

解説



高機能ワークステーションについて†

鈴木 則久††

1. ま え が き

現在日本ではパソコンが飛ぶように売れている。一方海の向うのアメリカでは、IBM の進出により弱小メーカーが脱落しはじめている。しかし、昨年度の IBM のパソコンの売上げは 60 万台¹⁾、業界全体で 250 万台であるから、パソコン産業が自動車産業を抜くのは 5 年以内であろう。

マーケットがこれだけ大きくなると、高級機と低級機のマーケットが開拓されるのは、他の商品と同じである。パソコンではワークステーションという名前が高級機が売り出し始められている。しかし、ワークステーションという名前を単に販売手段として使っている場合も多く、大型計算機用の高性能端末をワークステーションと言っているメーカーもある。

この特集では高機能ワークステーションの意味をかなり狭く解釈し、解説していきたい。ここでは、高機能ワークステーションの満たす条件として、高速演算装置、ビットマップ・ディスプレイ、ローカルエリア・ネットワーク (LAN)、ポインティング・デバイス (マウス等) を少なくとも装備し、システム・ソフトウェアを完備しているものを考えている。

第 2 章ではワークステーション開発の歴史を個人的体験を踏まえて説明する。第 3 章ではワークステーションのハードウェアを説明するが、詳しくは本特集の他の解説²⁾⁻⁶⁾も参照されたい。第 4 章ではソフトウェア、第 5 章では応用分野について解説する⁷⁾⁻¹⁰⁾。

2. ワークステーション開発の歴史

マイクロエレクトロニクスとソフトウェアの進歩により、一般オフィス業務はコンピュータの助けを借りれば能率良くできる仕事が多くなっている¹¹⁾。例えば、文書の作成・編集・印刷・保存、種々の設計の道

具、電子郵便等の業務である。初期の利用形態としては、それぞれの業務ごとに違ったシステム (それぞれコンピュータを内蔵しているのだが) を使っていた。ワープロとか、CAD 用の計算システム等がこれら単能システムである。しかし、CAD を使う人もワープロを使うし、電子郵便も使うので、これら多業務を統合して一台のコンピュータで使えるようにしていくのが技術革新の趨勢である。この方向が完成すると、日常業務を行っていくのにパソコンがいつでも必要になり、ワークステーションという言葉が生まれた。実際にワークステーションという名の付いた最初の商品は、Xerox の STAR ワークステーションであったと思う¹⁰⁾。

ワークステーションは各オフィス労働者の日常業務を一切取りしきるものであるから、職種により秘書用だとか、技術者用だとか、設計者用だとかいろいろ商品化され、分野によって許容される価格も違ってくる。秘書用の場合は安くなくては売れないので、ワークステーションと言っても、パソコンに毛のはえた程度のものである。一方最近では石油探掘場所の予測に LISP 専用のワークステーション⁶⁾が使われているし、VLSI の CAD にも使われているが、これらには高価なものが使われている。

さて、このような OA の伸び、ワークステーションの伸びの引金になった技術として特に次の二つの発明をあげたい。

先日 Newsweek (1983 年 8 月) で「コンピュータ分野での日本の挑戦」という特集号を読んでいたら、アメリカ軍の研究援助機関である DARPA の業績の一つとして、高度の文書作製用ソフトウェアの開発をあげていた。OA ブーム、ワークステーション・ブームの引き金となったのはなんとと言っても、コンピュータで文書の作成・編集が高速に出来るということを考え出したことである。これは恐らく重要な発明であったが、何もキーとなる論文が出た訳でもなく、特許も取られた訳ではない。

† High-performance Workstation by Norihisa SUZUKI (Department of Mathematical Engineering, University of Tokyo).

†† 東京大学工学部計数工学科

もうひとつの重要なアイデアは、点でもって字や絵を表わせると考えついたことである。絵では印象派の画家がすでに使っていた技術であるが、字にも応用されて大変な発明となった。現在のレーザ・プリンタ、ビットマップ・ディスプレイなどはみなこの考えを利用している。

ワークステーションは1973年に Xerox のパロ・アルト研究所 (PARC) で開発された ALTO¹²⁾ が原型である。ALTO の開発の動機は、電子の力を各ユーザの手元に置くことによって文書作成であるとか、点を使って文字や図形を扱う技術が誰にも使えるようになり、オフィスに革命をもたらすだろうと考えたからであろう。筆者は1972年から1982年までの滞米中 Xerox PARC, スタンフォード大学 (Stanford) カーネギー・メロン大学 (CMU) にいて、これらの技術の開発を見ました参画していた。この経験を踏まえてワークステーション開発の歴史を解説する。

文書作製技術の開発に一番熱意を示していたのは、Stanford の McCarthy であり、彼の率いる AI 研究所は TVED というエディタ、PUB という文書コンパイラ、それに種々のフォントを印刷できる XGP というゼログラフィ・プリンタを開発していた。TVED を使用する端末は、ビデオ・ディスクを中央で共用し、そこから同軸ケーブルでつながっていた。スクリーンを分割したり、いくつものスクリーンを使用して別別のジョブを走らせたりできた。今のウィンドウ・システムの原型である。PUB¹³⁾ は、Apple で LISA のソフト開発を統括した Larry Tesler によって作られた。これは現在良く使われている文書コンパイラの Scribe¹⁴⁾ の原型であり、コマンド体系は大変似ている。XGP はかなり古い機械で故障も多かったが、1973年に Xerox から寄付されたのに、AI 研究所の所員がソフトを書いて、CMU でも MIT でも使われた。筆者は1975年に CMU に移った後でも、ARPA ネットを通して Stanford で PUB を動かし、出来たファイルを CMU の XGP で印刷していた時期もあった。

