KBN

U-NEWS

FEBRUARY 1984 VOLUME 1, NUMBER 2 ISSN 0742-1206

MORE NEWS FROM UNIFORUM

CLARIFICATIONS A few items in the first section of this report last month turned out to be either incomplete or misleading. Texas Instruments is the last major manufacturer of minis and micros to offer Unix; there are still many medium and smaller manufacturers who are not even working on Unix ports. Lucasfilm stores information about each frame of Star Wars in a /rdb database, not digitized images of the frames themselves. Recent ad hoc enhancements to APL have split the language into two versions, incompatible both theoretically and practically. These versions' adherents cluster around their respective gurus: Ken Iverson, the original inventor of APL, and Trenchard More, the prime mover behind array theory. STSC's APL is based on Dr. More's ideas.

URBAN SOFTWARE released yet another of their free information collections re Unix systems. This time it's tables of applications software and a few support services, with a brief reprise of their previous hardware listings. As usual, the print listing includes a rundown of a few awk programs used to filter the data into form for tbl. This time, though, Urban didn't stop at a print version; they also put up their database for public access on three timesharing services, Calculon, CCA and Fargo. The database itself is open for unlimited, no-fee use on those systems or anywhere else you may care to move it (it's not copyrighted), and Urban also maintains a copy of their Leverage data handler on each of these systems, also for public use. Urban Software Corporation, 330 West 42nd Street, New York, NY 10036; phone 212-736-4030.

UNICORN, the Unix users group exclusively for federal employees, is not a new organization, but it was visible to the outside world for the first time at UniForum. Unicorn promotes information interchange among Unix users from different federal agencies at its semiannual meetings - always in or around Washington, DC, of course. The next of these confabs will be held the 9th and 10th of May. Any federal employee may write for more information to: Unicorn, c/o Robert Borochoff, Federal Judicial Center, 1520 H Street Northwest, Washington, DC 20005.

NIAL SYSTEMS announced commercial availability of Q'Nial, an interpreter for Nested Interactive Array Language. This is a new language based in APL, but with advances borrowed from Lisp and structured programming, and a number of indpendent changes to get around some of the obscurity of APL - the most noticeable of these are a standard ASCII character set replacing APL squiggles, English-named functions instead of APL's single characters, and automatic diagramming of nested data structures. Nested Interactive Array Language (or Nial) was designed jointly by Michael Jenkins, who headed development of the Q'Nial interpreter at Queen's University in Ontario, and Trenchard More, mentioned above. Q'Nial is written in C and runs on top of Berkeley Unix on Vax hardware; ports to other systems are promised later in 1984. It interfaces to Unix for file management, text editing and command processing - wise decisions considering the usual facilities for these tasks in APL systems. Nial Systems Limited, P. O. Box 2128, Kingston, ON K7L 5J8, Canada; phone 613-549-1432.

DENELCOR was explaining HEP/UPX, which is 4.2BSD for their supercomputer line. (The first Unix from a supercomputer manufacturer; nothing is expected from Cray until the Cray 2 debuts.) Denelcor's supercomputers feature lots of paralleling, for speed and for soft-fail purposes. Their Unix looks like rather complete 4.2, plus extras such as SCCS and an optimizer for both C and Fortran 77. User customization can be done by use of their optional software for hacking up the OS, right down to the kernel level. (This goes farther than mere reconfigurability.) To license HEP/UPX, you first obtain a Unix license from AT&T, then license 4.2BSD from Berkeley. Only then can you get HEP/UPX from Denelcor. Their own prices start at $15,000 - for either object or source code. Denelcor is in Aurora, Colorado.

QMS announced troff compatibility software for their Lasergrafix 1200, a 300 x 300 dot per inch laserprinter. Qtroff takes output from old, device-dependent troff running on System III, System V or 4.1BSD and translates it for input to the intelligent front end of their laserprinter. Qtroff licenses for $1500 in C source form, and the Lasergrafix 1200 has a one-price of $25,000. Quality Micro Systems, P. O. Box 81250, Mobile, AL 36689; phone 205-633-4300.

