

AUSTRALIAN UNIX USERS GROUP NEWSLETTER

CONTENTS

Editorial	1
L7 at EECS UNSW - Paper	3
UNIX Numerology	13
AUUGN Accounts	15
SUN Netmail	45
US Meeting Announcement	48



---

### In This Issue

---

True to his word, Kevin Hill has produced a huge summary of what he said in Queensland, as well as quite a bit he didn't. I have not yet produced my summary. Oh well.

---

### My Rave

---

As this is the last issue of AUUGN that I shall edit, at least in the foreseeable future, I shall use this opportunity to say a few things.

The last two years as editor have been fun, in much the same way as going out on the town every single night is fun. It's fun while you do it, but you start to realise that you don't look forward to it after a while. Still I have had the chance to meet and correspond with some interesting people. I have also been able to read some of the more interesting stuff I have published (and some I have not), though there is no way I have had time to read it all.

You will see from the statements of account reproduced in this issue that AUUGN is, as I hand over to the next editors, financially in a pleasingly healthy condition. As partial justification for the subscription rate rise, I should point out that the financial surplus is only due to the vast amounts of voluntary work that have been put into the newsletter by Kevin Hill, Robert Hegedus, Perry Brown, Paul Gillis and myself. The School of Electrical Engineering and Computer Science has also provided photocopying at cost and the services of some part time student assistants ("indians"). I must also thank Kerry, an indian with remarkable skills in english and spelling, who has helped in proof reading, although she could not weed out every error.

During the "World UNIX Meeting" in Melbourne in October 1980, I took part in a panel discussion on "UNIX, where to now?". At that time I said that I was not looking forward to the next few years as the use of UNIX became much more wide spread, running on numerous manufacturers CPUs in many different flavours. A year later I am still at least as pessimistic, if not more so.

Don't think that I am unhappy to see so many more people able to use a system that I have grown to know and love, whatever variant they choose or are forced to use. I am simply saddened by the changes occurring in what appears to be a completely random way. No standards have been developed or adopted in any area and although there is now talk of developing some sort of standard, everyone is still coding away in random directions.

Gone are the days when most UNIX systems, kernels and support programs, were pretty much the same. Getting a tape from some far removed site now often involves massive effort even to read the tape, let alone compile (with whatever version of the C compiler you have) the result.

Also gone are the days of "small is beautiful" to be replaced by "bigger is better". Obviously small systems (usually on hardware constrained machines)

nave their problems and graduating to a machine with a large address space is a nice thing. But I fail to see why the availability of more space often promotes the filling of that space with sloppily written monoliths to some programmers' ineptitude.

Cough, foam, froth, ok I have had my little rave..... Best wishes to all our readers for the festive season.

---

#### The New AUUGN Editors

---

I now welcome Bob Kummerfeld and Chris Rowles, the new AUUGN editors. The new subscription price is (Australian)\$24-00. Send subscriptions and contributions to Bob at the address below.

Bob Kummerfeld  
Basser Dept. of Computer Science  
Madsen Building F09  
University of Sydney  
Sydney 2006  
N.S.W.  
AUSTRALIA

---

#### The Art Work

---

After last issue's individually drawn, multi colour abstract I feel that another artwork would be an anti-climax. I wish to thank Perry Brown, Graham Smith and Uncle Kev for donations to the cause.

Peter Ivanov  
Dept. of Computer Science  
Electrical Engineering  
University of N.S.W.  
PO Box 1  
Kensington 2033  
AUSTRALIA

(02) 662-3781

Level 7 UNIX  
at  
Electrical Engineering and Computer Science,  
University of New South Wales

Kevin Hill

This paper is intended to cover some of the points raised in my talk at the last Australian Unix Users Group Meeting in Brisbane. I spoke briefly then about some of the changes that had been made to UNIX level 7 at Electrical Engineering, in order to support our teaching load, to preserve level 6 "features" that we had become used to, and to increase compatibility between our PDP-11/70 running level 7 and our VAX-11/780 running a mixture of UNIX 32V and Berkeley 3.0. These changes are far from finished, but it was felt that there may be some interest from other sites who have yet to make the changeover from level 6 to level 7. I hasten to add that I am not the only one making these changes, and that most of the ideas for the changes are in no way original. I was simply volunteered to give the talk.

