

# ttyrec: 端末を録画再生するツール

高林哲 鵜飼文敏 大和正武

Linux Conference 2001 投稿論文 (フルペーパー)  
<http://namazu.org/~satoru/pub/lc2001/> [PDF]

## はじめに

UNIXの世界では伝統的に文字ベースの端末および端末エミュレータ (以下、端末に統一) が広く用いられている。古くは VT100 をはじめとする専用の端末、グラフィカルなウィンドウシステムが普及した現在では xterm や kterm をはじめとする端末など、時代を問わず端末は UNIXには欠かせない存在となっている。UNIX が得意とするコマンドライン指向のテキスト処理との相性のよさが、これほど端末が広く用いられている理由といえる。本論文では端末を録画再生するツール ttyrec を紹介し、その応用および関連するソフトウェアについて述べる。

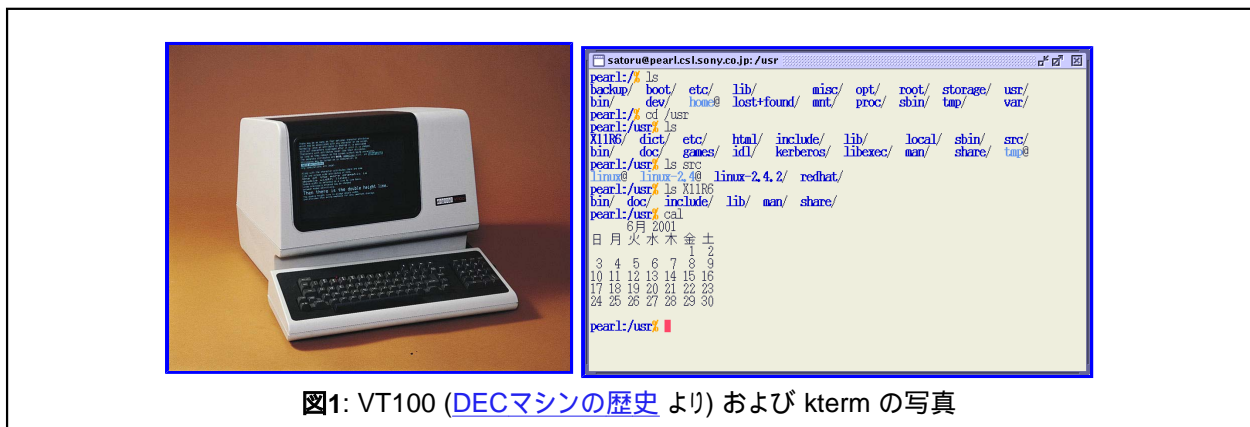


図1: VT100 (DECマシンの歴史 より) および kterm の写真

## ttyrec

ttyrec は端末を録画再生するツールである。端末上の行動は emacs -nw でも vi でも w3m でも、端末上で動くプログラムなら何でも録画できる。録画したデータは付属の ttyplay コマンドで再生が可能である。ttyrec は [ttyrec のWebサイト](#) からソースコードが入手可能なほか、次の各種パッケージが用意されている。

- [Debian パッケージ](#)
- [Red Hat Linux用のRPM パッケージ](#)
- [Vine Linux用のRPMパッケージ](#)
- [FreeBSDのports](#)
- [OpenBSDのports](#)

## ttyrecの開発

ttyrec は script コマンドにマイクロ秒単位で時刻情報 を埋め込む機能を追加しただけの単純な

コマンドである。再生用のttyplay コマンドはファイルに記録された文字列と時刻情報を元に、録画データを再生する。script コマンドの man を読むと、HISTORYの節から BSD由来のコマンドであることがわかる。

```
HISTORY
  The script command appeared in 3.0BSD.
```

ttyrec を開発するにあたって、高林は Red Hat Linux で使われている script コマンドのソースコードを入手することから始めた。rpm コマンドを使って次のように実行すると、script コマンドを含む RPM パッケージがわかる。

```
% rpm -qf /usr/bin/script
util-linux-2.10s-12
```

パッケージ名がわかれば、ソースコードパッケージ (SRPM) から目的のソースコードが得られる。なお、util-linux の最新版は <http://www.kernel.org/pub/linux/utils/util-linux/> から入手できる。

ソースコードが入手できれば、後は、時刻情報を埋め込むように script コマンドに手を入れて ttyrec を作り、再生用の ttyplay コマンドを新たに作るだけである。開発はものの半日で完了した。