GOULD was showing parts of what must be the broadest size range of Unix systems offered by a single company. The smallest was the PowerStation 1000, built on Intel's 8 megahertz 80186 chip and occupying $\frac{1}{4}$ to $\frac{1}{2}$ cubic foot of desk space, not counting the display and keyboard included with it. The largest is the Concept 32/8780, which "Provides 10 Times VAX 11/780 Performance", according to Gould. The larger Gould systems were developed in-house, the smaller ones OEMed from Convergent Technologies in silicon valley. James Clark, Gould's Unix Product Manager, explained at some length that Gould intends to bring the versions of Unix for its various size systems to where a piece of software that will run on one will run on them all. Computer Systems Division, Gould Inc., P. O. Box 9148, Fort Lauderdale, FL 33310-9148; phone 305-587-2900.

UNIDOT SYSTEMS, known up to now as advocates and suppliers of Don Knuth's TeX typesetting system for Unix system use, introduced their Cerebra line of microsystems, based on National Semiconductor's NS16032 chip and Genix port of 4.1BSD. In keeping with the name, dual CPU as well as single CPU models are offered. Both feature Multibus, intelligent controllers, cache memory, NS16081 floating-point chips, virtual memory, full-system battery backup to allow graceful shutdowns when the power fails, and lockable cabinet that even prevents tampering with the I/O cables. Cerebra singles prices start around $37,000. Unidot Systems, 602 Park Point Drive, Golden, CO 80401; phone 303-526-9263.

JMI SOFTWARE showed Bastoc, their Basic-to-C translator running on yet another version of Unix - Uniplus+ on the Apple Lisa. They also spoke of teaching Bastoc more dialects of Basic in the near future, including Cbasic. JMI Software Consultants, Inc., 1422 Easton Road, Roslyn, PA 19001; phone 215-657-5660.

ARETE showed the first Unix system I've seen that's designed for transaction-processing applications (e.g., automated bank tellers). Areté has gone at this job by modifying System V (with Berkeley features added) to handle up to four close-coupled 68000 CPUs. Their systems also feature four buses, cache memory, very high I/O bandwidth, plenty of internal diagnostics, ECC memory, mirrored (duplicated) disks, auto restart and, of course, full battery backup. Areté Systems Corporation, 2040 Hartog Drive, San Jose, CA 95131; phone 408-263-9711.

PARALLEL also had its fault-tolerant computer on display. Although it is not promoted specifically as an on-line transaction system, it should be as relevant there as in other high-reliability applications. The Parallel 300 uses 68010 CPUs running 4.2BSD Unix, and features multiple options for network communications. Parallel Computers, Inc., 3004 Mission Street, Santa Cruz, CA 95060; phone 408-429-1338.

CANTA FORDA COMPUTER introduced its dial-up timesharing service. The computer is a Sun Workstation running 4.2BSD, with 300 and 1200 baud modems, a 1600 bpi tape drive and "an assortment of text and b/w graphics printers." This service is only for users who need at least $250 per month of time and know both computing and the Unix system well enough to need very little support. Canta Forda's other offerings are a computer science library and consulting service. Canta Forda Computer Laboratory, 12326 Hatton Point Road, Fort Washington, MD 20744; phone 301-292-7110; Usenet [ucbvax!allegra!]rlgvax!cfcl1!rdm.

COSI had Link/pc a communications package for the IBM PC. Link/pc makes a PC emulate a VT-100 terminal. As extra features, the PC can dial up a connection through an autodial modem, sessions can be logged, and the whole process can be script driven. Computerized Office Services, Incorporated, 313 North First Street, Ann Arbor, MI 48103; phone 313-665-8778; Telex 466568 COSI CI; Usenet sb1!mb2c!uofm-cv!cosivax.

PPI explained its Objective-C, which adds message/object programming in the Smalltalk-80 style to the C language. (In object-oriented programming, pieces of data have some idea of what they should do and how; programs manage the data at higher levels. If this doesn't tell you too much, PPI offers seminars on object-oriented programming concepts.) Objective-C is a pre-processor, and does not restrict the use of conventional C programming alongside objective modules. Productivity Products, Inc., 37 High Rock Road, Sandy Hook, CT 06482; phone 203-426-1875.