At the School of Electrical Engineering and Computer Science within UNSW, we have an 11/70 with 640Kb core, two RP04 88Mb disk drives, a TEL6 tape drive, and 40 DZ terminal lines, to support a user population of 600, a batch link to a cyber, and network traffic. Almost all of our work has been done on this system. Unfortunately, UNIX level 7 (from Bell) does not run immediately on machines without separate I and D space capability, such as the vast majority of machines running level 6 UNIX locally. This is because the system itself, and a lot of the commands, are quite large, and are severely constrained in the PDP's 16-bit address space. Adopting the "I'm all right Jack" philosophy, our changes have not been primarily aimed at reducing the size of the system for small machines (because we have a large one). In fact only very recently have we even attempted to get our 11/70 system running on a PDP-11/40 with 240Kb core and 3 RK05 2.5Mb disk drives. Although this system is working, those sites wishing to run level 7 on small machines are nevertheless referred to people such as Rick Stevenson at the University of Queensland, or Craig McGregor of the Faculty of Architecture in UNSW, both of whom have spent a lot more time on squeezing level 7 into small machines. We have even at times had serious problems fitting our system onto the 70, and have (for example) found it necessary to address the buffers through one segmentation register, and to keep the block addresses packed into 3 bytes per long in the in-core inodes.

Why change over from level 6 to level 7? There are certainly several minuses:

- \* Dedicated AUUGN readers may have noticed some less than complementary comments of mine in the previous issue of AUUGN on two of the commands provided under level 7 (mail, cpio). Similar comments could easily be levelled against other commands i.e. too much code with too little thought.
- \* Our experience at UNSW is that level 7 can only support about 2/3 the number of people that our level 6 system could support (this is discussed again below).
- \* Level 7 (and 32V to an even greater extent) appears to be adopting the "big is beautiful" solution to problems; "simplicity" and "elegance" appear to be dying terms; problems are being solved by throwing a few more megabytes

or a few more disk drives at them. This is probably understandable given the way prices have come down (especially from non-DEC sources), but there are still quite a lot of small machines and even smaller budgets around.

\* A lot of the fixes and genuine improvements made at UNSW (and elsewhere) to the system and the commands are still missing. Some examples of this are:

- when unmounting a disk, there would still be blocks in core waiting to write all over the next disk mounted with the same system device number;
- the test in "physio" to make sure a raw i/o area was wholly within a process's address space was still wrong (although wrong in the right direction - it would sometimes needlessly refuse an i/o that could have been done);
- superblocks were still stored in special places; this meant they could not be reliably accessed through the buffer pool;
- the "mv" command was still a pile of worms (e.g. "mv a b" where "b" was a directory would be interpreted as "mv a b/a" which is great; if "b/a" existed it was unlinked first, which is also great; but if "b/a" was an existing directory, then simply unlinking it from a great height corrupts the file system!);
- it is still possible to create orphan leaves in the filesystem (i.e. files without directory entries).

\* Some dubious decisions have been made (in our opinion) - for example:

- the command interpreter (shell) prompt has been needlessly changed from "%" to "\$" - a petty point, but an equally petty change;
- the shell sets up pipelines with a totally ridiculous parent/child relationship (the last command is set up as the parent of all the others); one minor result is that if such a pipeline is started asynchronously, only the process id number of the last command is reported; more significantly, the last command may not expect (and may not be able to cope with) its children-in-law;
- ordinary users were allowed to change the access and modify times of files to anything at all (may as well just lump them together and call them "fantasy time");
- the first inode (number one), which used to be the file system root directory inode, is now reserved for a bad block file - inode two is the root inode (a better scheme was mentioned by Dave Horsfall in the last AUUGN); whereas this change may indicate how easily fairly fundamental constants may be changed in a well-written program, it was totally unnecessary;
- the system now partially recognises the "." directory name, to enable one to change directory off a mounted file system. Whereas the idea is noble, it could be done in many better ways rather than casting "." in bronze in the system (the system should only "know" about two directories: "/" and "", being the root directory and the current directory, respectively; everything else is simply a perturbation of the filesystem);

