

ペンコンピュータの Unix

大木敦雄¹、久野 靖¹、角田博保²、粕川正充³
¹筑波大学、²電気通信大学、³お茶の水女子大学

キーボードを有しないペンコンピュータでも、従来の Unix 環境をそのまま利用できることを示す。T-cube や UniStroke 等のペン入力方式でキーボードと等価な入力機能をカーネルレベルや X クライアントとして実現すれば、キーボードベースのインタフェースを何ら変更することなく利用できる。

Unix on pen computers

We show that usual Unix environment is usable on pen computers which have no keyboard. Using stylus input schemes like T-cube or UniStroke, we have implemented functionalities equivalent to keyboards into kernel as well as X clients. These functionalities make it possible to use keyboard-based user interface without any modifications.

Atsuo Ohki¹, Yasushi Kuno¹, Hiroyasu Kakuda², Masaatsu Kasukawa³

¹University of Tsukuba,

²University of Electro-Communication,

³Ochanomizu University

1 はじめに

現在、携帯可能なコンピュータの種類は、重量 1~3Kg の A4 サイズノートパソコン、より軽量の B5 サイズパソコン、パームトップコンピュータ、ザウルスなどに代表される PDA などさまざまである。これらの表示装置は、表示面の大きさ、カラー/モノクロ、解像度の差こそあれ、液晶表示が一般的である。一方、入力装置は、

- トラックボールなどの指示装置とキーボード
- 指示装置を兼ねたペン

の 2 種類が一般的である。

筆者等は後者の場合において、Unix がどの程度使えるか試すために、

- コンソール (VGA 画面) モード
 - 画面最下行の英数字入力行 (+画面再利用)
 - T-cube (+画面再利用)
- X Window System
 - 仮想キーボード (xkeycaps の修正版)
 - T-cube
 - UniStroke

の合計 5 種類について、ペン入力機能を実装し、簡単な評価を行った。

使用したペンコンピュータは、主に、三菱電機 (株) 製 Amity/sp (A5 サイズ、重量 850g)¹。

¹他には、同社製 Amity/vc、(株)ワコム製 PenTop PL-200V 等。

表示装置は、タブレットを兼ねた透過型 DSTN カラー液晶ディスプレイ (VGA 相当、80 文字 × 25 行、640 × 480 ドット、7.5 インチ) である。指示装置は専用のペンであり、液晶表示面から約 1cm 以内にペン先を近付けると位置検出が可能になり、

- ペン先の xy 座標
- ペン先が表示面に触れているか否か
- (ペン軸にある) サイドスイッチが押されているか否か

を表す固定長のバケットが (本体内部で接続された) 専用のシリアルポートに連続して出力される。したがって、ソフト的には専用シリアルポートに接続された (絶対位置を返す) 2 ボタンマウスとして扱うことができる。

ソフトウェアは、FreeBSD[1] + PAO[2] と XFree86 3.1.2[3] を変更して使用した。

2 コンソール版ペン入力機能

つぎの 2 つを実装した。

- 入力行
コンソール画面の最下行に英数字 / 特殊記号等を表示しておき、入力文字をペンで拾う。
- T-cube[5]
文字ベースのバイメニューから、フリック動作で文字入力を行う。(バイメニューはコンソール画面上の任意の位置へ移動可能)

ペンは X の指示装置としても使用するため、X サーバの動作の有無を確実に判定する必要がある。このため、上記 2 つのペン入力機能はカーネルレベルで実現した。つまり、

- シリアルポート モジュール
コンソール画面が文字表示時には、バケットをコンソール画面 / キーボード モジュールに迂回する機能を追加。
- コンソール画面 / キーボード モジュール
バケットの処理機能とともに、入力行 / T-cube のいずれかの処理機能を追加。

