Description of the SEP Computer Facilities

Chuck Sword
Jeff Thorson

Occasionally we are asked to describe out computer system -- what kind of computer we use, which operating system, what sorts of peripherals, and so on. In order to answer these questions in a fairly systematic fashion, we have written this paper. First, here is a summary of what we have:

Stanford Exploration Project Computer System
(as of April 11, 1982)

Mainframe:	DEC VAX11/780 with Floating Point Accelerator
Word Size:	32 bits
I/O Buses:	2 DEC UNIBUSes
	2 DEC MASSBUSes
Memory:	4 megabytes RAM (1 byte = 8 bits)
Operating System:	Berkeley Version 4.1 VAX UNIX (UNIX is a Trademark of Bell Laboratories)
Array Processor:	Floating Point Systems AP120B
Disks:	1 492-megabyte DEC RP07 disk
	2 244-megabyte CDC 9766 disks on SI 9400 controller
	1 166-megabyte DEC RP06 disk
Tape Drive:	DEC TU77 (800 and 1600 bpi, 125 ips)
Plotters:	11-inch Calcomp-Gould 5200, 200 rasters per inch
	11-inch Benson-Varian, 200 rasters per inch
	AED 512 graphic display terminal, 512 by 512 pixels, 8 bits per pixel
Video Recorder:	Sony 3/4-inch Umatic VTR attached via Ikegami converter to AED
Lineprinter:	Printronix 300
Terminals:	7 Datamedia Elite 2500 terminals
	9 DEC GIGI color graphics terminals
Dial-In Lines:	2 1200-Baud lines
	1 300-Baud line
Dial-Out Line:	1 300-Baud line

1. Hardware

1.1. Mainframe

Our mainframe, a DEC VAX11/780, is probably the most popular 32-bit minicomputer at this time. Installed in our VAX is an optional DEC Floating Point Accelerator, which executes floating point instructions much faster than the standard VAX microcode can. If we purchased a second memory controller, we could expand our random-access memory to 8 megabytes, but we have no current plans to do so. For communications with our peripherals, the VAX has two MASSBUSes and two UNIBUSes. The MASSBUSes are used for communicating with the RP06 and RP07 disks and the TU77 tape drive, while all of our other peripherals are connected to the UNIBUSes.

1.2. Array Processor

For large-scale processing of real data on a minicomputer, an array processor is indispensible. We have an early version of Floating Point System's AP120B Array Processor with slow memory. It can hold up to 65,536 floating point numbers, as well as holding programs written in AP Assembly Language (APAL) or chained with the Vector Function Chainer. Communication rather than computation is often the main bottleneck in using the AP. That is, it sometimes takes longer for the VAX to process a call to the AP than it does for the AP to execute that call. The Vector Function Chainer gets around that problem by allowing the programmer to combine several individual AP calls into one big call, thus reducing the overhead. This sort of AP programming is fairly straightforward on the AP120B. Although it is more difficult than using the Vector Function Chainer, students here have also written programs in APAL (see Jacobs, 1981).

Those of us who do not use the array processer are often tempted to get rid of it. It seems to break down at least twice a year, and repairs are not cheap. We don't throw it away, however; the reason is speed. The AP can run much faster than the VAX. For instance, a velocity analysis program that takes 15 minutes of CPU time on the VAX takes about 60 seconds of *elapsed* time when run on the AP. Unless we get ourselves a large, fast mainframe computer, we are likely to continue using the AP and putting up with its problems.

1.3. Memory

We currently have 4 megabytes of random-access memory. This is usually more than we need, but when we are running large movies it comes in handy. We bought our first two megabytes from DEC, and our other two megabytes from Standard Memories, which was, at that time, selling VAX-compatible memory at 40 percent of DEC's price and with 3-week delivery. We haven't had any problems with either vendor's memory.

1.4. Disks

Since we have only one tape unit, most of our data processing involves disk-to-disk work. Thus, even though we do not generally process huge data sets, we still need quite a bit of disk storage. At this time we have, theoretically, more than a gigabyte of on-line disk storage. (The theoretical figures are shown in the table on the first page of this paper.) About eight percent of that amount, however, cannot be used for data storage, either because it is being used to store filesystem information, because it is used for swapping, because we are not making full use of all the disk cylinders, because some of the sectors are bad, or because it is reserved for bad-block forwarding. This leaves us with 1058 megabytes that can be used for storing the operating system, source files, on-line documentation, disk-to-disk backups, our programs, and our data. (A reel of 1600 bpi tape, by the way, holds about 32 megabytes.)

We have had the most trouble with our SI (CDC) disks, while we have so far had no problems at all with our DEC RP07 unit (not too surprising, since we have only had it since last August). Actually, only one of the SI disk units has given us any real trouble, and that problem was finally solved when SI (Systems Industries) took the unit away and gave us a new one. We have not had any head crashes for about a year now, partly, perhaps, because we never swap disk packs in our SI units and only occasionally in our RP06 (the RP07 disk is sealed and cannot be removed). However, just in case, we make daily RP06-to-RP07 disk backups and weekly disk-to-tape backups of our main file systems. We are, by the way, thinking of buying another RP07 to replace our two SI units.