## ttyrecの使い方

ttyrec と ttyplay コマンドの使い方を紹介する。

### 録画

```
% ttyrec
(シェルが立ち上がるので、自由に作業してから exit)
```

### 再生

```
% ttyplay ttyrecord
```

オプションを指定すると、倍速再生 (-s2) や、他人の録画データをリアルタイムに覗き見する (-p) ことができる。このように極めて単純なソフトウェアであるが、ちょっとした工夫でおもしろい応用ができる。たとえば、端末だけですべての作業をこなしている [旧世代](#) のUNIXハッカーなら、ログインシェルの設定ファイルに細工をして端末上のすべての行動を ttyrec で記録するだけで、そのハッカーの生涯を記録に収めることができるだろう。設定例:

```
if test -z $TTYRECLLOG; then
  export TTYRECLLOG=$HOME/var/log/ttyrec/`date +"%Y-%m-%d"`-$$
  ttyrec -a $TTYRECLLOG
  exit
fi
```

## ttyrecを用いたファイル転送

ttyrec には端末を介してファイル転送を行うという、一風変わった機能がある。たとえば、

```
サイトA(ローカル)   サイトA(出島)   サイトB(出島)   サイトB(ローカル)
```

のようにログインしているときに、「サイトB(ローカル)」のファイルを手元の「サイトA(ローカル)」に転送するのは大変である。そこで、`ttyrec` を `-u` オプションつきで動かしておく、このような状況でのファイル転送が簡単に行える。

仕組みは簡単である。「サイトB(ローカル)」では転送したいファイルを `uuencode` して端末に垂れ流す。すると、「サイトA(ローカル)」で動いている `ttyrec` が `begin 664 ...` という `uuencode` のヘッダを見つけて、手元にファイルを作成する。このように、ファイル転送は端末への出力を介して行われる。

`ttyrec` によるファイル転送は、[山下](#) が考案したアイデアに基づいている。氏は、ファイアウォールの内側の計算機からファイルを転送しようとした際に困難に直面し、端末上で見えているファイルが簡単に手元に転送できないのはおかしい、と強い不満を感じて `ttyrec` によるファイル転送というアイデアを産み出した。

これは名案であるとして `ttyrec -u` をすぐさま実現して Web上に公開すると、端末をファイル転送のメディアとして用いる手法は古くから存在することがわかった。たとえば、パソコン通信の通信文を常に監視して、メッセージに含まれる `ish` フォーマットのテキストを裏で検出して復号するという `rish` というツールがある。これは、`ttyrec -u` が行っている処理とほぼ同様である。つまり、`ttyrec` によるファイル転送は斬新なアイデアでも何でもなく、大昔からある手法の一種に過ぎなかったのである。

## ttyrecの問題点

`ttyrec` の録画データには、端末の大きさに関する情報が含まれないという問題がある。再生は録画したときと同じ大きさの端末で行う必要があるが、録画したときの端末の大きさは録画データから知ることができない。そこで、後述する `tty mania` では、端末の大きさを80桁×24行に限定して、録画データを配布している。

録画データに端末の大きさに関する情報を残すことも考えたが、録画中に端末の大きさが変わったときや、再生時に端末の大きさを変更する処理が複雑になると想定されたため、実現は見送った。

なお、`ttyrec`は端末への出力のみを記録するため、`telnet`などで入力したパスワードは端末上に出力されないかぎり `ttyrec` の録画データに残ることはない。

## tty mania

`ttyrec` は元々、端末上の行動記録を後から検索してソフトウェアの使用方法などの備忘録として利用しよう、という発想の元に高林によって作られたが、Web上に公開して他の人からの反応を得ると、他の使い道の方がおもしろいということがわかった。

大和は `ttyrec` をソフトウェアのプロモーションビデオに使えるのではないかと思いつき、[土屋](#) が開発した [emacs-w3m](#) という Emacs上の Webブラウザ向けのプロモーションビデオを作成し、[不可解なアジテーション](#)とともに、その宣伝活動を行った。次にその檄文の一部を抜粋する。

最近特に開発が盛んな`emacs-w3m`が動作している様を撮影し、Emacs Video計画の第一段として、世に問うことにした。すでにS式中毒に`cons`されている同志諸君においては、このVideoに対する忌憚のない意見を聞かせて欲しい。またおのおの、育て上げてきた自慢のライブラリの動作や`share/lisp`を漁って身に付けた秘密の技術を撮影し、広く公表することにより、この計画に参入してくれることを希望する。すべてのソフトウェアの設定ファイルのフォーマットがS式であるという素晴らしい世界を実現しよう