BMDP continues to release improvements for their StatCat, a Callan 68000/Unix box adapted to host BMDP's huge assortment of statistical software. A StatCat gives you full Unix power in conjunction with the stat programs, and adds special I/O management facilities, the j screen editor and the jtalk terminal emulator. Prices for the hardware and Unix System III seem about the same as Callan's direct prices, and annual licensing fees for the statistical packages are on a per-package basis. BMDP Statistical Software Inc., Suite 202, 1964 Westwood Blvd., Los Angeles, CA 90025; phone 213-475-5700.

BENCHMARKING

Back when I was editing Pipes & Filters, there was no topic that would generate more letters than benchmarking. more than a few readers seemed to regard benchmarks as nothing more than an adult equivalent to drag racing. Even the benchmarking enthusiasts could never agree on what tests were worth running, how they should be run, or (least of all) how to interpret the results.

From this I've learned to be cautious when I set about writing on any aspect of benchmarking. Before I drift into a few of my theories, and a report on the latest product in the field, I present the latest benchmark results I've come across, starting at the top of the next page. This piece is reprinted from the European Unix Systems User Group Newsletter, Winter 1983 issue, by virtue of Uni-Ops' reciprocal reprinting arrangement with the EUUG. Where it was published originally, I don't know.

Two programs, many UNIX systems (reprint)

Andrew S. Tanenbaum
Vrije Universiteit, Amsterdam

Teus Hagen
Mathematical Centre, Amsterdam

UNIX meetings gives a splendid opportunity to run test programs on the machines present at the exhibition. At the recent meetings and exhibitions in Europe and the US, we have run two test programs on a wide variety of machines. Test program #1 measures CPU/memory speed; test program #2 measures I/O speed. Program #1 was tested six times, with the 'TYPE' declared in six different ways: short, register short, int, register int, long, register long. On the small machines, the test were generally made in single user mode; on the large mainframes we had the share the machine with other users.

The programs:

```
/* Test 1 - CPU/memory */
main()
{ TYPE i, j, k;
  for (i = 0; i < 1000; i++)
        for (j = 0; j < 1000; j++)
                k = i + j + 1983;
}
```

```
/* Test 2 - I/O */
main()
{ int i, j, n;
         char a[512];
         if ( (n=creat("foo",0755)) < 0)
                perror("error: foo");
  for (i = 0; i < 500; i++)
        write(n, a, 512);
}
```

Notes:

The times reflect a combination of several factors, among them, the CPU type, the clock rate, the speed of the memory management unit, the speed of the memory itself, the width and speed of the bus, and last, but certainly not least, the quality of the C compiler used on the machine. Also, the times were obtained using the time(I) command. There is reason to believe that not all vendors understand that 50 Hz != 60 Hz, which makes some of the times slightly suspect.

Conclusions:

None. You should take these measurements with a grain, or better yet, an imperial gallon, of salt. For example, comparing the PDP-11/70 with the SUN, we see that for test #1 and register short, the PDP-11/70 is nearly three times faster, but comparing register long for the same two machines, the SUN is twice as fast. The difference can be explained by the fact that the PDP-11/70 really is faster, but uses memory instead of registers for register longs, whereas the the SUN uses the 68000's hardware registers.