一方同じ頃丘をへだてた反対側に Xerox が PARC という研究所を建て、そうそうたる計算機科学者を集め始めていた。ここでは最も初期の高機能ワークステーションを製作していた。これは ALTO と呼ばれて現在ではあまりにも有名になっているが、当時これを開発したのは、Lampson, Thacker, Metcalfe, Sproull 等 20 代の人々であった。

現在までも ALTO は高機能ワークステーションの原型である。1976年に Xerox PARC でコンサルタントとして働く機会を得て行って見て驚いた。今までまったく見たこともないようなコンピュータがあり、Bravo というすばらしいエディタが動いていた。早速 CMU をやめて PARC に就職した。

さて ALTO の本体は 80 センチ角のメタルの箱にはいって、机の下に置くようになっていた。これには、コンソール・パネルとか、ランプとかいうものは一切なく、スイッチとパワー・ランプがついているだけであった。

ディスプレイは縦型だった。見ればコロンブスの卵であるが、文書を書くにはディスプレイも紙の形に合わせて縦型にしたのは成程と感心させられた。これはビットマップ・ディスプレイと呼ばれ、主記憶中のある領域の 0 と 1 の記憶パターンがそのまま画面上の点の輝度に対応する。こうすると任意の形の文字や図形をディスプレイに表示できる。また白地に黒い字を表示しているが、これは大変読みやすい。

それまでディスプレイ装置というと必ずキャラクタ・ジェネレータがついているハード制御であり、印字できる字の種類は非常に少数に限られていた。一方 ALTO のスクリーンでは文字を扱う特別なコマンドはなく、長方形のビットのパターンをコピーするだけで画面上に字が画けるので、あらゆる字体と大きさの字を表示できた¹⁵⁾。こんなことをしたら遅くて困るだろうと思うだろうが、これもコロンブスの卵で、出来て見ると高速に字を表示できた。

画面を指す道具 (ポインティング・デバイス) としてはマウスが付いている。ライトペンやパッドに比べて精度が高いとか、手を離しても指している位置が変わらないという利点のため、現在ではワークステーションやパソコンの標準ポインティング・デバイスとなっている。

最も有名なのは Ethernet¹⁶⁾ である。ワークステーションが分散してあると、相互の通信はどうしても必要となる。これは LAN の標準品として広く使われている。

またプロセッサは当時としては目新しいダイナミック・マイクロプログラム制御方式であった。ALTO 固有の命令というのはマイクロ命令しかなく、使用される言語に応じてマクロ命令が定義されていた。

ソフトウェアとしても種々のグラフィックスを使用したプログラムが開発された。

まずテキスト・エディタとしては Bravo がある¹⁷⁾。Bravo では画面上の任意の場所に瞬時に字を挿入出来、また文書上の任意の場所に飛ぶことも出来た。任意の字体とサイズの文字を混合した文書を作ることが出来、見たままがきれいにレーザ・プリンタで印刷出来るようになっていた。ソフトウェア工学上最も驚くべきことは、これだけ複雑なプログラムと、フレーム・バッファ、それに Ethernet 通信用ソフトが主記憶 128 K バイトしかないコンピュータの上で動いていたことである。

その他 Laurel¹⁸⁾ という電子郵便システムも作られた。

その後出て来たワークステーションは PERQ⁶⁾ とか MIT-LISP マシン⁹⁾ とか ALTO と同じマイクロプログラム制御であった。

それが 1970 年代の後半に MC 68000 というスーパー・ミニ並みのアーキテクチャのマイクロプロセッサが出て来て、ワークステーションの分野も急速に変わった。

68000 を使ったワークステーションとしてまず Apollo DOMAIN⁴⁾ が現われ、次に SUN ワークステーション^{3), 19)} が出てきた。SUN は Stanford で開発され、商用化された。ALTO で成功したハードウェア機能をすべて備え、UNIX で動いている。

現在ではマイクロ・コード系は言語専用マシン、例えば LISP マシンとか Smalltalk マシン、に多く使われ、マイクロプロセッサ系は汎用型に多く使われている。

3. ハードウェア

3.1 プロセッサ

ワークステーションに使われているプロセッサには大きく分けてマイクロプログラム型とマイクロプロセッサ型がある。

マイクロプログラム型はハードウェア部品が多くなるので高価なものが多い。Xerox の STAR¹⁰⁾、PERQ⁶⁾、Symbolics の 3600⁵⁾ はこの型である。マイクロプロセッサ型は、Apollo²⁰⁾、SUN ワークステーション³⁾、Apple の LISA 等がある。

