は、[Goodheart 1991] にある。[Strang, Mui, and O'Reilly 1991] には termcap と terminfo に関する記述がある。

termcap と terminfo のいずれも、それ単独では本章で見た問題、つまり、端末モードの変更、端末の特別な文字の変更、ウィンドウサイズの変更などを解決しない。この 2 つは、(画面消去、カーソル移動などの) 典型的な操作をさまざまな種類の端末で行う方法を提供するのである。一方、curses は本章で述べたいいくつかの問題の解消に役立つ。curses が提供する関数は、ローモードの設定、cbreak モードの設定、エコーのオン/オフなどを行う。しかし、curses は文字ベースの

端末向けに設計されているが、現在の傾向はピクセルベースのグラフィックス端末である。

11.14 まとめ

端末にはさまざまな機能やオプションがあり、そのほとんどは必要に応じて変更できる。本章では、入力文字やオプションフラグなどの端末操作を変更する数々の関数を説明した。端末のすべての特別な文字を調べ、端末装置に設定したり解除する多くのオプションを見た。

端末入力には、カノニカルモード(行単位)と非カノニカルモードの 2 つのモードがある。2 つのモードの例を示し、POSIX.1 の端末オプションと BSD の古い cbreak モードとローモードとを対応させる関数を与えた。さらに、端末のウィンドウサイズを取得したり変更する方法を述べた。第 17 章と第 18 章では、端末入出力の例をさらに示す。

11.15 演習問題

- 11.1 tty_raw を呼び出し(端末モードを再設定せずに)終了するプログラムを作成せよ。読者のシステムで(SVR4 と 4.3+BSD にある) reset(1) コマンドが使える場合、端末モードを復旧するためにこのコマンドを使用してみよ。
- 11.2 c_cflag フィールドの PARODD フラグで、偶数パリティか奇数パリティかを指定できる。しかし、BSD の tip プログラムでは、パリティビットに 0 や 1 を指定できる。どのようにしてこれを行っているのか?
- 11.3 読者のシステムの stty(1) コマンドで MIN と TIME の値を表示できる場合には、つぎの実験を行え。システムに 2 回ログインし、一方のログイン側で vi エディタを起動する。別のログイン側で vi が(端末を非カノニカルモードにするため)設定した MIN と TIME の値を stty コマンドで調べる。
- 11.4 端末インタフェースの回線速度(line speed)が速くなるにつれ(今日では 19,200 や 38,400 が一般的になりつつある)、ハードウェアによるフロー制御の必要性が重要になってきている。これには、XON と XOFF 文字の代わりに、RS-232 の RTS (送信要求) と CTS (送信可) が必要である。ハードウェアフロー制御は POSIX.1 では規定されていない。SVR4 と 4.3+BSD において、ハードウェアフロー制御を有効あるいは無効にするにはどうすればよいのか?