- mail is now collected in a common directory `"/usr/mail"`. The only thing this achieves is to run your `/usr` disk out of space and to give the system administrator one more place to look in when moving users around. We have changed it back to `".mail"` in the person's login directory.
- \* The level 7 distribution appears to have been put together in a very hasty manner. It is obvious that rather than being a finished, polished, product, level 7 is very much just a snapshot of a working system. People who talk about a "level 7 standard" are simply having themselves on. It is in fact quite instructive to compare the Bell 32V release with the level 7 release - plainly, a lot more care and attention (and probably time) has gone into 32V. Some of the observed shortcomings are:
  - handles are provided for parts that are obviously missing or only partially completed (although the system will still run OK);
  - several facilities are obviously experimental or are provided with the warning that they are "not considered a permanent part of the system" (e.g. `phys` - allows a process to access physical addresses, `lock - lock` a process in primary memory);
  - `adb` (the only debugger available under level 7) is supplied with a known bug of not being able to print out floating point numbers in core dumps or when tracing another process (this bug was actually easy to fix; it was only due to some sloppy coding);
  - a much worse (and unforgivable) problem with `adb` is that it cannot patch single bytes;
  - the system DZ driver was supplied with the confidence-inspiring comment at the top "CAUTION -- MODIFIED FROM WORKING VERSION BUT NEVER PROPERLY TESTED";
  - there was a lot of unnecessary code in the system-support file `"mch.s"` to do with system startup that was now being done by the secondary bootstrap;
  - there are times when a process switch would occur without having saved the floating point registers (so that when you were resumed your registers would contain someone else's junk);
  - registers 0 and 1 and the processor status word were not being cleaned up properly when a new process was born (it would inherit rubbish from its parent);
  - device drivers were not passed consistent arguments to do with how they were opened at all points in the system;
  - the assembler "backup" routine (to back up failed instructions) would still not work on some processor types (see article by Jim McKie on page 73 of AUUGN vol 2 nr 3);
  - the system did not make correct use of the assembler `"rti"` and `"rtt"` instructions when returning from interrupts/traps; one result of this was that the debugger would sometimes skip instructions during single-stepping (which uses T-bit trace traps);

- the system did not allow total recovery from floating-point exceptions
  - the floating point error register is saved in the u-area, and it can thus be examined in a core dump or when tracing another process, but the process itself cannot access it;
- whereas some care had been taken to properly swap in and out maximum-sized processes, they still would not core-dump correctly (you get only the system per-process data area);
- system compilation still depends on a lot of horrible shell procedure files; very little use is made of "make" (which still has many of its own problems).

Some mention should also be made of a disturbing tendency observed in level 7. Both `adb` and the shell are written in a style that can at best be called "unique". They make heavy use of the C-preprocessor by defining (for example) "BEGIN" to be "{", "END" to be "}", "IF" to be "if(", "THEN" to be "){", etc., in an attempt to make the source appear to be some sort of cross between wounded `algot` and a road accident. This foolishness simply does not help the readability or understandability of the program, and should never have been allowed. In such critical system programs, we expect only crystal-clear coding. Both have been converted back to C at UNSW.