特殊言語専用マシンである LISP マシン、Smalltalk マシン、また第 5 世代用ワークステーション PSI²¹⁾ はマイクロプログラム型である。

一方マイクロプロセッサ型は兼価に作れることから多く出ていて、特に UNIX を走らせているものに需

要が多い。

3.2 ネットワーク

ワークステーションのネットワークについては本誌 1982 年 12 月号及び 1983 年 10 月号²²⁾ に詳しくのっているの、ここでは簡単に解説しよう。

歴史的には ALTO 用に開発された Experimental Ethernet¹⁶⁾ が最初のワークステーション用のネットワークである。これが増強されて 10 MHz バンド幅になり、IEEE の標準としても推薦されている。

現時点で、数社以上のベンダがいて、ハードもソフトも買ってきて使えるパーソナル・コンピュータのネットワークはこの Ethernet と OMNINET である。OMNINET は、より線を媒体とし、1 MHz の帯域幅で CSMA 方式をとっている。だから衝突が起こると、全パケットの通信完了まで検出できないという欠点があるので、小規模のパソコン向けネットワークである。

Ethernet 用の LSI²³⁾ は 4 グループで開発しており、Intel のチップはすでに販売されており、Ethernet 用のハードウェアは急速に安くなっている。

一番問題になるのはソフトウェアである。Ethernet ではハードウェアの部分は標準化しているが、ソフトウェアは各社独立である。現在広く使われているプロトコルには、DARPA の TCP/IP^{24), 25)}、Xerox の NS²⁶⁾、DEC の DECnet²⁷⁾ がある。TCP/IP プロトコルはアメリカ政府がサポートしていることと、UNIX 4.1 bsd 用のソフトウェア・パッケージが BBN から売り出されていたり、UNIX 4.2 bsd²⁸⁾ には標準品としてついているので、広まりつつある。ただし TCP/IP は複雑なプロトコルで、ソフトウェアが巨大である欠点がある。しかし、これも 256 K DRAM が出現すれば解決されるであろう。

3.3 グラフィックス

ビットマップ・ディスプレイを装備することにより自由自在に文字や図形を描けるようになったのがワークステーションである。特にこれからの工業技術者、その他の専門家はコンピュータと対話をしなければならなくなるので、グラフィックスの重要性は増々大きくなる。

基本原理は、記憶領域に書いた 0 と 1 のビットのパターンが、そのままディスプレイ上の白黒の点となって現われるようにしたものである。ディスプレイ上に現われるパターンを記憶しておくところをフレーム・バッファと呼ぶ。