ではないか。

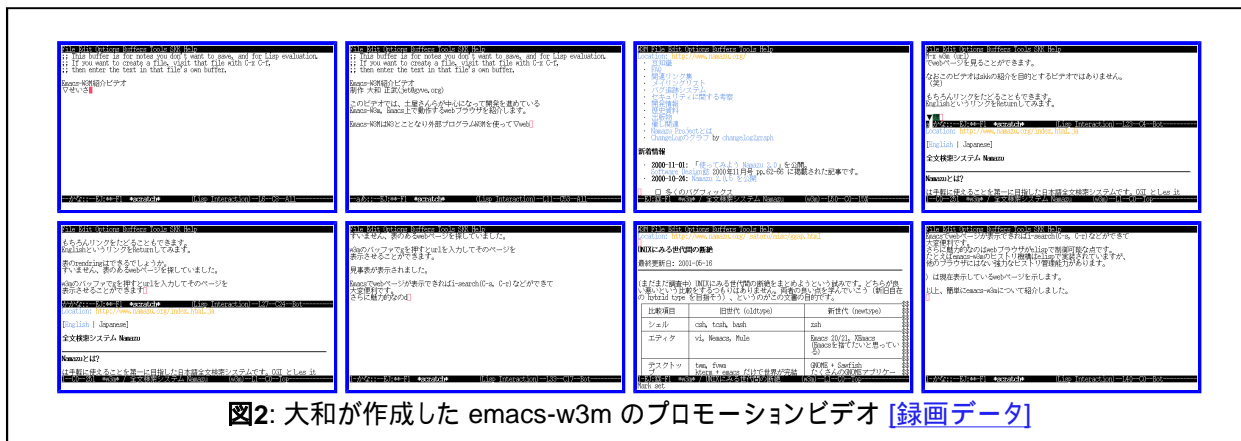


図2: 大和が作成した emacs-w3m のプロモーションビデオ [録画データ]

tty mania は、このようにして各方面に飛び火したプロモーションビデオ作成熟が高じて出来上がった Web サイトである。大和のビデオをはじめとして、各種のプロモーションビデオが公開されている。tty mania では単に ttyprec で録画したデータをファイルとして置くだけではなく、ttyplayサーバによる動画配信サービスの提供も行っている。ttyplayサーバは、クライアント側に telnet コマンドだけしか要求しないという点で優れている。つまり、クライアント側には ttyprec をインストールする必要がない。たとえば、大和のビデオを見るためには、端末から telnet コマンドを使って次のように実行するだけでよい。

```
% telnet tty.namazu.org 12345
```

## ttyplay サーバ

ttyplayサーバの実現は北目によって考案された。といっても、単に /etc/inetd.conf に次のような設定を加えるだけである。

```
12345 stream tcp nowait root ttyplay /somewhere/emacs-w3m.tty
```

さらに、ttyplay に -p オプションを指定すると、リアルタイムのストリーミングが行えることがわかった。この仕組みにより、自分が行っている作業を遠隔地の人間にリアルタイムで伝えることができる。ごく狭い仲間内で行われたストリーミング実験では、特に意味のあることをしたわけでもないのに、IRC によるチャットと連動して大いに盛り上がった。

このように、新たなコードを何も書かずに、既存のツール (inetd) と組み合わせるだけで、ttyplayサーバが実現できたのは UNIX ならではといえよう。ポート番号でコンテンツを切り替えるという素朴なサーバではあるが、とにかく動くことは動く。労少なくして功多しである。

## tty mania のコンテンツ

tty mania に登録されているコンテンツの中でも最もインパクトがあるのは小松の体操である。これは、PCに接続したビデオカメラで撮影した小松が踊っている映像を aatv でアスキーアートに変換したものである。実際に ttyplay で再生すると、非常になめらかに動く。

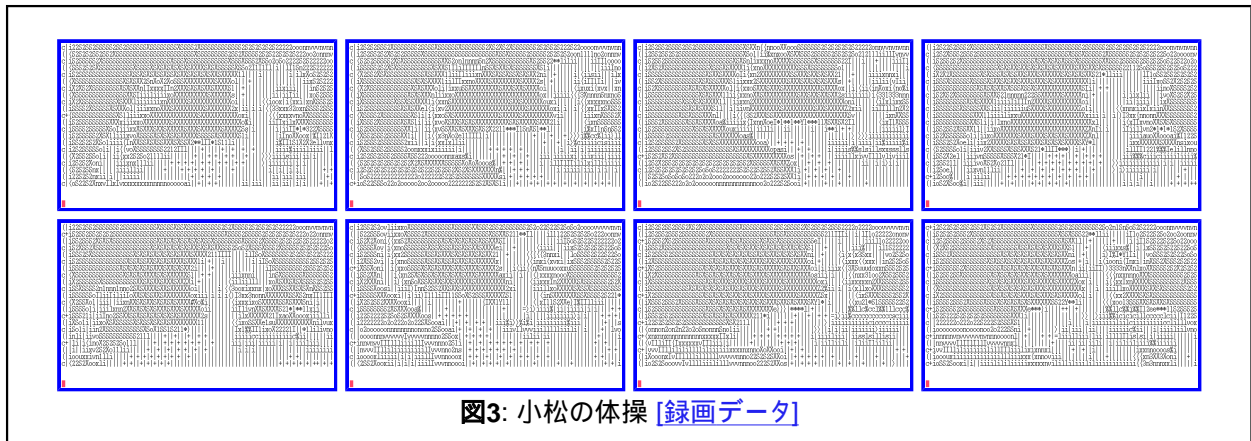


図3: 小松の体操 [録画データ]