Goal:

Our goal in making these measurements is to stimulate you into making your own measurements and to make you cautious when looking at (carefully selected) comparisons thoughtfully supplied by vendors. Remember: Figures don't lie, but liars figure. If anyone wants to run the tests on other machines, we would appreciate hearing from you.

times in seconds:		usr	usr	usr	usr	usr	usr	real	real			
machine	**MHz**	**short**	**short**	**int**	**int**	**long**	**long**	**cc#1**	**cc#2**	**#2**	**year**	**rem**
			reg		**reg**		**reg**					
DEC:												
VAX 780		8.8	8.6	6.7	4.7	6.7	4.7	3	3	2	83	
VAX 750		16.2	16.5	10.9	7.3	10.8	7.3	11			84	
VAX 750		18.9	19.3	13.0	8.5	11.6	8.6	4	4	2	83	
MicroVAX		28.2	28.3	22.2	12.4	22.2	12.4	12			84	
VAX 730		37.4	37.5	23.9	14.4	24.1	14.4	11	12	6	83	
11/70		7.4	2.8	7.4	2.8	14.3	14.3	11	12	14	83	
11/60		10.4	4.2	10.4	4.2	20.2	20.2	17	19	16	83	
11/44		11.2	5.7	11.2	5.7	21.8	21.8	16	8	21	83	
11/45		19.1	7.9	19.1	7.8	38.5	38.5	12	20	12	83	
11/34		22.0	10.9	22.0	10.8	44.8	44.8	18	19	14	83	
11/24		27.7	12.2	27.4	12.4	57.0	57.5	21	20	19	83	
AEDS11(/23)		34.5	14.9	34.6	14.9	72.2	72.2	28	30	8	83	
micro 11(/23)		36.8	16.2	36.9	16.2	76.8	76.8	29	31	24	83	
68000's:												
Plexus P/35	12.5	8.9	5.3	10.0	5.1	10.1	5.2	12			84	
Ch Rivers	12.5	10.1	6.0	11.8	5.9	11.8	5.9	12			83	
QU68030	10	12.0	6.0	13.0	6.0	17.0	8.0	8	11	4	83	
Parallel	12.5	11.6	7.0	13.5	6.7	13.5	6.7	17	18	8	83	
Altos	8	13.9	13.9	13.9	13.9	17.8	17.8	18	25	5	83	
SUN 1	10	13.0	7.8	14.6	7.6	14.6	7.5	9	10	5	83	
Cyb	8	15.3	9.0	15.3	8.2	17.5	8.2	27	28	12	83	
Momentum 32E	8	14.5	8.6	16.7	7.6	16.7	7.6	22			83	
TI NU	10	13.8	6.7	18.2	6.5	17.7	6.6	16			84	3 usrs
Codata	8	17.1	10.5	19.3	10.5	19.3	10.5	28	30	28	83	
CCI	8	20.5	9.3	20.5	9.3	29.9	13.2	13			84	
Pacific	10	18.1	10.7	20.7	9.7	20.8	9.7	24	23	13	83	
Power 520	8	21.1	9.2	20.7	9.2	31.4	12.5	16	16	19	83	
Hawk 32/E	8	19.0	11.3	21.5	10.3	21.5	10.3	29	19	27	83	
Pixel 100/AP	8	18.6	11.0	21.5	9.6	22.9	9.6	44	47	10	83	
Corvus	8	19.8	11.6	22.5	10.1	22.6	10.1	40	42	no sp	83	
QU68000	10	24.5	11.9	24.5		33.0	15.1	10	10	7	83	
Fortune	8	21.7	12.8	25.0	12.4	25.0	12.3	18	20	6	83	
Apple Unisoft	5	22.6	13.7	25.9	12.1	25.9	12.1	41	43	28	83	
Apple Xenix	5	22.7	13.9	26.0	13.0	25.9	13.0	58	65	no sp	83	
Altos-12	5	26.7	14.1	26.7	14.1	53.7	53.9	31	33	13	83	
Plessey S68	8	23.9	13.9	27.5	12.8	27.5	12.6	30			83	
Wicat WS150	8	24.8	14.4	28.1	13.1	28.1	13.1	19	21	12	83	
Victory 68K	10	23.7	15.0	28.3	12.6	29.1	12.5	19			83	
TRS80	8	25.2	14.9	28.3	14.0	28.4	14.2	23	28	16	83	
Codata	8	25.5	14.9	29.2	12.8	29.2	12.8	16			84	
Dual	8	26.9	15.6	30.7	13.3	30.7	13.3	21	20	27	83	
Four Phase	8	27.4	16.2	31.1	14.9	31.1	14.9	32			83	
Ch Rivers	8	28.2	15.8	31.7	15.8	31.7	15.8	28	42	14	83	
Unistar 200	8	28.7	16.5	32.8	14.3	32.8	14.3	36	40	28	83	
IBM Sritek	8	30.3	17.6	34.2	17.3	35.1	16.9	31	35	18	83	ut>rt?
IBM PC Idris	8	37.9	22.7	37.9	22.7	73.3	73.3	18	26	39	83	
Cosmos Antaris	8	34.1	19.5	38.7	16.4	38.7	16.4	26	27	9	83	
ULAB	8	37.2	21.0	44.1	18.9	44.1	19.0	38	44	15	83	