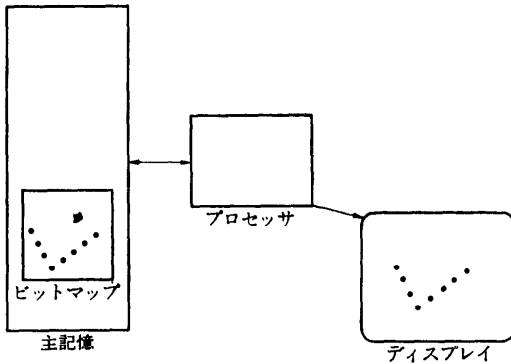


図-1 ALTO のディスプレイ制御はプロセッサがタスク・スイッチで行っている。

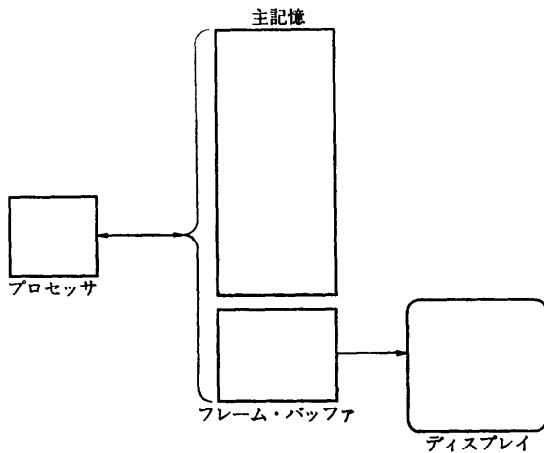


図-2 普通のパソコンのビットマップ・ディスプレイ装置では、フレーム・バッファはプロセッサのアドレス・スペースの固定位置にあり、そこへの読み書きは遅くなる。

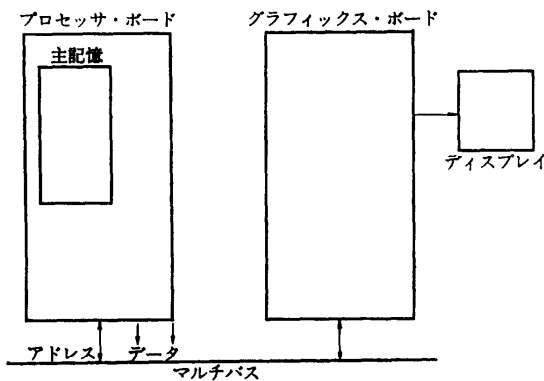


図-3 SUN ワークステーションのグラフィックス・ボード

このフレーム・バッファの取り方にはいくつか方法があり、それによってシステムの性能・価格に差が出て来る。ALTO では、全記憶領域がフレーム・バッファとなりうる。ALTO ではプロセッサは一個であり、一般プログラム実行用、ディスプレイ制御用、その他入出力制御用に共用されている。通常は一般プログラムを実行しているが、平行スキャンが終わるごとにタスク・スイッチが起きて、プロセッサはディスプレイ制御用のタスクを実行する。プロセッサはフレームバッファと指定された記憶域から何ワードか読み、ディスプレイ装置に表示する(図-1)。このため、全画面にビットマップを表示しようとすると7割ものプロセッサ時間はディスプレイ制御に費やされる。これはプロセッサが高かった時代の設計であり、最近のようにマイクロプロセッサが安くなるとこのようなことはしない。Apple LISA²⁹⁾でも主記憶の任意の所をフレームバッファにできるように、ディスプレイ制御用のマイクロプロセッサがサイクル・スティーリングで記憶アクセスをするので、演算用の68000は普通の半分の5MHzで動いている。これは設計の失敗例である。

最近の普通のパソコンに付いているフレーム・バッファは、主記憶中の固定領域にあり、ここだけ2ポートになっていて、プロセッサがフレーム・バッファに読み書きする時だけ遅くなるようになっていて(図-2)。

SUN ワークステーションでは、マルチバスにプロセッサ・ボードとグラフィックス・ボードがついている。プロセッサ・ボードから出されるある128Kバイトのマルチバス空間への読み書き命令が、ラスタ命令と解釈されて、ラスタ操作が起こる。この命令の紹介は本特集のSUNワークステーションの項³⁾を参照されたい(図-3)。

さて以上の命令はハードウェア固有の命令であり、このまま、グラフィックス・ソフトウェアの基本演算とする訳にはいかない。特にソフトウェアの互換性を良くするには、もっと高レベルの操作をソフトウェアとハードウェア・システムのインタフェースに取るべきである。ここではそのようなインタフェースとして Smalltalk-80³¹⁾ で使われている BitBlit (ビット・ブリット) という操作を説明しよう。Smalltalk-80 は ALTO の上で最初開発されたので、ALTO のハードウェアの特質を残している。すなわち、記憶上の任意の場

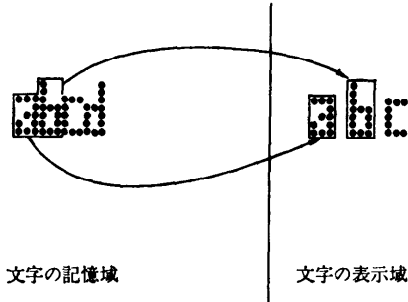


図-4 BitBlt による文字表示

所をフレーム・バッファと指定できる。フレーム・バッファと指定された記憶のビット・パターンがそのままディスプレイに表示される。この方法の利点は、いくつかフレーム・バッファを取っておいて、瞬時に画面をいくつも変えられる点である。

BitBlt は主記憶上の任意の、ディスプレイに平行な長方形をコピーする操作である。ただコピーするだけでなく、コピーの元（ソース）と行き先（デスティネーション）の間で論理演算を行うことが出来、またマスクとクリッピングが指定できる。マスクは一定のパターンを4語で指定しており、これを繰り返して、ハーフトーンのバックグラウンド等を作る。クリッピングは、行き先の大きさを制限する機能である。文字表示等はこれで行っており、記憶上の一部に、文字のパターンを密な形でしまっておき、エディタ等で文字を表示する時はその一部を切り出して BitBlt でコピーする(図-4)。これだと一見遅いようだが、実際にはあまり問題なくいろいろな字体を表示できる柔軟性の利点があるので多くのシステムで使用されている。

3.4 その他の入出力装置

ワークステーションは職場でのペンであり紙でもあり、またその他の多くの道具のかわりをしてる。そのため種々の入出力装置が付いて始めて役に立つ。現在付けられている入出力装置としては、

- ディスク
- 音声入出力装置
- プリンタ
- 画像入力装置

等がある。

4. ソフトウェア

ワークステーションに付いている基本ソフトウェアについて解説する。

表-1 代表的なカーネル・ソフト

ワークステーション	カーネル・ソフト
ALTO ¹⁾	ALTO OS ²⁾
SUN ¹⁾	UNIX 4.2 bsd ¹⁾
STAR ¹⁾	PILOT ⁴⁾
PERQ ⁴⁾	POS
APOLLO ⁴⁾	AEGIS ⁴⁾
Spice ⁴⁾	Accent ⁴⁾
Dolphin } Dorado }	{ALTO OS Inter Lisp-D Smalltalk-80
Symbolics 3600 ⁴⁾	Lispmachine Lisp

4.1 カーネル・ソフト

ここでカーネル・ソフトと言っているものは、記憶管理、プロセス管理、ファイル・システム、ネットワーク通信等を行うもので、従来のオペレーティングシステムと同じ機能を持っているほか、ワークステーションのカーネル・ソフトではネットワーク関係のソフト、グラフィックスのソフトが重要である。代表的なカーネル・ソフトを表-1に示す。