一方、旧世代UNIXハッカーの面目躍如たる森本は xterm のTektronix互換モードを生かし、グラフィック表示を ttyrec で録画することに成功した。当初、ttyrec は文字ベースの情報しか扱えないと思われていたが、Tektronix互換モードを活用すれことによりグラフィック表示の録画再生という新たな可能性が生まれた。温故知新である。

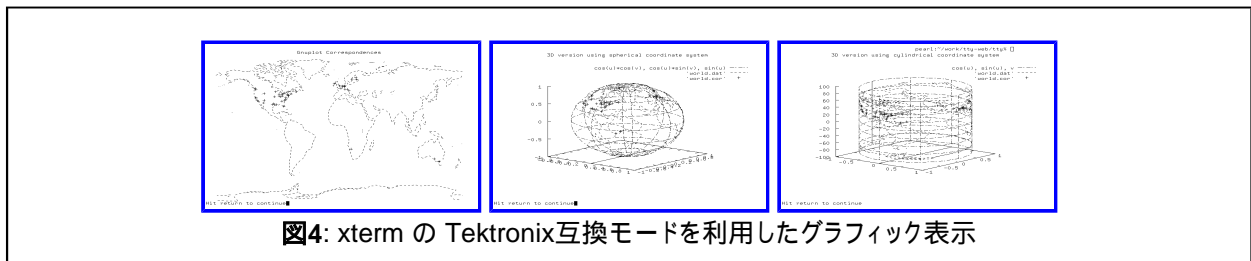


図4: xterm の Tektronix互換モードを利用したグラフィック表示

tty mania は運営が開始されるとともに、すぐさま Web日記業界 (?) の各方面で話題にのぼり、tty.namazu.org のトップページへの初日のアクセスは約 800件に達した。Web日記業界における情報伝達のすばやさには目を見張るものがある。

ところで、Kevin On Demand! では、Kevin Mitonick がTutomu Shimomura の計算機に不正侵入する様子を、tty maniaと同様に telnet コマンドだけで見る ことができる。通常の端末の出力に加えて経過時間を表示するなど、非常によくできているが、残念ながら Kevin On Demand! を実現するソフトウェアは公開されていない。Webサイトの説明によると、この録画データはtcpdump の改良版の出力から再構成したそうである。

## 関連ソフトウェア

本節では ttyrec に関連するソフトウェアを紹介する。

### Joey Hess の Debianパッチ

ttyrec が公開される 4 か月ほど前に、Joey Hessが同様のツールを Debian向けに開発していたことが ttyrec の公開後に判明した。アプローチはほぼ同様だが、Hessの作には次のような特長がある。

- 新たなコマンドは作らずに script コマンドへの拡張という形で仕上げている。
- 文字列を記録するファイルと時刻情報を記録するファイルを分けている。(2つのファイルを出力する)
- 再生時に遅延が起きないように一工夫している。
- 再生プログラムは Perl で書かれている。

北目が `ttyrec` の Debian パッケージの作成を提案すると、Hess は、自分のパッチを `script` コマンドに適用する方がいいし、本家の `script` に修正を反映させるといい、と主張した。確かに一利あるが、我々は、今さら本家の `script` に修正を反映させてそれを普及させるのは難しいだろうと考えている。それに、`ttyrec` を公開してから Hess のパッチの存在に気づいたというのは手遅れである。かくして車輪は再発明され続ける。

## ttyplayd

北目によって考案された `ttyplay` サーバは、`inetd` を利用しているために各録画データに対して、`inetd` のエントリを追加する必要があった。これは、新たな録画データの追加にはルート権限が必要であることを意味する。また録画データそれぞれに対して異なるポート番号を割りあてる必要がある。そこで、鵜飼は `inetd` を使わない `ttyplay` サーバの実装 `ttyplayd` を考案した。`ttyplay` サーバがやるべきことは、TCP コネクションを確立し、そのコネクションに `ttyplay` の出力を送ることである。単一ポートで複数の録画データを扱えるようにするためには、接続後にプロンプトを出してユーザに再生したいデータを選択させればよい。Perl や [Ruby](#) といったスクリプト言語を用いれば、このような処理は簡単に記述できる。特権ポートさえ使わなければ、ルート権限を必要とせず一般ユーザ権限でもサーバを走らせることができる。