Nevertheless, for all its faults, there are some very compelling reasons for adopting level 7:

- \* The problems mentioned above, although annoying, can be fixed (and in most cases we have done so). Beauty is very much in the eye of the beholder - a style one person thinks is terrific will be loathed by another. Some things are bigger simply because the problems they attempt to solve are more ambitious than before. Our own 70 system currently contains many handles for things not yet implemented, and we know full well ourselves how much time and effort it takes to put together a software distribution. After all, Bell did not have to release level 7; we are grateful for what we got at the price we got it for.
- \* There is no level 6 for VAXes; 32V (the VAX system) is almost the same as level 7 (except for the machine dependent parts). Hence for maximum compatibility with VAX UNIX sites, as well as all the other PDP sites in the world that are already running level 7, it must be seriously considered. It is this very problem of compatibility that has forced us to get our 70 system running on our 11/40 (and by the end of the year on our 11/34).
- \* Some areas of level 7 have obviously received a lot of hard work and inspired thought. Examples include:
  - system support for shared-text programs has been done much better than in level 6;
  - the new "save/resume" mechanism is superior to, and much clearer than, the level 6 "savu/retu/aretu";
  - the new "ioctl" system call allows far greater flexibility than "stty/gtty";
  - the whole system-call mechanism has been made far cleaner;



- Adb, although having a lot of its own problems, leaves db, cdb, and all the other level 6 variants, for dead. As well as providing normal facilities such as C-stack analysis and allowing for single-stepping and breakpointing of a program, it allows one to write small templates with which to explode system tables. This makes debugging the system much easier.
  - The file system is a lot more tolerant of system crashes - under level 6 we would sometimes spend hours putting a disk back together. Under level 7 we have had several crashes (some software, but mainly parity errors in our cheap memory) and have ended up with only minor file system errors.
  - A lot of programs previously written in assembler have been rewritten in C, increasing their portability and maintainability.
  - an earnest attempt has been made to identify machine-dependent parts and to collect them in identified places.
- \* Level 7 is (with a few exceptions) a very "tidy" system of high standard; our local level 6 system could only be described as an absolute mess. Level 7 is helped in this area by having a much better C compiler that allows type-casts (previously you could only stand back and wonder what the compiler had done to your mixed-mode expressions; whatever it was it was guaranteed to be not what you had intended). Those sites planning a more gradual changeover to level 7 are strongly advised to get a level 7 C compiler and standard i/o library up on their level 6 systems (we have them on our smaller systems in order to compile and run the things we are now writing in level 7 C). Lint has proven to be an invaluable tool for cleaning up messes and enforcing some sort of discipline on C code. The initial system booting process is also a much cleaner process now; for an account of the contortions done by level 6 see my article on page 76 of AUUGN vol 2 nr 4.

We started work on level 7 at the AGSM (in UNSW) on their PDP-11/70 just over a year ago (at the time we had non-standard disks on our 70, and could not access level 7 filesystems from a level 6 system). Although we had no trouble at all getting it up and going, certain differences were immediately obvious. Going back to a standard Bell editor from our heavily modified QMC editor (em) was painful to say the least. Lack of a multi-column output filter (mc) was immediately obvious, especially as we had to work (initially) on a decwriter console (until we got the DZ driver working). A carriage-return line-feed pair bubbling up from the bowels of the system just because the kill character had been entered (even after a slosh) was particularly annoying, especially on a slow decwriter. The reason this was done is fairly obviously to show a clear line again after typing the kill character. However, this does not help if you type erase characters back over your kill character - you don't suddenly jump back up to the previous line again. I should add here that the capability of erasing an accidentally typed kill character is one of those many local level 6 "features" that we had come to know and love, and that was added to level 7 again quite quickly. Going back to "icheck", "dcheck", and "ncheck", from our "chk" (modified from the original from Vrije University, Holland) was a giant leap backwards.

Another local "feature" that was missed immediately was the magnetic tape (HT) interface. Instead of rewinding, or spacing to the next Tape Mark, when closed after reading, or writing a Tape Mark when closed after writing, it would simply stop where it had got up to. This meant that you actually had to "read"

the Tape Mark, and deliberately write them when necessary to separate files on the tape. It was disappointing to see that one still could not write a tape block containing an odd number of bytes. This problem is partially due to the DEC hardware (one has to load a word count into the RH70 which implies an even number of bytes, and a frame count into the TM02 tape formatter which is effectively in bytes and may be even or odd; one then has to ignore the Frame Count Error interrupt that this may cause). It was also disappointing to see that no good tape-manipulating program came with level 7. Under level 6, nothing even came close to Ian Johnstone's "dtp". Under level 7 we will have to write our own again.