ALTO OS²⁾ は Bcpl で書かれており、Strachey, Stoy の OS6³⁾の思想に基づいて作られている。コマンド・インタプリタは、コマンド・ラインに書かれたファイル名を実行可能なプログラムだと思って実行する。ファイル名のみならず、何個でもパラメータを与えることが出来る。セミコロンでコマンドを続けると順次コマンドを実行していく。またファイル名を@でくくると、そのファイルはコマンド・ファイルと解釈され、ファイルの中身をコマンドとして実行する。コマンド・ファイルは何重にもネスト出来る。

ALTO OS では記憶管理は行っていない。ALTOの上で走る言語には Bcpl, Mesa, Interlisp-D, Smalltalk-80 があるが、それぞれの言語のランタイム・システムが独自に記憶管理を行っている。

ネットワーク・システムとしては、PUP³⁾ プロトコルがあり、その上にファイルを転送する FTP, 郵便を転送する MTP, 文書をレーザ・プリンタで印刷する PRESS 等のアプリケーション・プログラムがある。

ALTO には記憶保護装置がないので、いつたんどきユーザ・プログラムが暴走してカーネルをつぶすか判らない。そこで、ブート・ボタンがあって、システムがこわれるとすぐブート・ボタンを押して全システムを初期化する。そのため常駐するカーネルは何もない。記憶管理にしても、ネットワークソフトにしても

アプリケーション・ソフトが必要とする場合には一緒にロードモジュールとしてバインドされている。デバッグは、アドレス空間が一つしかない場合には、他のアドレス空間で動くことなどできないのでどうしているかという、まず全主記憶をファイルに書き込み、次にデバッグを読み込んで、それはファイルに書かれた主記憶のイメージをデータとして扱っている。

ファイルシステムは、一様な空間で、サブディレクトリはない。

SUN での UNIX 4.2bsd は UC バークレーで開発されたシステムで 4.1bsd との最大の違いは、Ethernet を TCP/IP プロトコル^{24), 25)}でサポートしている。

STAR についている Pilot³⁴⁾ は ALTO の上で Mesa³⁹⁾ を動かす時のランタイム・システムが拡張されて出来たものである。普通の OS だと、応用プログラムが OS の機能を使う時は、スーパーバイザ・コールで行うがパイロットでは普通のプログラムの手続き呼出しと全く同じ形で行える。また、仮想記憶、ダイヤモンド・ページング、ネットワーク・システムも同じ形でサポートしている。

PERQ の POS は Pascal 用の OS である。

Apollo の AEGIS は使ったことが無いので良く知らないが、大変意欲的なネットワーク OS らしい²⁰⁾。オブジェクト指向で作られており、システムのリソースはオブジェクトで、64ビットのユニーク・ネームが付いていて区別される。ファイルはネットワーク上どこにあっても場所を気にせずアクセスできる。

Spice の Accent は PERQ の上で動くカーネル・ソフトであり、特にプロセス間の通信を、プロセスが同じワークステーション上にあるが、違うワーク

ステーション上にあるが可能にしてある。特徴としては、プロセス間の通信はポートを介して行うようにしてある点である。プロセスはポートへメッセージを送り、プロセスはポートからメッセージを受け取る。一つのプロセスしか受け取ることはできないが、ケイパビリティを持っているプロセスはいくつでもメッセージを送れる。またすべてのプロセスは違うアドレス空間を持っている。このため違う言語で書かれたプログラムの共存も可能である。Accent の上に UNIX ものっている。

Dolphin は、Xerox のワークステーションであるが、これには ALTO OS があり、その上に Interlisp-D^{40), 42)} と Smalltalk-80 が動く。それぞれ独立に記憶管理などをやるので、一端 Interlisp-D や Smalltalk-80 が動き出すと、ALTO OS は一切必要ない。

Symbolics 3600 は Dolphin と同様に LISP マシンとして使われる。これには Flavor という対象指向言語が LISP のサブシステムとしてあって、ウインドウ・システムの記述に使われている。

4.2 ウインドウ・システム

ウインドウの考えは古くは Stanford のエディタの TVED に多重スクリーンという機能があり、一つのファイルの別々の所を同時にディスプレイ上で見るようにして、エディタの能率を上げたのが最初であろう。その後 ALTO の上で動く Bravo エディタにも多重スクリーン機能があり、これは標準化した。なぜエディタで多重スクリーンが必要かという、良い文章を作るにはグローバルな修正を多くしなければならぬ。狭いディスプレイ上に広い文書を写し出すためには多重スクリーン機能がどうしても必要になる。これらはどれも一つの画面を横割りにするだけであった。

本格的なウインドウ・システムは Smalltalk で最初に考えられた。任意の大きさのエリアをディスプレイ上の任意の場所に取れて、そこでファイルを見たり、またプログラムを走らせることができる。このエリアをウインドウと呼び、枠で囲まれていて左上に名前が付けられる。図-5 に Smalltalk でのウインドウの例をあげる。

ウインドウ・システムは特にモジュール化した言語のプログラム作製には必要である。プログラムの作製は最も複雑な頭脳活動の一つであり、同時にいくつかのプログラムを見ながら、数カ所に変更を加えつつ作製していく。そのために、それらの箇所を一度に見え