また、カーネル内の画面表示バッファを直接操作できるので、画面に表示されている文字 / 単語をペンで拾って入力することも可能となった。すなわち、通常の文字表示部分にペンを当てると、その部分より右側に表示されている特殊文字や単語の一部を拾って、入力とする。これにより、(ls コマンドや bash 等の補完機能で) 画面に表示したファイル名を容易に拾うことができる。頻繁に使用するコマンド名を画面に表示しておけば、コマンド名も容易に入力できる。

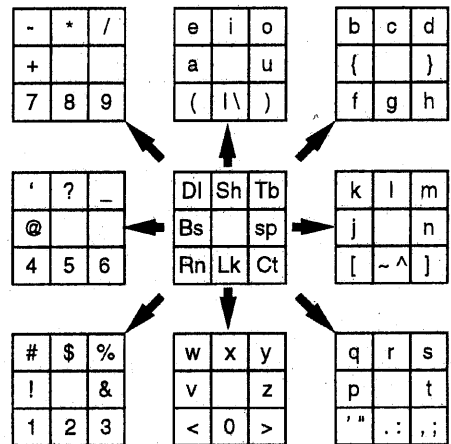


図 1: T-cube の文字の割り当て

T-cube では 72 種類の入力を区別可能であるが、すべての ascii 文字を割り当てるには足りない。そこで、キーボードと同様にシフトを用いる。図 1 に、T-cube の各メニューの文字割り当てを示す。英大文字は「シフト + 英小文字」、2 つ割り当てがあるものはシフトの有無で左あるいは右の文字を選択する。これで、標準的な英字キーボードにある文字はすべて入力可能である。

コンソール版では、図 1 の中央のメインメニューを 6 文字 × 3 行で画面に表示する。このメインメニューは画面上の任意の位置にいつでも文字単位で移動可能である。

メインメニューの中央にペンを当てたまま、8 方向のいずれかにペンを移動し、ペンを表示

面から離すと、対応する文字を入力できる²。メインメニューの中央以外の部分にペンを当てると、対応するサブメニューを選択する。そのまま8方向のいずれかにペンを移動してから離すと、当該サブメニューの対応する文字を入力できる。

3 X Window System 版ペン入力機能

まず、ペンをXの指示装置として使用可能なようにXサーバを修正した。Xクライアントにとっては、ペンはマウスなどと同様のボタンを持つ指示装置として認識され、

- 左ボタンを押す → ペン先を表示面に触れる
- 右ボタンを押す → ペン軸のサイドスイッチを押す
- 中ボタンを押す → 両者を同時に行う

という形で3つのボタン操作を区別できる。

X Window System 版では、T-cube[5]方式とUniStroke[6]方式の2種類のペン入力を実現した。どちらも、Xのクライアントとして動作し、それぞれ、図2と図3に示すクライアントウィンドウをしている³。いずれの場合も、クライアントウィンドウ内でのペン操作から入力文字を認識する。認識した文字は、各修飾キーの状態を加味してスキャンコードに変換し、(システムコールを用いて)カーネルレベルのスキャンコードのキューへ押し込む。Xサーバは、このスキャンコードのキューからキーボード入力を読み取るため、Xサーバからすればキーボードからのキー入力と全く等価であり、Xサーバ配下の他のクライアントに対しても完全に透過である。ただし、T-cube / UniStrokeのクライアントウィンドウ内でペンを用いるため、入力先となるクライアントに予め入力フォーカスを設定しておく必要がある。つまり、ポインタフォーカス方式は使用できない。

²ただし、Sh、Lk、Ctの3つは実際の文字ではなく、シフトキー、シフトロックキー、コントロールキーに対応する。

³114 × 114 ドット。実寸は 27mm × 27mm

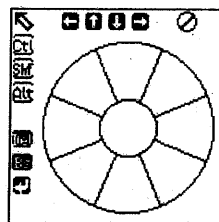


図 2: T-cube クライアントウィンドウ

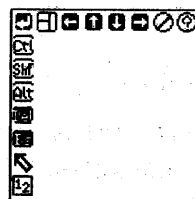


図 3: UniStroke クライアントウィンドウ (1)

