

FDclone に見る UNIX 技術

しらい たかし

FDclone って何？

- UNIX 汎用のテキスト端末用ユーザインタフェース
 - Linux や Solaris を始め多くの UNIX に対応
 - 別途 MS-DOS 版もある（一部機能が未実装）
- ビジュアルシェル
 - いわゆる「ランチャ」「ファイルマネージャ」
 - ファイルの操作を視覚的行なう
- フリーソフトウェア
 - GPL や BSD ライセンスとは異なる独自ライセンス
 - DFSG は満たしているらしい

FDclone の機能(1)

- シェル機能を内蔵
 - /bin/sh に依存している system(3) は使わない
- Bourne shell の上位互換
 - POSIX 準拠 (ほぼ)
- 初期設定ファイルやコマンドライン入力で利用
 - ビジュアルシェルでは賄いきれない細かな操作を補う
 - alias や補完等ユーザシェルとしての機能も充実

シェル機能に必要な技術

- fork (2) & execve (2)
 - 外部コマンドを呼出す際の UNIX 常套手段
- 端末制御（いわゆるジョブ制御）
 - tcsetpgrp (2) 又は ioctl (2) + TIOCSPPGRP を使う
 - 歴史的には 2BSD の C shell が起源
- 割り込み処理
 - SIGTSTP と SIGCONT
 - 子プロセスへのシグナル伝播

FDclone の機能(2)

- 擬似端末機能を内蔵
 - 画面分割されたスクリーン上の独立した擬似端末
 - 同時に複数の端末を操作可能
- vt100 互換の端末エミュレータ
 - スクリーンバッファは非実装
- 漢字コードの動的変換
 - 実端末側で非対応の漢字コードを使用可能
- 内蔵シェルのため擬似端末との親和性が高い
 - 擬似端末側のシェルが不定だと制御しにくい

擬似端末に必要な技術

- pty (Pseudo Teletype) マスタ/スレーブデバイス
 - スレーブ側入出力がマスタ側入出力に伝播
 - マスタ側入力からはキー入力を送出
 - マスタ側出力は端末エミュレーション処理を経て画面へ
- pty スレーブデバイスのログイン端末への割付け
 - `setsid(2)` によるセッションリーダ化
 - `ioctl(2)` + `TIOCSCTTY` による制御端末化
- プロセス間通信
 - `pipe(2)` によるマスタ/スレーブ間のコマンド送受

libutil を使った擬似端末の実装

```
#include <stdio.h>
#include <termios.h>
#include <sys/types.h>
#include <sys/ioctl.h>
#include <sys/wait.h>
#include <pty.h>

int main(int argc, char *argv[])
{
    struct termios t;
    struct winsize w;
    pid_t pid;
    int fd;
    u_char c;

    tcgetattr(0, &t);
    ioctl(0, TIOCGWINSZ, &w);
    w.ws_col -= 10;

    pid = forkpty(&fd, NULL, &t, &w);
```

```
    if (!pid) {
        execv("/usr/bin/less", argv);
        exit(1);
    }

    pid = fork();
    if (!pid) {
        while (read(fd, &c, 1) > 0)
            write(1, &c, 1);
        exit(0);
    }

    sleep(3);
    write(fd, " ", 1);
    sleep(3);
    write(fd, "q", 1);
    close(fd);
    while (wait(NULL) != pid);

    exit(0);
}
```

- 但し FDclone では可搬性のため libutil は未使用

FDclone の機能(3)

- 日本語 IME を内蔵
 - 標準で単漢字辞書が付属
 - pubdic 形式の辞書に対応
- 単文節変換は出来るが連文節変換は非対応
 - 文節区切りは構文解析もインタフェースも重荷になる
- 自身の入力のみならず擬似端末入力でも対応
 - FDclone 単体で日本語入力によるファイル編集可能
 - 組込み用途等 Wnn/Canna が重た過ぎる環境に便利

日本語 IME に必要な技術

- ローマ字入力と辞書検索
 - 特殊な技術を必要としない力技
- 複数の文字コードへの対応
 - IME への入力文字コード
 - IME からの出力文字コード
 - 内部処理用文字コード
- 漢字の文字コードに関する基礎知識
 - 意外と知られていない穴が多い
 - UNICODE は最悪

UNICODE の何がいけないのか

- 全く新規のコード体系
 - 従来のコード体系との間の変換にはテーブルが必要
- 字体 (glyph) に対する定義のみ
 - 見た目の似ている文字に対するコード割当ての問題
 - 既存文字コード体系とのマッピングはベンダ依存
- 拡張ルールの複雑さ
 - 異体字タグ (variant tag) → 漢字圏への無理解
 - 正規化形式 (Normalization Forms) → Mac OS X

総括

- 元々は初心者向けだったが現在はマニアック
 - UNIX 技術の集大成のような肥大化ツール
 - 作者の独善ばかりでユーザ無視？
- 使用している技術は結構高度で複雑
 - ノービスプログラマから抜け出したい人向けのお手本
 - 教科書としてはコメントも無く汚いソースコード
- 今後も作者の趣味で開発は続きます
 - どうぞ暖かく見守ってやって下さい :-)

関連 URL

- FDclone なページ
<http://hp.vector.co.jp/authors/VA012337/soft/fd/>
- FDclone ML のご案内
<http://www.unixusers.net/fdclone/>
- ダウンロード
<ftp://ftp.unixusers.net/src/fdclone/>