times in seconds:		usr	usr	usr	usr	usr	usr	real	real			
machine	**MHz**	**short**	**short**	**int**	**int**	**long**	**long**	**cc#1**	**cc#2**	**#2**	**year**	**rem**
			reg		**reg**		**reg**					
Z8000's:												
Z6000	5.5	13.6	6.3	13.6	6.2	23.2	12.6	20			83	
Zilog	6	14.7	7.3	14.7		25.7	13.3	20	20	8	83	
Plexus	5	15.2	7.0	15.4	7.0	27.5	27.6	24	26	13	83	
ONYX	4	15.9	7.2	15.9	7.3	23.8	14.1	14	14	8	83	
Bleasdale	4	33.3	15.6	33.3		56.2	56.2	20	21	16	83	
8086's:												
Altos	10	13.7	7.2	13.7	7.2	27.8	27.7	18	20	7	83	
Intel	8	29.2	17.6	29.0	17.5	59.1	59.0	17			84	
SBC 86/12A			68.1	83								
8088's:												
IBM XT	4	53.4	30.0	53.4	30.0	108.8	108.8	25			84	
286's:												
Intel	5.5	17.3	10.9	17.3	10.7	34.6	34.6	27			84	
16032's:												
National	6	27.4	25.6	29.3	14.0	32.4	11.7	20			83	
IBM Sritek	6	27.4	27.5	32.1	13.4	30.7	13.6	103			84	
National	4	49.9	45.0	56.7	23.3	57.4	27.2	47	49		83	mem flt
others:												
Amdahl		0.5	0.5	0.3	0.3	0.3	0.3	10	5	1	84	1 usr
Concept32/87		1.5	1.5	0.9	1.4	1.0	1.4	10	16	2	83	1 usr
Pyramid		3.3	3.3	2.0	1.9	2.0	2.0	9			84	
Ridge		7.1	2.9	4.0	1.6	4.0	1.6	19			84	
Eclipse		4.6	4.6	4.4	4.4	4.4	4.4	28			84	
Arete		5.7	5.7	5.7	5.7	5.6	5.8	15			84	
HP 9000		9.4	9.4	7.4	7.4	7.4	7.4	28	26	3	83	
Concept32/27		12.0	11.0	10.0	10.0	10.0	10.0	18	20	5	83	
BBN C/60		14.5	8.2	14.6	8.2	47.2	30.3	7	9	3	83	
PE3210		16.7	6.7	15.9		15.9	6.7	4	4	3	83	
Perq 1		44.5	15.6	22.2			15.0	25	25	7	83	
Perq 2		46.7	15.1	23.0	15.0	22.9	15.1	20			83	
IBM S/1 4954		37.2	37.1	37.1	37.1	62.3	62.3	28	30	32	83	

Brand-new Unix users who want to try these benchmark programs on their own need a few instructions. You start by using any editor on the system you're testing to create a text file that contains either one of the benchmarks exactly as it's shown, except that in the first program, the word TYPE must be replaced by one of the six types named in the first paragraph. Give your text file a name that ends in .c (for instance, test1.c will do). Then, the two command lines:

```
time cc -O test1.c
time a.out
```

will tell the system to calculate and display the information Andy and Teus put in their charts; and a lot of other figures, too. If you want to compare results on the same system using different compilers, find out the command-line names of any other compilers on the system (not the trademark names; what you want will be only two or three characters long and all lower case) and substitute one of those for cc when you rerun the two command lines above. Use only C compilers, of course.