ところで、最近の `telnet` クライアントでは、わざわざユーザ名を入力しなくてもデフォルトで現在と同じユーザ名でログインセッションを開始することができる。これは `TELNET` プロトコルの [TELNET Environment Option](#) を使って、`USER` 環境変数をクライアントからサーバに送ることによって実現されている。また、`-l` オプションで別のユーザ名を指定することもできる。`ttyplayd` ではこの機構を用いて、再生するデータの選択を可能にした。たとえば、大和の `emacs-w3m` のビデオを再生するには次のように実行すればよい。

```
% telnet -l emacs-w3m tty.namazu.org 10023
または
% browser telnet://emacs-w3m@tty.namazu.org:10023/
```

[URL](#) 表記の場合は、ユーザ名のフィールドを利用して録画データを指定する。ただし、`telnet` URL をサポートしたブラウザは少なく、サポートされていても、適切なクライアントが起動するとは限らないため、誰でも簡単に使える状況とはいえない。

`ttyplayd` は、`ttyplay` サーバだけでなく Web サーバとしての機能も有している。我々はこれをハイブリッドサーバと呼んでいる。`ttyrec` の録画データが保存されているディレクトリを指定すれば、その一覧を Web ページとして提供できる。当然ながら、そのページには `ttyplay` サーバへのリンクを表示する。また録画データ自体を `application/x-ttyrec` という MIME type でダウンロード可能である。従ってブラウザ側で、この MIME type に対して `ttyplay` を実行するように設定しておけば自動的に再生が行われる。以上のような機能をもつ `ttyplayd` を用いると `tty mania` の維持管理コストを削減できる。

## M-x play-function

大和は `ttyrec` を Emacs のヘルプ機能の一部として利用することを考えた。M-x `describe-function`、M-x `describe-mode`、M-x `info` など Emacs のヘルプは充実している。しかし現状のヘルプ機能で得られるのは文章化された情報だけであり、それを読んで理解する必要がある。特に Emacs を使い始めた初心者にとって、文章を読んで機能を理解するのは負担が大きい。

そこで、M-x `play-function` は Emacs の関数を動画によって説明するための枠組みを提供する。ヘルプ動画の提供者は、説明したい関数の動作を Emacs 上で実演し、それを `ttyrec` を用いて録画する。そして、ユーザが M-x `play-function` に続いて関数名を入力すると、Emacs は新しいフレームを開いてヘルプ動画を再生する。

ttyrec を用いると、このような単純な仕組みで動画ヘルプが実現できる。ヘルプ動画が端末の表現力に限定されるという欠点があるものの、Emacsの基本的な編集操作は十分にカバーできると我々は考えている。たとえば、矩形を切り取って空白で埋め尽くすclear-rectangle は言葉で説明されても理解しづらいが、play-function で動画として説明されると一目瞭然である。

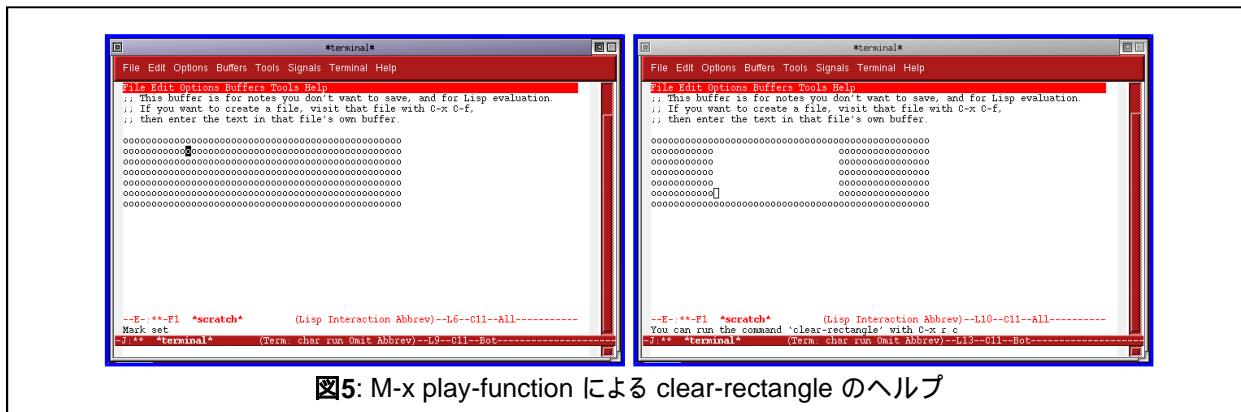


図5: M-x play-function による clear-rectangle のヘルプ

## gttyplay

gttyplayは ttyplay コマンドに [GTK+](#) の GUIを付加した再生ツールである。gttyplay は、再生の一時停止、再生速度の動的な変更、プログレスバーによる経過表示といった機能を提供する。再生の表示は kterm などの端末の中で行われる。