3.1 共通機能

クライアントウィンドウの左辺と上辺に配置した両者に共通のボタンは、キーボード上の英数字や特殊文字キー以外のキー入力を可能にする。まず、左辺にある「Ctl」、「Sht」、「Alt」の各ボタンは(キーボードの)コントロールキー、シフトキー、ALTキーに対応し、ペンで突いてオン/オフする。これらの修飾キーの状態は、認識した入力文字のスキャンコードに付加される。

その下の3つのボタンは、DELキー、BSキー、RTNキーに対応し、同様に、ペンで突いて入力する。上辺の残りの4つの矢印ボタンは、キーボード上の4つの矢印キーに対応し、やはり、ペンで突いて入力する。

クライアントウィンドウからペンが出ると、コントロールキー、シフトキー、ALTキーのうちオンになっているものがあれば、その状態を反映したスキャンコードをキューに押し込む。これにより、『キーボードの修飾キーを押しながらのポインタ操作』が可能になる。クライアントウィンドウにペンが入ると、シフト状態を

初期化しコントロールキー、シフトキー、ALTキーもオフにする。

3.2 T-cube

コンソール版 T-cube と同等な機能に加え、ペンがメニューに接してから一定時間(約0.3秒程度)経過してもペン移動がない場合には、サブメニューを表示する。また、クライアントウィンドウ内で(ペン軸の)サイドスイッチを押すたびに、シフト状態を巡回する。これにより、Sh や Lk を予め入力しないで済む。

3.3 UniStroke

[6] では英字のみの割り当てであるが、シフト状態を追加し、未使用部分に特殊文字や数字を割り当てることで、(T-cube と同様に) 標準的な英字キーボードにある文字はすべて入力可能にした。図4に各ストロークへの文字の標準割り当てを示す⁴。

| | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|
| ↑ | a | A | | o | O | 0 | | — |
| > | b | B | 3 | p | P | | ! | ? |
| < | c | C | | q | Q | | . | : |
| ↖ | d | D | 4 | r | R | | , | ; |
| ↗ | e | E | | s | S | | [| (|
| ↖ | f | F | 5 | t | T | |] |) |
| ↗ | g | G | 8 | u | U | | ' | " |
| ↖ | h | H | 9 | v | V | | | # |
| ↗ | i | I | 1 | w | W | | ~ | + = |
| ↖ | j | J | | x | X | | < | - |
| ↗ | k | K | | y | Y | 7 | > | * % |
| ↖ | l | L | 6 | z | Z | 2 | \ | / |
| ↗ | m | M | | | | | & | @ |
| ↖ | n | N | | | | | ~ | . |

図4: UniStroke の文字の割り当て

クライアントウィンドウ内で(ペン軸の)サイドスイッチを押すたびにシフト状態を巡回し、図4の何列目の文字を選択するかを切替える。図5のように3分割の場合には、ストロークの書き始め位置により

- 左下 → 図4 第1列

⁴設定ファイルにより変更可能

- 左上 → 図4 第2列
- 右側 → 図4 第3列

のように選択できる。さらに、サイドスイッチを押しながらストロークを書くと、コントロールキーを付加する。

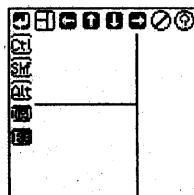


図5: UniStroke クライアントウィンドウ (2)

4 評価

4.1 定性的評価

英数字 / 特殊文字をランダムに表示するプログラムで T-cube による文字入力を練習したところ、コンソール版 / X 版のいずれでも、1時間程度の練習で文字の配置を覚え、1~2秒 / 文字程度の入力が可能であった。ただし、入力誤りが1割程あり、viによるプログラム編集には、かなりのストレスを感じた。

UniStrokeでも、同じプログラムで練習したところ、やはり、1時間程度でストロークを習得できたが、入力速度は T-cube より若干速い程度であった。入力誤りは数%程度であり、viによるプログラム編集には少々ストレスを感じるものの、muleなどのundoが充実しているエディタでは我慢できる程度であった。