IS BENCHMARKING REAL?

At the very least, benchmarking is not a well-developed science. Andy and Teus aren't the only ones who want to take more than a grain of salt with benchmark results. A few of the more cogent reasons for distrusting Unix benchmarks:

¶ It's fairly easy to design a benchmark to test systems that will be doing just one thing all the time: processing credit card slips or running fast Fourier transforms. But Unix systems are almost always sold for general purpose use, which is the real forte of the Unix system, and there's no agreement on which performance specs denote a good general system.

¶ Ordinary benchmarks exercise both the hardware and the operating system, and to be transportable between various Unix systems benchmarks must be written in C and compiled on each system under test. The results of this sort of benchmarking necessarily blur the combined capabilities of the hardware, the Unix implementation and the C compiler.

¶ The huge range of Unix system sizes makes benchmarking over the full size range highly dubious. For instance, on a supermini intended to handle dozens of users, scheduling and swapping will be important factors; on a network of workstations being considered for the same application, swapping and scheduling will be minor factors and network communications will come to the fore.

¶ The Unix clock is not particularly precise. Basically, each cycle of the power line generates an interrupt that increments a counter, so timing accuracy on short intervals depends on how well those interrupts are handled. Making the benchmarks run longer avoids that problem at the expense of worsening the size scale factor mentioned above. That is, a benchmark test that runs long enough on an Amdahl to give a meaningful result would take all afternoon on an IBM PC XT.

¶ Any knowledge the user does have of his requirements is not usually taken into account in benchmark suites - no advice offered on which tests might be particularly applicable to software development, say, or secretarial work, much less on relative weights.

This list could go on, of course; I suspect that most of you have pet gripes of your own about benchmarking. The point to note here is that none of these problems is really inevitable. A little work on them could yield a lot of improvement, and with the Unix boom here at last, the demand for useful benchmarks is probably here, too.

AIM BENCHMARKS - SUITE II

The only commercial, supported benchmark package to date now has a younger brother. The differences start with purpose. The original benchmarks were a tool for technical experts, intended as much for system tuning as for purchase decisions. Suite II is a managerial tool: easy for low-tech people to run, giving bottom-line results instead of obscure technical data, and slanted strongly toward "which system to buy?" decisions.

Before I go into the details, I must make a disclaimer. I've been watching Suite II develop from raw ideas to finished product over the past eight months, and I can never look at it disinterestedly now. I will try to be objective, though.

Suite II is written in C, and the first step in testing any system is to compile Suite II on it. The programs are set to compile themselves using cc; it's possible to change to another compiler if you have an editor handy and know what to look for, but the non-technical user is effectively limited to cc.

There are thirty-six individual tests, ranging from hardware-intensive operations in the memory to nearly-pure-Unix tests such as getting the process ID. A single command runs them all in series, and the whole testing process takes about ten minutes. Running time does not vary much with system power because Suite II uses autoranging - the number of repetitions in each test is adjusted on the fly. The first test is run at the smallest number of repetitions and, when it is complete, the program checks whether the resulting timing is long enough to be meaningful; if not, the test is run again with repetitions increased by a suitable power of two. Subsequent test lengths are multiplied by that same power of two; the program keeps examining result times, but usually no more repetition changes are needed.

The numbers generated by these tests are not normally spewed out on the console or the lineprinter. They go into a little database that's supplied with Suite II, along with figures for about half a dozen representative systems of different sizes that are supplied with Suite II and with figures from any previous systems the user has collected. The user collects these figures into a single database by tarring the database back to tape after each benchmark. Given the vagaries of machines under test, it's obviously necessary to back up the database after each benchmark. This database can hold results from more than twenty systems, I'm told.