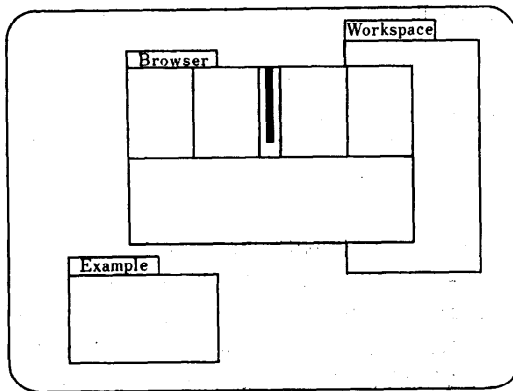


図-5 Smalltalk におけるウインドウ

るようにしなければならず、ウインドウ・システムなしではプログラム作製を全部エディタ上で行うのは不可能であろう。

その他 OA 用とか CAD 用にもウインドウ・システムは有効性を示している。

5. 応用分野

ワークステーションはまだ高価であるので、日本ではまだ広く出まわってはいない。しかしマイクロエレクトロニクスの進歩と量産効果で、急激に安くなるであろう。そうなったとき、どれ程広い分野で使われるようになるかはソフトウェアの豊富さによって決まってくる。ここではどんな所ですでに使われているか解説するが、ソフトウェア次第でもっと広い応用分野があることを言うておこう。

5.1 プログラム作製

Symbolics 3600 や SUN ワークステーションが急激に販売台数を増やしているのは、LISP なり UNIX の使用者が、旧来の TSS によるプログラム作製に比べてワークステーションでのプログラム作製の方が格段と生産性が上がると判断したからである。

ワークステーションでのプログラム作製の方が生産性が上がる理由は、

- エディタがすぐれているので、プログラムの打ち込み、修整の速度が速い。
- ウインドウを沢山開けて、同時にいくつものファイルを見たりエディットできる。
- 図や表を用いて、プログラム・ライブラリを表示したり、プログラムの構造を示したりできるので、プログラムの理解力が増す。
- ワークステーションの画面は TSS 端末の画面よりずっと大きく、沢山の情報を表示できる。
- ウインドウはスクロールができるので、前に出てディスプレイから消えてしまった情報もスクロールダウンして見ることができる。
- 応答時間が一定である。

等である。

更に最近出て来た Smalltalk-80 のプログラム作製支援環境には、すべてのプログラムを木構造のライブラリにして見ることができるシステム・ブラウザとか、プログラムの虫で脱線したときの原因究明や変更・再実行可能なデバッガを持っており、プログラム生産性は LISP マシンよりも高いと思われる。

5.2 ドキュメント作製

レーザー・プリンタの普及化と同時に、見た目の良いドキュメントを作りたいという要求は大変高くなる。ドキュメント作りのパイオニアはなんといってもアメリカであり、Stanford, CMU, Xerox PARC, ベル研等で現在までのほとんどの仕事は行われて来た⁴³⁾。

ドキュメント作りには2通りの方法がある。ひとつはドキュメント・コンパイラを使う方法で、これはドキュメントの中に、ドキュメント・コンパイラへの指示をするコマンドと本文となるテキストを混ぜて書き、それをドキュメント・コンパイラというプログラムに通して印刷出来るフォーマットを作る。これの初期のシステムには PUB⁴⁴⁾ と roff がある。現在は TEX⁴⁵⁾ と SCRIBE⁴⁶⁾ が最も充実しているが、SCRIBE は有料なので roff もまだ広く使われている。TEX は数式等を印刷するのに適しているが、コマンドが大変むずかしく一般向きではない。SCRIBE はかなり簡単で、特に決まったフォーマットのドキュメントを作るに適している。

もうひとつのドキュメント作成法は、What-you-see-is-what-you-get 型エディタを使って画面上で、印刷された時とまったく同じものを作製する方法である。Bravo とか STAR エディタ、LISA Write 等はこの種類に入る。

5.3 OA

ドキュメント作製以外の OA への応用にはどんなものがあるだろうか。

OA 分野へはパソコンやワープロの進出は著しいが、Xerox の STAR や Apple の LISA 等高機能ワークステーションの進出はまだまだである。これは事務分野では高機能ワークステーションはまだ高価すぎるからだろう。

値段の問題はいずれ解決されるが、その時にはどのような用途があるだろうか。一番有望なのは電子郵便である。筆者は10年間の滞米中 Stanford, CMU と Xerox PARC にいたが、これらはすべて一つの電子郵便網の中にあり、研究者間の対話、ファイルのやりとり等に使われている。また Xerox PARC 所内でも電子郵便は最も重要な通信手段であり、単に郵便をやりとりするばかりでなく、会議の議事録を作るとか、日程表を作るのにも使われている。

5.4 CAD, CAM

CAD, CAM 分野へはワークステーションは急速

に進出している。これは低価格で精度の高いグラフィックス・システムを供給しているからである。特に VLSI の設計用とか石油井戸調査に急速に導入されている。これは、これらの応用分野は高価な商品を作るので、投資効果があがりやすいこと、またそれまで使っていたシステム、例えば CALMA の設計システム、等に比べてずっと安いことによる。

特に最近の特徴としては、CAD 会社がワークステーション会社から OEM で買って来たワークステーションに自作のソフトウェアを付けて売り出していることで、Silvar-Lisco は Apollo に乗せているし、CADLINK は SUN ワークステーションのライセンスを Stanford から取って自作している。