図6: gttyplay

## Ruby/TtyPlay

[Ruby/TtyPlay](#) は ttyplay と同様の処理を行う Ruby用のライブラリである。Ruby/TtyPlay を用いると、Rubyから手軽に ttyrec の録画データを扱うことができる。実際、[ttyplayd](#) では、ttyplayコマンドを呼び出す代わりに、Ruby/TtyPlay の TtyPlayer オブジェクトを用いて録画データの配信を行っている。

## x11rec

ttyrecによって端末の録画が実現されると、今度は任意の X上のアプリケーションを録画するツールが欲しくなった。調べると、GTK+ のアプリケーションの操作を記録・再生する [Gerd](#) というツールが見つかったが、対象が GTK+に限定されることと、出力ではなく操作を記録するために再生時に実際に GTK+アプリケーションが動かされるという点が不便である。そこで、高林は X 上のアプリケーションを録画するツール[x11rec](#)を開発した。これは、xwd を連続的に実行して、最後に [ImageMagick](#) と [Gifsicle](#) を用いて [MNG](#)またはアニメーションGIFのフォーマットで動画ファイルを作成するという素朴な代物である。しかし、このような [富豪的](#) なやり方であるにも関わらず、十分まともに動く。実際、Pentium III 700 MHz の計算機では [gnomine](#) を 14.5 fps で録画することができた。

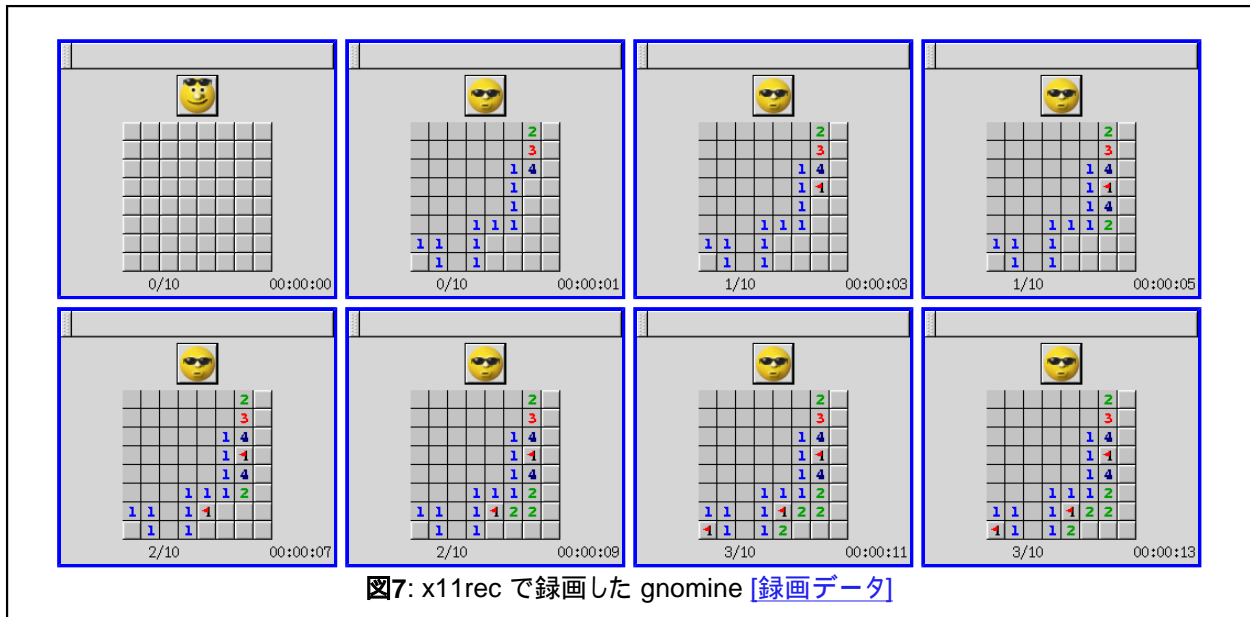


図7: x11rec で録画した gnomine [録画データ](#)

x11rec は極めて素朴な手法によって実現されているため、マウスカーソルをキャプチャできない、録画のフレームレートが正確ではない、遅い計算機では使い物にならない、などの多くの欠点がある。一方、[林](#)の [vncrec](#) は [vnc](#) の描画プロトコルを元に録画を行うため、スクリーンを連続的にビットマップ画像としてキャプチャするのではなく、一連の描画イベントを記録する。vncを経由しているため、マウスカーソルのキャプチャも可能である。さらに、x11rec が完成してしばらくすると、[Rasca](#) による [XVidCap](#) が見つかった。これはスクリーンを動画としてキャプチャする GUI つきのツールである。