Now the fun begins. Suite II has another set of programs that bend over backwards to help non-technical users decide what the raw results mean in terms of their own needs. First, the programs ask the user what he expects his workload mix to be, using classes like software development and word processing. (As an example of how thoroughly Suite II is optimized for the managerial rather than technical user, it asks for the workload mix in percentages, but does not bomb or even protest if the percentages given don't add up to 100%.) It also asks for the price of each system that's been tested. Now Suite II is ready to tell the user about relative system performance: in raw form, in terms of the user's expected workload, compared to prices, or as a combination of the latter two. These comparisons come out both as tabular charts and as graphs; the latter in the form of kiviat diagrams, which look like deformed starfish and work like line graphs rolled into circles. Because the database is permanent, these analysis runs can be repeated with parameter changes as often as desired. Even the relative weightings of the 36 test factors with respect to the workload classes are not sunk in cement. Anyone technical enough to have his own ideas on this subject will easily be able to go into the file with any Unix editor and change the weightings to correspond to his ideas.

There you have a serious effort to turn benchmarking into an automated management tool - a first in the Unix field, and the first such in any type of computing that I'm aware of. I frankly think the Aim Suite II benchmarks are a major step toward making Unix palatable to large organization executives who up to now have always bought what IBM (or Dec or Data General) offered because that was obviously the safe route through a thicket of technical issues they couldn't begin to understand. If your thinking is at the opposite pole - that equating certain test characteristics to real-world performance is arrogant presumption, you have an ally in Gene Dronek, who wrote both suites of the Aim benchmarks. Gene spent the first thirteen years of his computing career at a university computing center, where he acquired the usual academic reluctance to make a definite statement on anything as complex and controversial as system performance. It took two of us several months to get him to design the analysis part of the package, called Select, and he still makes a point of how easily anyone can rewrite the weighting factors to suit his own ideas. Actually, both Aim benchmark suites got started rather indirectly. Suite I was originally developed as a marketing tool for Aim's hardware line (since discontinued), and Suite II was an outgrowth of Suite I customers' continuing requests for help in interpreting the results of the tests.

Suite II is designed to run on all releases of Unix and on most of the workalikes. It comes on 9-track tape in tar format. The current price for a tape is $2575, but that's after deducting a 25% introductory discount, and there's no word on how long this discount will last. A free copy of the Suite II user manual is available if you answer a short questionnaire about your potential use for benchmarks. Aim Technology, Suite 390, 4655 Old Ironsides Drive, Santa Clara, CA 95050; phone 408-727-3711.

LOCAL USER MEETINGS

National conferences are fun, and NetNews debates can be exciting, but neither is a satisfactory substitute for local get-togethers in your own metropolitan area. Unix is more real, somehow, after an hour or two of discussing who's going with the new start-up on the other side of town, what's happening with Sam's new vertical-market package, whether the city council will be able to slap a tax on disk reads, and whether the most recent Unix release is working for people you know. Herewith, a listing of some groups that hold local meetings with some regularity, and (for those who aren't near any of them) a few suggestions from experience on starting a local group.

SACRAMENTO Started last January, usually meets the first Wednesday evening of the month at the SMUD training building, east of central Sacramento. Meetings center on a speaker. No charge to attend, nor to receive meeting notices. Bill King, 916-927-5158.

DIABLO VALLEY This area is halfway between Sacramento and San Francisco. It had one of the earliest local groups, which lapsed about a year ago but has recently been revived. Lunchtime meetings take place on the second Thursday of each odd-numbered month - restaurant varies. Meetings are mostly conversation, with a five- or ten-minute talk at the end. The only cost is whatever you order for lunch. Typical turnout is 15 to 20 people. Chuck Phillips, 415-932-6900.

SILICON VALLEY Evening meetings at the Hewlett-Packard auditorium in Palo Alto once a month. Meetings center around a featured speaker of the evening, although many people arrive early or leave (very) late in order to talk with other attendees in the lobby before or after. These meetings are in their third year, there's no charge to attend or to receive notices, and the turnout averages 65. John Bass, P. O. Box 251, Menlo Park, CA 94026-0331; phone 408-996-0557.