T-cubeでは、メニューを突くペン位置が重要となるため、(少なくとも筆者の一人は)練習を積んでも、キーボードでのタッチタイプのようなヘッドアップ入力はできなかった。[7]やトンツウ方式⁵の T-cube も試作してみたが、

⁵最初のトンでメニューの中心位置を定め、続くツウ(フリック動作)の開始位置でサブメニューを選び、その移動方向で文字を選択

ヘッドアップ入力が可能になったものの、入力速度、入力誤り等に改善は見られなかった。

一方、UniStrokeでは、ストロークの書き始めの位置は無関係であり、練習当初からヘッドアップ入力が可能であった。

T-cube / UniStroke いずれの場合でも、キーボード入力のように指が自然に動くようには入力できない。T-cube / UniStroke の入力速度の遅さに加えて、緩りを意識して入力する必要がある。

総じて、タッチタイプを習得したユーザが、1本指でキーボード入力を行う程度の入力速度は可能と思われる。

4.2 入力速度

一定時間内にどの程度のテキスト入力が可能かを5種類の入力方式について筆者等の一人が被験者となって実験した。実験条件は、「2分間に気づいた入力誤りを修正しつつ、英文を見ながら入力する」である。

| 入力方法 | 総文字数 | 秒 / 文字 |
|---------------|------|--------|
| キーボード | 514 | 0.2 |
| コンソール版 T-cube | 83 | 1.4 |
| コンソール版入力行 | 88 | 1.4 |
| X 版 T-cube | 84 | 1.4 |
| X 版 UniStroke | 173 | 0.7 |
| X 版仮想キーボード | 113 | 1.1 |

日常的にも使い慣れているキーボードが最も速く、これに次いで UniStroke が速い。X 版仮想キーボードよりも T-cube が遅くなっているが、これは、T-cube を使用しなくなってしばらく経ってからの実験であったためであろう。慣れているときに実験すれば、UniStroke に次ぐ速さになるとと思われる。

5 まとめ

筆者等の一人は、X 版 UniStroke で約1年間定期的に使用している。メールや Net News の読み書きが主であるが、カーネルの修正やプログラミングにも使用している。この環境は、タッチタイプ可能なキーボードほど快適ではないものの、場所を選ばずどこでも使用できる。

同じソフトウェア環境のデスクトップ機では、筆者等のほとんどは、マウスをほとんど使用せずにキーボードでの操作が主である。一方、ペンコンピュータ環境では、指示装置としてのペンを多用する操作形態になった。デスクトップではキーボードとマウスの間で手を移動するのが煩わしいが、ペンコンピュータ環境では、文字入力のためにペンを持っている手を動かすだけでよく、しかも、マウスに比べて指示が直接的である。初心者にはマウスよりペンのほうが操作しやすい可能性もある。

筆者等のほとんどは、タッチタイプ可能なキーボードが使用可能な環境では、まず間違いなくキーボードを選択する。しかし、キーボードがなくペン入力のみが利用可能な機器の場合でも、デスクトップ機と同じソフトウェア環境がそのまま使える可能性を示した。

なお、ペンの特性を生かした GUI については [4] で研究中である。

参考文献

- [1] "<http://www.freebsd.org/>"
- [2] "<http://jaz.jp.FreeBSD.org/pub/FreeBSD-jp/PAO/>"
- [3] "<http://www.xfree86.org/>"
- [4] 久野、角田、大木、粕川、『アイコン投げシェル』、第38回プログラミングシンポジウム、1997年1月
- [5] Dan Venolia, Forrest Neiberg, "T-Cube: A Fast, Self-Disclosing Pen-Based Alphabet", Human Factors in Computer Systems, ACM, April, 1994.
- [6] David Goldberg, Cate Richardson, "Touch-Typing With a Stylus", INTERCHI'93, ACM, April, 1993.
- [7] Gordon Kurtenbach, William Buxton, "The Limits Of Expert Performance Using Hierarchic Marking Menus", INTERCHI'93, ACM, April, 1993.