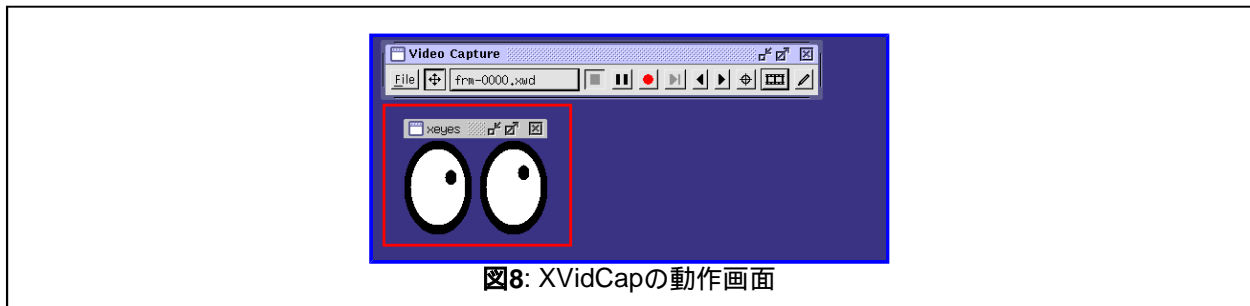


図8: XVidCapの動作画面

vncrec や XVidCap と比べると x11rec はずいぶん素朴なツールである。実際のところ、x11rec の利点は、Ruby でたった 100行程度で実現されているという単純さくらいしかない。

ところで、XVidCap も内部的に xwd と同様に Xサーバから特定の領域のビットマップ画像を得るため、マウスカーソルのキャプチャは行えない。通常の Xサーバから直接マウスカーソルをキャプチャすることは難しいようだが、[田岡](#) は、[XFree86](#) の DGA (Direct Graphic Access) の拡張機能を利用してマウスポインタ込みでスクリーンをキャプチャするツール [xdggrab](#) を公開している。

なお、マルチメディアに強い UNIX である [IRIX](#) には スクリーンやビデオ入力などを動画および静止画としてキャプチャする MediaRecorder という高機能なツールが付属している。MediaRecorderの完成度は高く、マウスカーソルのキャプチャが可能である。また、Windows にはスクリーンを録画再生するツールとして [ScreenCam](#) や [SnagIt](#) がある。

## Expect

[Don Libes](#) による [Expect](#) は、ftp や telnet などの対話的なソフトウェアの操作を自動化するツールである。Expect を用いると、ユーザ名やパスワードの入力が自動で行えるため、シェルスクリプト

や Makefile の中から対話的なソフトウェアを一括に処理できる。

## screen

GNU Project の [screen](#) は VT100/ANSI 端末をエミュレートする端末マネージャである。screen を用いると、1つの端末で複数の仮想端末を切り替えて使うことができる。screen は、端末の出力のログを記録したり、その時点のハードコピーを記録することができる。さらに、端末に出力されたものをコピーアンドペーストすることもできる。仮想端末を管理している部分と端末をエミュレートしているフロントエンドは分かれており、任意の時点でそれを切り離したり、接続し直すことも可能である。したがって、作業中にネットワークが不意に切断されても、screen で作業していればあとでscreen のセッションに再接続すれば、作業をそのまま再開できる。さらに複数のフロントエンドで1つのscreenセッションを共有することもできる。これは、2人のユーザが1つの仮想端末を共有して作業を行うのに利用できる。下の図は画面を上下に分割してそれぞれのウィンドウに異なる仮想端末を割り当てている様子を示す。

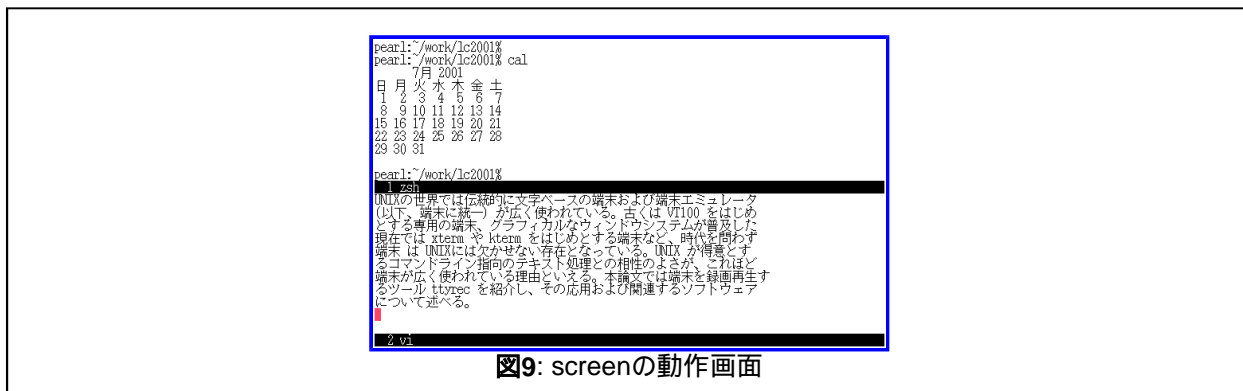


図9: screenの動作画面

## おわりに

ttyrec はソフトウェア単体としてみるとたわいもない代物であるし、実際にも大したことはできないのであるが、どういうわけか「これを使えばいろいろなことができそうだ」という幻想を物好きな人たちに与える効果を持っていた。半日で作ったようなソフトウェアで多くの人を楽しませることができたのは嬉しいかぎりである (とはいっても狭い世界での話だが)。