There is another, brand-new group also meeting in Palo Alto. These folks are new to Unix and red-hot to get themselves up to speed. They meet on Saturday afternoons and listen to intense tutorials; solid computing background and usually a college education are required to join their group. Membership fee is $20 per year, and their are extra costs for anyone who participates in certain specials. Unix/C Club, P. O. Box 32784, San Jose, CA 95152. Gilbert Dalit, 415-854-3300 x 2034 days, 408-238-5833 evenings.

LOS ANGELES The Los Angeles Unix User's Group meets one weekday morning each month at TRW in Redondo Beach to hear a speaker or panel. The expertise level is fairly high, and (as you'd expect given the meeting time) attendees are largely from big computer facilities within big organizations. But everyone is welcome, and there is no charge. Dale Hensley, 213-535-1314.

SAN DIEGO A group called UNIX/C GIG (the last stands for General Interest Group as differentiated from Special Interest Group) meets the third Wednesday of each month at Epicom, Inc. Meetings feature a product demonstration or a middle-level technical talk, and a Unix/C bookstore. Typical attendance is around 20 people. Dave Haldy, 7844 Convoy Court, San Diego, CA 92111; phone 619-292-5690.

HOUSTON A fairly new group called Unix Houston has a dinner meeting preceded by a cocktail hour the third Tuesday of each month. The dinners cost about $11; at present there's no other charge for being involved. Brenda Sedgwick, Suite 505, 5645 Hillcroft, Houston, TX 77036; phone 713-556-8573.

DALLAS A group is just starting here. I have not been in touch with them personally, but Brenda Sedgwick of the Houston group knows them.

MINNEAPOLIS The Minneapolis Unix Users Group is also new; they meet the first Wednesday evening of each month. So far, meetings have presented local people talking about local activities, but major speakers from outside the area are under consideration. Twenty to thirty people per evening is typical right now; the major Unix users in the area are regulars, but the small outfits with desktop Unix systems are just finding out that the group exists. No dues or fees, although they may ask for a small voluntary contribution from each regular to pay the cost of mailing the monthly notices. Scott Bowman, 612-894-7752.

WASHINGTON, DC This is the oldest local group by far that I know of. Their evening meetings have well-thought-out topics and attendance is typically over 70, I'm told. No charge to attend. Rick Wilder, Calculon Corporation, 1301 Piccard Drive, Rockville, MD 20850.

NEW YORK CITY The most ambitious of local Unix user groups is Unigroup of New York, Inc. Aside from local meetings, they publish a bimonthly newsletter of about a dozen pages per issue, and are working on a local Unix trade show. All this is expensive: dues are $35 per year, and nonmembers must pay $10 to attend one of meetings. Both the meetings (usually held in midtown Manhattan) and the newsletter, Unigroup Journal, have a lot of substance to them. Unigroup of New York, Inc., GPO Box 1931, New York, NY 10116.

BOSTON At last I've gotten some information on the Flying Dutchman of local groups, New England Unix User Group. Unigroup of New York tells me they meet the second Tuesday of each month in one of the high-tech suburbs around Boston, and that most of their meetings have product demonstrations. Sheldon Laube, Consumer Financial Institute, 430 Lexington Street, Newton, MA 02166; phone 617-969-2632.

LONDON Beer and dinner meetings at the Lyric Pub the last Thursday of each month. Meetings feature Unix conviviality rather than organized presentations. As of last fall, meetings started at 5:30 pm and didn't completely break up 'til 11:00 or so. The Lyric is on Great Windmill Street off Shaftesbury Avenue, near Piccadilly Circus. For the latest details, contact Mike Banahan, last seen at Taurus Technology. If you can't find Mike, phone Mrs. Helen Gibbons (Secretary of the European Unix Systems User Group) at 763-71209 or 763-73039 (both in England) for suggestions on contacting the group.

[Sorry; ran out of space. Continued next month.]

U-NEWS is published monthly by Uni-Ops, a non-profit association of Unix users, at P.O. Box 27097, Concord, CA 94527-0097. Editor is Walter Zintz, telephone 415-945-0448.