5.5 教育用

現在アメリカの大学ではキャンパス内にネットワークを張り、学生全員にワークステーションを持たせようというプロジェクトが大流行である。筆者は 1980 年の bit 誌⁴⁷⁾で、1985 年までにはアメリカの大学の計算機科学の研究施設はワークステーション中心に取って変われると予測したが、これはすでに 1985 年を待たずして実現している。CMU での Spice プロジェクトも大きな峠を越え、外部にソフトを供給し始めている。Stanford は SUN ワークステーションで、MIT は Symbolics 3600 で計算設備を強化している。

さて次の予測であるが、1985 年までには現在の SUN ワークステーション程度のもは 100 万円以下になり、1990 年にはアメリカのかんりの大学でワークステーションは学生全員に筆記用具のように使われているであろう。その頃に使われるワークステーションの計算能力は現在の 10 MHz 68000 より 4~10 倍程度のものであろう。

この分野で一番進んでいるのは CMU で、ここでは IBM と協同で ITC (Information Technology Center) という組織を作り、所長には筆者の親友のジム・モリスを Xerox PARC から招いて所員 35 人でシステムの開発を行っている。その他 Brown, Purdue, MIT, Stanford 等でも同じような計画を進めている。

これ程各大学でワークステーション導入が真剣に考えられているのは、1980 年代後半から大学への進学希望者が減る予測があり、その中で大学が存続していくには、学生になるべく魅力のある所にしなければならぬという危機感があるからである。

また、最近のマイクロエレクトロニクス、コンパイラ作成技術、コンピュータ・アーキテクチャの進歩により、超高速のマイクロプロセッサ作製が可能になったことも見逃がせない。

6. 将来の展望と課題

ワークステーションのハードウェアは標準化の一途をたどっている。ワークステーションは現在 1000 万円もするが、適当な市場競争さえ起きれば、数年後には 100 万円になるであろう。その価格低下を実現する原動力は VLSI チップの出現である。これにはマイクロプロセッサ、ビットマップ・ディスプレイ用チップ、ディスク制御用チップ、イーサネット用チップがいかに速く使われるかによる。またその他の部もゲートアレイ化されるだろう。

そして、ワークステーションがいかに広く浸透するかはソフトによる。いろいろの分野ごとにソフトが揃ってくれば、それぞれの分野の専門家によって使いやすく販路も広がって来る。その意味で CMU と IBM の共同事業はワークステーションの将来の伸びをすすめるうえで重要な事業である。

ワークステーションに関する新しい技術としては、3次元映像を一番注目したい。すでにビットマップ・ディスプレイで 2次元ウインドウを作る技術は確立された。これからは 3次元物体をディスプレイで見ることが重要な使い方になる。その意味からも Jim Clark の geometry engine⁴⁸⁾はこの分野のパイオニアである。

参考文献

- 1) 日本経済新聞, 10月11日(1983).
- 2) 坂村 健: 高機能ワークステーションのアーキテクチャ, 情報処理, Vol. 25, No. 2 (1984).
- 3) 多田好克: SUN ワークステーション, 情報処理, Vol. 25, No. 2 (1984).
- 4) 山村紀夫: アポロ DOMAIN, 情報処理, Vol. 25, No. 2 (1984).
- 5) 元吉文男: Lisp マシン, 情報処理, Vol. 25, No. 2 (1984).
- 6) 齊藤信男: 高機能個人用計算機としての PERQ, 情報処理, Vol. 25, No. 2 (1984).
- 7) 鈴木, 寺田, 多田, 和田: 三二郎プロジェクトにおけるシステム・ソフトウェア, 情報処理, Vol. 25, No. 2 (1984).
- 8) 齊藤信男: 高機能ワークステーションの応用, 情報処理, Vol. 25, No. 2 (1984).
- 9) 上林憲行: ワークステーションユーザインタフェースの構成技術, 情報処理, Vol. 25, No. 2