素朴なアイデアと単純な実装が取り柄の ttyrec であるが、そのコンセプトを発展させて ttyrec を内蔵した端末があればおもしろいのではないかと考えている。これは、GUIのスライダーを使って自由自在に過去に遡ったり、過去の行動履歴を動的に検索したりできるという端末である。計算機資源が加速度的に豊かになってゆくに連れて、あらゆる履歴を豪華的に記録する [タイムマシン式](#) のアプローチはごく自然なものになりつつある。

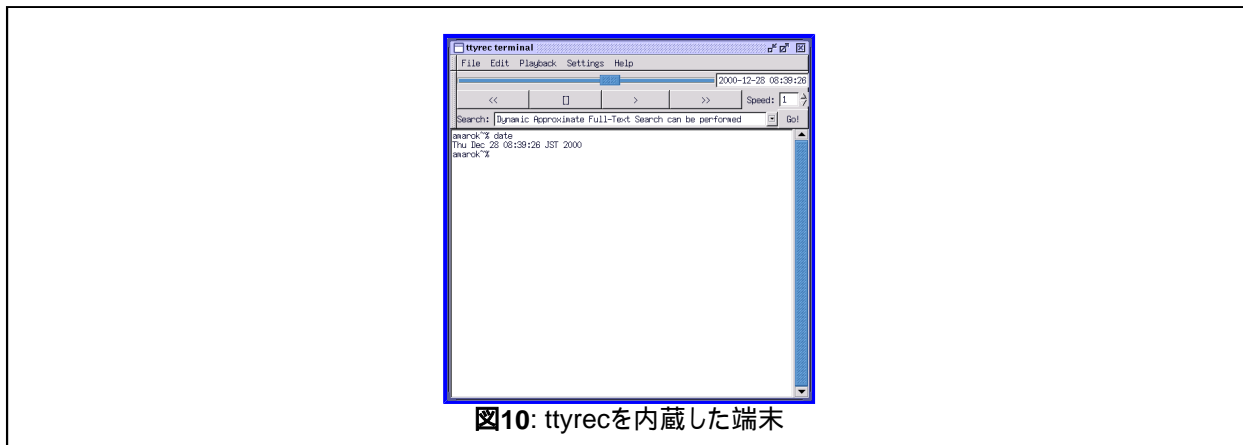


図10: ttyrecを内蔵した端末

ところで、ttyrec を不正侵入者を追跡する用途に使えるのではないか、という指摘を海外からもらったが、逆に、悪意を持った人間によって悪用される危険性も考えられる。我々は平和的な利用法を期待するものであり、ttyrec が悪用されないことを願っている。

## 参考文献

---

参考文献は本文中にハイパーリンクとして埋め込んだ。Web版は <http://namazu.org/~satoru/pub/lc2001/> から閲覧可能である。なお、本文中は敬称略とした。

## 著者紹介

---

### 高林哲

ソニーコンピュータサイエンス研究所。奈良先端科学技術大学院大学博士後期課程。大学時に [全文検索システムNamazu](#) を開発して以来、フリーソフトウェアの世界に興味を持つ。最近 [UNIXにみる世代間の断絶](#) と題して、旧世代と新世代のUNIXハッカーの違いを分析しようと試みたが、一般の人から見ればどちらも同じではないかと気づき、どうでもよくなりつつある。

### 鵜飼文敏

日本ヒューレット・パッカード株式会社。 [Debian Project](#) オフィシャルメンバー、 [Debian JP Project](#) リーダー、日本Linux協会会長。大学院在籍中に386BSD やLinuxをPC98アーキテクチャで動かして以来、フリーなオペレーティングシステムの世界にはまる。Debian JP Project創設時のメンバで以後Debianを中心に活動。 [debian.or.jp](#) および [linux.or.jp](#) などの運用管理を行なっている。

### 大和正武

奈良先端科学技術大学院大学博士後期課程。 [gyve](#), [gtkDPS](#), [dgs](#), [gs-cjk](#) の開発に参加。フリーソフトウェアの開発は情操教育の役に立つような気がしている。

---

*satoru@namazu.org, ukai@debian.or.jp, jet@gyve.org*