Apart from fixing some of the above things up, the main work at the AGSM was to get a system running that knew about our (then) non-standard unibus disks. This involved adding code to both the system and to the secondary bootstrap to set up and correctly maintain the 11/70's unibus map, to allow one or more unibus DMA devices to be active simultaneously. Needless to say the system did not work when brought down from the AGSM and tried on our 11/70 for the first time. However, by working out what blocks constituted the file "unix" on the level 7 disk, and by using our level 7 C compiler under level 6, we were able to recompile systems and copy the individual blocks across to the level 7 system! Fortunately, this was only necessary a small number of times, before the problem was found to be a race condition in the secondary bootstrap disk driver (the 70 was too fast for our microprocessor-driven disk controller; it would say "go" to it and interrogate the "finished" bit before the micro had even lowered it in response to the command). It is interesting to note that the primary bootstrap, written in assembler (the secondary bootstrap is in C), exhibited this problem much less frequently.

We committed ourselves to running level 7 full time around Christmas time, 1980, and have only run our level 6 system on a handful of occasions since, all to do with accessing certain database files (the database manipulation package has since been rewritten to run under level 7). Conversion of the file systems presented a minor problem (under level 7 the inodes are twice the size, and block addresses are stored as either 3 byte quantities in the inodes or 4 byte longs in the indirect blocks). Initially we used cpio, but this took so long that it seemed it would take all day and so it was killed. We then tried a version of "dtp" that was running under level 6 and level 7. This failed because dtp does not keep a record of links between files, and so we ran out of space. Thus we were forced back to using cpio, which eventually did the job over a weekend.

Not long into our teaching session, it became clear that level 7 was simply not going to handle the load. Whereas we used to run 45 people at reasonable response, under level 7 that number was reduced to 25. There were many reasons for this:

- \* We observed a scheduling bug very early on (that we have not yet had time to fix) that meant that the CPU would (under certain conditions) execute one process to completion before executing another.
- \* The DZ driver would service one DZ per interrupt, rather than all DZs. This meant there was a very high interrupt load from our 40 DZ lines (most of which run at 2400 baud, but some run at up to 19.2Kb).
- \* The DZ driver has also since been altered to vastly reduce its use of timeouts, which are now grabbed from the c-lists anyway to avoid the extra timeout array.

- \* The number of buffers was inadequate (29 on the Bell system; this is now 80 using a single segmentation register and a hashed lookup scheme).
- \* The assembler routines "copyseg" and "clearseg" have been altered to use the kernel address space, rather than supervisor space (on the 70; user on the 40) and mfpd/mtpd instructions.
- \* A lot of unnecessary disk i/o has been eliminated by the use of bdwrite instead of bawrite during execs and simply throwing these blocks away when finished (up to 10 blocks worth of arguments are stored in buffers attached to the swap device). Similarly, blocks belonging to pipes are simply thrown away when read. The result of these two changes is that we can now run for quite some time with the disks write-protected! A related change involved marking those buffers that required no i/o and putting them at the head of the free list when freed; delayed-write blocks were allowed to accumulate at the end of the free list.
- \* The pipe strategy has been altered to implement true circular pipes, significantly increasing their throughput.

Many minor changes have also been made, not necessarily directed at improving response times. Examples that leap to mind are:

- \* In-core limits structures (lnodes) have been implemented on a per user basis to control disk space usage and number of processes per user.
- \* Our locked file scheme (n-readers, one writer) has been implemented for certain files so marked (e.g. the password file, as well as being hashed for speedy access for our 600 entries, may also be locked so that only one writer is active at a time).
- \* Our ttyconnect system call that allows for the logical connection of one tty line to another, and our multiplexed line driver (mx), have been implemented, so that (for example) we presently have 16 virtual lines multiplexed over one physical DZ line (running at 19.2Kb) between our 11/70 and our VAX. People are able to use these pseudo-lines to the other machine as though they were connected directly to it. Our mx scheme is quite distinct from the level 7 MPX multiplexed files, which have been removed until we figure out what to do with them.
- \* We have implemented the Stack Limit Register to make sure the kernel stack does not collide with the user structure (core dump analyses shows that the kernel stack can grow to about 500 bytes).
- \* Two extra signals have been added for CPU time limits and memory parity errors (which used to tell processes that they had had a bus error!). This meant that some 16-bit shorts had to be replaced by 32-bit longs.
- \* The "iomove" procedure was altered to be smarter about odd-byte transfers (that happen quite frequently on our system). The previous trend was to put such code in the various device drivers.
- \* Reads and writes on the memory special files (kmem and mem) were made vastly more efficient by using iomove and mapping the area directly into the kernel address space (using the buffer segmentation register). Byte operations are also possible (previously these did behind-the-scenes word operations which could be fatal for some devices).

- \* An extra system call was added (one of several) to return certain system table addresses and sizes; this meant that programs such as "ps" no longer had to depend on "/unix" being the current system, and dive through its symbol table for addresses. These last two items, plus a rewrite of "ps" anyway, have sped it up significantly over the original.
- \* Code has been added to "mfree" to detect map overflow (previously this just wrote all over the following data area). This has in fact caused us to crash at least once.
- \* We have deleted "dup2" and added the VAX's much more flexible "fcntl" system call (which does the equivalent as one of its many tricks).
- \* The "open" system call was altered, to bring it into line with that on the VAX, which allowed far greater flexibility (such as allowing exclusive opens and opens that actually creates). This involves passing an optional third argument to the system. For maximum compatibility with old level 6 programs, and those written in assembler, this argument is passed in register 0.
- \* A new batch system was written to grunt to our central-site cyber. This version does not use the previous UNSW shared-data scheme (which will not be re-implemented in that form under level 7), and is about 1/4 the size of our old level 6 one (I am forced to admit that it does not yet have a convenient queue display; however, it is reliable, primarily because it "trusts nothing the cyber says").
- \* The C-compiler and the assembler were altered to create unique temporary file names using process id numbers; previously the superuser could not compile more than one thing at a time.
- \* The "unlink" system call has been changed so that you now need to own either the file, or the directory containing it, or be the superuser, as well as all the normal tests. This prevents cretins unlinking your compiler temporaries in the "/tmp" directory (which has general read/write/search permission) and replacing them with their own (yes, this has happened here at UNSW under level 6, and did result in some students becoming superusers).
- \* The default setting of our FP11-C floating-point unit has been altered to trap on overflow, use of undefined variables, and on integer conversion error. This was considered better than just carrying on blindly with rubbish. We are amused to see the occasional C-compiler core dump now (implying that the code it was previously generating for the program it just core-dumped on was worthless).
- \* The password encryption algorithm has been modified slightly. A lot of work has gone into the new encryption algorithm at Bell, with the result that an attack based on the exhaustive-search principle is totally impracticable. However, presumably because of this, the algorithm would only bother looking at the first 8 characters of your password. One of only two (known) level 7 security breaches occurred when the "bad guys" managed to memorise the first 5 characters of our superuser password, and exhaustively find the last (effective) three (however, being novices, they were found from the tank-tracks they had left behind). Our encryption algorithm now looks at all characters of a password (our current superuser password is 50 characters long, spread over 10 words that make up a rather meaningless sentence). On a semi-related matter, we do have our login

process return "Who?" if it does not recognise the login name. The convenience factor here far outweighs any potential security problems - I personally thought that one of the nicest things UNIX did for me when I migrated to it from a cyber was to say "Who?" to me when I bungled my login name. It was one of those many things that simply made it a friendly system.

- \* Our second (known) level 7 security breach occurred only recently, when the "bad guy" exploited sloppy coding in both our mail and network programs (this problem was found after my outburst in the previous AUUGN about mail). We have since rewritten our own mail program, which we are 100% happy with, and have "accidentally" lost all trace of its predecessor. The network software has been patched to avoid the problem, but should also be rewritten.