- (1984).
- 10) 上林, 伊東, 上田: オフィスプロフェッショナルワークステーション (JStar) と統合プログラミング環境ワークステーション (1100 SIP), 情報処理, Vol. 25, No. 2 (1984).
 - 11) ジュリアーノ, V. E.: オフィスの機械化, サイエンス, Vol. 12, No. 11, pp. 102-115 (1982).
 - 12) Thacker, C. P. et al.: *Alto: A personal computer*, Report CSL-79-11, Xerox Palo Alto Research Center (Aug. 1979).
 - 13) Tesler, L.: *PUB: The Document Compiler*, Operating Note 70, Stanford Artificial Intelligence Project, Stanford, Calif. (Sep. 1972).
 - 14) Reid, B. K.: *Scribe: A Document Specification Language and Its Compiler*, Ph. D. dissertation, Computer Science Dep., Carnegie-Mellon Univ., Pittsburgh, Pa. (Oct. 1980).
 - 15) Newman, W. M. and Sproull, R. F.: *Principles of Interactive Computer Graphics*, 2nd ed., McGraw-Hill, New York (1979).
 - 16) Metcalfe, R. M. and Boggs, D. R.: *ETHERNET: Distributed Packet Switching for Local Computer Networks*, *Comm. ACM*, Vol. 19, No. 7, pp. 395-404 (1976).
 - 17) Meyrowitz, N. and van Dam, A.: *Interactive Editing Systems: Part II*, *ACM Computing Surveys*, Vol. 14, No. 3, pp. 353-416 (1982).
 - 18) Brotz, D. K.: *Laurel Manual*, Report CSL-81-6, Xerox Palo Alto Research Center (1981).
 - 19) SUN Microsystems: *User's Guide for the SUN Workstation*, Part No. 800-0400 (Oct. 1982).
 - 20) Nelson, D. L.: *Apollo Domain Architecture*, Apollo Computer Inc. (1981).
 - 21) Yokota, M. et al.: *The Design and Implementation of a Personal Sequential Inference Machine: PSI*, New Generation Computing, Vol. 1, No. 2, pp. 125-144 (1983).
 - 22) 鈴木則久: パーソナルコンピュータ・ネットワーク, 情報処理, Vol. 24, No. 10, pp. 1282-1286 (1983).
 - 23) イーサーネットの普及を促すネットワーク・コントローラ LSI, 日経エレクトロニクス, 1981年1月3日号, pp. 81-104.
 - 24) Postel, J. ed.: *Internet Protocol, DARPA Internet Program Protocol Specification*, RFC: 791, DARPA (Sep. 1981).
 - 25) Postel, J. ed.: *Transmission Control Protocol, DARPA Internet Program Protocol Specification*, RFC: 793, DARPA (Sep. 1981).
 - 26) *Internet Transport Protocol*, Xerox System Integration Standard, Xerox Corporation (1981).
 - 27) *DECNet, DIGITAL Network Architecture (Phase IV) General Description*, Digital Equipment Corporation (1982).
 - 28) Joy, W. et al.: *4.2 BSD System Manual*, U. C. Berkeley (Sep. 1982).
 - 29) Williams, G.: *The Lisa Computer System*, *Byte*, pp. 33-50 (Feb. 1983).
 - 30) Morgan, C.: *An Interview with Wayne Rosing, Bruce Daniels, and Larry Tesler*, *Byte*, pp. 90-114 (Feb. 1983).
 - 31) Goldberg, A. and Robson, D.: *Smalltalk-80: the Language and its Implementation*, Addison-Wesley, Reading, Mass (1983).
 - 32) Lampson, B. W. and Sproull, R. F.: *An Open Operating System for a Single-user Machine*, *Operating Systems Review*, Vol. 13, No. 5 (1979).
 - 33) Stoy, J. E. and Strachey, C.: *OS 6-An Experimental Operating System for a Small Computer*, *Computer Journal*, Vol. 15, No. 2 and No. 3.
 - 34) Redell, D. D. et al.: *Pilot: An Operating System for a Personal Computer*, *Comm. ACM*, Vol. 23, No. 2, pp. 81-91 (1980).
 - 35) Rashid, R. and Hibbard, P.: *Research into Loosely-coupled Distributed Systems at CMU*, Notes from IEEE Workshop on Fundamental Issues in Distributed Systems (1980).
 - 36) Rashid, R. F. and Robertson, G. G.: *Accent: A Communication Oriented Network Operating System Kernel*, Proc. of the Eighth Symposium on Operating Systems Principles, ACM (Dec. 1981).
 - 37) Symbolics, Inc.: *Symbolics Software*, 21150 Califa Street, Woodland Hills, CA 01367 (1981).
 - 38) Boggs, D. R. et al.: *Pup: An Internetwork Architecture*, Technical Report CSL-79-10, Xerox Palo Alto Research Center (July 1979).
 - 39) Geschke, C. M. et al.: *Early Experience with Mesa*, *Comm. ACM*, Vol. 20, No. 8, pp. 540-553 (1977).
 - 40) Masinter, L. and Deutsch, L. P.: *Local Optimization in a Compiler for Stack-based LISP Machines*, Proc. of the 1980 LISP Conference, pp. 223-230 (1980).
 - 41) Deutsch, L. P.: *ByteLisp and its Alto Implementation*, *ibid.*, pp. 231-242 (1980).
 - 42) Burton, R. et al.: *Overview and Status of Dorado LISP*, *ibid.*, pp. 243-247 (1980).
 - 43) Furuta, R. et al.: *Document Formatting Systems: Survey, Concepts, and Issues*, *ACM Computing Surveys*, Vol. 14, No. 3, pp. 417-472 (1982).
 - 44) Tesler, L.: *PUB: The Document Compiler*, Operating Note 70, Stanford AI Project, Stanford, CA (1972).

- 45) Knuth, D.E.: *TEX and Metafont: New Directions in Typesetting*, Digital Press and the American Mathematical Society, Bedford, MASS (1979).
- 46) Reid, B.K. and Walker, J.H.: *SCRIBE Introductory User's Manual*, Unilogic, Pittsburgh (1980).
- 47) 鈴木則久: 1980年代の研究・教育の課題, bit, 通巻第159号, pp. 78-79 (1980).
- 48) Clark, J.H. and Hannah, M.R.: *Distributed Processing in a High-Performance Smart Image Memory*, LAMDA, 4th Quarter, pp. 40-45 (1980). (昭和58年10月19日受付)
-