1.5. Tape Drive

Our tape drive is a DEC TU-77, which can read and write at 800 and 1600 bpi, at about 125 inches per second. We are generally satisfied with it. However, it has sometimes had problems with auto-loading (and if the auto-loading doesn't work, it can't be loaded at all), and it is limited in that it can only handle record lengths of 64 kbytes or less. We now have on order a DEC TU-78 tape drive, which will be able to read and write at 1600 and 6250 bpi. DEC is not the cheapest vendor for the 6250 bpi tape drive, nor necessarily the best, but we chose to buy the drive from them because we are satisfied with their ability to service it.

1.6. Hard-Copy Plotters and Lineprinter

For making hard-copy plots, we have two plotters: the Calcomp (Gould) 5200, and the Benson (Varian). The Gould has been around a lot longer than our VAX, and it is getting old and cranky. We keep it around, however, since it can help relieve the load on the Varian, and because it is sometimes useful as a backup when the Varian breaks down. Although both plotters claim to have 200 rasters per inch, the Varian's output looks a bit cleaner and definitely comes out faster. This paper, by the way, was printed on the Varian. Both plotters have their little quirks. The Gould needs to have its humidity salts watered every day, and when a roll of paper has been in the Gould for too long, the plots start fading away. The Varian, on the other hand, drops a few bytes now and then, with the result that long plots sometimes come out looking a bit scrambled.

Our Printronix lineprinter can also make plots, although we very seldom use it for that. It has only rarely caused us any problems, and it doesn't need any day-to-day maintenance, so there isn't a whole lot to say about it.

1.7. AED

Our latest acquisition in graphics technology has been the AED 512 color graphics terminal. This terminal was described in SEP-28 (Ottolini, 1981), so we won't go into great detail. Suffice it to say that we can easily fill its 512 by 512 pixel screen in about one second, and we can fill fractions of the screen in fractions of that time. This gives us the capability of making real-time movies, many of which can be seen in SEP Tape 1, which was handed out in conjunction with SEP-28. We were not able to record videotape directly off of the AED, since the AED color display uses the RGB video system, while our videotape recorder needs NTSC format video signals. We bought an Ikegami converter to make the RGB to NTSC transition. Since the time that the SEP videotape was handed out, our main improvement has been the addition of joystick control. This means that the user can now choose any particular frame, or run the movie back and forth, simply by moving the joystick attached to the terminal.

There are a few limitations. Since it does take about a second to fill the entire screen, it is not possible to make full-screen high-resolution movies. So far we have been storing all the movie frames in the VAX core, and sending them to the AED as needed, one at a time. This means that with 4 megabytes of core, there are certain limits on the size of any movie intended to be shown in real time. We have gotten around some of these limitations, however, by purchasing a 16mm movie camera that is capable of stop-frame animation under computer control. So far we have made only one experimental movie this way, but it worked out well. This is obviously the way to make movies that are too large or have too high a resolution to be made in real time and recorded on videotape.

It has been more than six months since we have needed to talk to AED field service (we have had the AED now for about 9 months), and it is possible that they have gotten a bit more organized since when we dealt with them, so we won't circulate (in writing, anyway) any of our horror stories. We also won't recirculate (here) the rumors floating around the Stanford campus about newly-delivered AED 512s that didn't work. Instead, we will simply note that ever since we got it working reliably, we have been quite happy with our unit, and we are constantly coming up with new applications that involve movies of seismic data.

1.8. Terminals

We have two type of terminals: the Datamedias, which predate the VAX, and the DEC GIGI color graphics terminals. There is not much to say about the Datamedias, except that they look sort of home-made, they work quickly and reliably, and they are apparently no longer manufactured. We have seen newer terminals that are cheaper, nicer-looking, and just as fast as our Datamedias (the Televideo 950, for instance), but we have not used them on our system, and so are unable to evaluate them.

The GIGI terminals are briefly described in SEP-28 (Claerbout, 1981). They have a resolution of 384 by 240 pixels (quite a bit less when color is used), eight colors (counting black), and can be used as ordinary computer terminals. We mainly use their graphics capability when we want a quick look at a line graph before we make a hard copy of it. The color capability is handy, although we cannot currently make color hard copies. The graphics capability is especially useful when the GIGI is taken home and hooked to the VAX over a 1200 baud modem, since of course it is convenient to be able to see some preliminary plots without having to come back onto campus.

The authors of this paper have always had a few doubts about GIGIs, for two main reasons. First, although they claim to be 9600 baud terminals, they have always seemed to run quite a bit slower than the Datamedias. This is especially noticeable when scrolling through long programs or text files. Second, they lack something called Insert-Delete Mode. This means that the full-screen editor that we use cannot run quite as nicely on GIGIs as on

Datamedias. Ours seems to be the minority view, however. Recently we decided to purchase some new terminals, since in most of the student offices there were more students than terminals. When we asked everyone whether they wanted GIGIs or something more equivalent to the Datamedias, they all chose GIGIs. (The 5 new GIGIs that we ordered have not yet arrived, so they are not included in the table on the first page of this paper).

2. Software

2.1. UNIX Operating System

We have used the UNIX operating system (UNIX is a Trademark of Bell Laboratories, by the way) ever since we bought our PDP 11/34, and when we went from the 11/34 to the 11/70, and then to the VAX, we stayed with UNIX. Our particular dialect of UNIX is known as Berkeley Version 4.1, since it is maintained by UC Berkeley for VAX computers under some sort of Defense Department contract. We think that UNIX is one of the nicest features of our computer. We won't describe many of the specific features, but more detailed descriptions can be found in Kernighan and Morgan, 1982, and Thomas and Yates, 1982. The native language of UNIX systems is C, which is a higher-level structured language equivalent to Algol, Pascal, or Ratfor. One of the useful features of C, and of UNIX as well, is the device-independent I/O. This means that you don't have to worry too much about whether you are writing to (or reading from) a disk file, a tape, or a terminal. Most of the operating system itself is written in C, as are all of the device drivers. For those of us in charge of maintaining the system, this means that when we write a driver for a new device or try to find the cause of a crash, we don't need to deal with machine or assembly language.

Included as part of the UNIX operating system came some extremely useful software, including the full-screen editor, and a typesetter that can handle equations. The full-screen editor is no longer a novelty for anyone who has used any of the more recent computer operating systems, but the kind of typesetting capability that we have is still fairly unusual. We now can enter our paper into the VAX, use the typesetting software to print it out on the Varian, edit the paper until we like it, and put it, camera-ready, into the SEP report, with the aid of a part-time secretary whose job no longer includes typing final manuscripts and trying to decipher illegible equations. As a result, our SEP reports contain results that have been obtained right up to the deadline for sending material to the printer, and our secretary no longer has to take the blame for spelling errors and erroneous equations. The last few SEP reports have been produced in this way, and we think that this kind of do-it-yourself word processing is the wave of the future, especially for any sort of scientific or technical papers that contain a number of equations.

There are, of course, disadvantages to any operating system. Here are some that we have encountered in Berkeley Version 4.1. One problem is that UNIX, and especially Berkeley UNIX, is not yet in a standardized form. Whenever Berkeley puts out a new version, we have to make various changes in our system -- some small, some large. After the last revision, we had to make some changes in all of our device drivers (how to go about doing this was fairly well explained in Berkeley's documentation), and rumor has it that the next revision will necessitate changes in our file systems. We have also heard that Berkeley's virtul memory swapping system is not as efficient as it might be, so that when our core memory doesn't have enough room for a program, the system slows down more than it should. We feel, however, that the benefits of using UNIX far outweigh the problems. (Another disadvantage of the Berkeley UNIX is the number of games that are provided. We have probably lost several man-months to a certain game called "rogue".)

2.2. SEP Software

Of course, UNIX does not provide all the software that we need for our research, so we have developed quite a bit ourselves. This software includes programs for reading seismic data tapes of various formats, as well as for windowing, plotting, and processing seismic data. We have written most of the programs used for plotting on the AED, the GIGIs, the Gould, and the Varian. Our plot language is device-independent, so a plot that looks good on a GIGI can subsequently be run off on a hard-copy plotter. Most of our programs are written in C, with Ratfor (Rational Fortran, a structured version of Fortran (Kernighan and Plauger, 1976) coming in second.

3. Conclusion

In general, we are quite happy with our system. It is small and slow compared to any sort of large IBM-type computer, but our system has the sort of flexibility, ease of use, and quick turnaround conducive to research, and since usually no more than 10 people are using it at any one time there is seldom much of a response-time problem. One advantage of a system this size is that we can (and do) operate it ourselves, rather than having to put it under the control of some data-processing bureaucracy. We have no plans to get a bigger or faster computer, or to change over from a UNIX-based operating system. While we will continue to upgrade our system in terms of on-line and off-line storage, plotters, and terminals, the basic configuration of our system will remain the same for at least another couple of years.

REFERENCES

Claerbout, J.C., 1981, On-Line Movies: SEP-28, pp. 1-4.

Jacobs, B., 1981, An AP Program for Gaussian Elimination of Banded Complex Matrices: SEP-28, pp.177-188.

Kernighan, B.W., and S.P. Morgan, 1982, The UNIX Operating System: Science, v. 215 (12 Feb. 1982), pp. 779-783.

Kernighan, B.W., and P.J. Plauger, 1976, Software Tools: Reading, Massachusetts, Addison-Wesley Publishing Co.

Ottolini, R., 1981, Color Graphics Terminals at SEP: SEP-28, pp. 5-10.

Thomas, R., and Yates, J., 1982, A User Guide to the UNIX System: Berkeley, OSBORNE/McGraw-Hill.