A few of our old friends have been converted from our level 6 system. "Dcopy", a program to copy a filesystem from one logical disk to another allowing changes in filesystem size and number of inodes, compressing directories by removing the empty slots, and taking into account the magic number for interleaving the file blocks for disk rotational optimisation, has been made to work again. Using it recently, 100 blocks were recovered through directory compression in a 65208 block filesystem (which took about 30 minutes to copy - cpio would have taken hours). Two other local, and invaluable, programs have also been converted. One is fed a process id number, and prints out a small table showing all the open files, mode of opening, any locks in force, etc. The other does the inverse operation - when fed a file, it will find all things accessing that file. It also handles the more general case in that you can feed it a filesystem, and it will tell you about all processes currently active on it. Both of these were real "dog's dinner" material under level 6; they are both much more readable under level 7 thanks to the superior C compiler.

Some changes have been made purely on a "offends my dignity" basis. For example, the "find" command had a "-cpio" option, which made it very convenient to create cpio tape dumps, but meant that a significant part of the code for cpio was duplicated in the find command. Regardless of convenience, this price was considered too high. It was removed. The various include files have been rationalised and moved into the one directory, instead of being spread around three. They have been changed so that (as far as possible) no defined constant appears in more than one file (sgtty.h and tty.h were particularly bad here), and so that no include file declares space or contains code (e.g. "map.h" declared the actual coremap and swapmap arrays as well as the structure; this meant that any program including map.h for the structure definition also got the added bonus of an unnecessarily large bss segment). We have altered the assembler to know about the more specialised instructions (such as reset, wait, spl) to avoid having to "declare" these at the top of programs that use them, so reducing the chance of error.

The changeover to level 7 has had unexpected advantages - for example, it has helped us solve a problem with our LA180 printers that we have known about for years but did not rate sufficiently high on the annoyance scale to warrant direct attention. We had set these printers up to raise and lower their carrier line as their buffers fill and drain (our DZ driver detects this and either suspends or restarts output). Nonetheless, we would still lose the occasional line or two of output every 30 pages or so. A change made to level 7 involved ringing the bell on terminals when they approached TTYHOG characters in the input queue without some program having read them (when this happens all input and any pending output is thrown away by the system). We were thus amused to

hear the bells on our LA180s ringing from time to time, just as they would lose a line or so of output. Further investigation revealed that the LA180s were still sending back copious numbers XON/XOFF characters (even though we had done the appropriate arm-waving to tell them they were carrier-controlled), eventually reaching TTYHOG (every 30 pages or so), thus causing the system to throw everything away (resulting in the loss of whatever was still in the system character buffers at the time for the printer). The problem has now been fixed with a soldering iron.

Due to severe time constraints, a lot of our changes have been done by direct modification of the system source (in other words you would have to "diff" it with the original to see what we had done). This was considered superior to riddling the code with conditional compilation sections to the point of making it totally unreadable (as in our level 6 system). We have used SCCS in some cases, but it can be incredibly slow, and in any case would require recipients of our system to have a PWB UNIX licence.

Future developments include re-doing the DZ and magnetic tape drivers, fixing the scheduling bug mentioned above, and looking at some sort of hashed least-recently-used (LRU) inode lookup scheme. There are still several "tame" bugs in the system that have to be found and fixed - for example, single-open devices occasionally hang because the system thinks they are still open; every now and again "proc on q" is printed out on the console indicating that a process that was about to be put on the runnable list was already there (a recent crash was caused by a related problem - a "free" process was found to be in the queue of sleeping, and hence active, processes); our dynamic disk limit part of the system has been poorly implemented and has serious problems. We want to spend a lot more time looking at schemes for squeezing level 7 onto small machines - at present we are using the UNSW BIG UNIX scheme, in which parts of the system are addressed as required by one of the segmentation registers. This scheme is a real black art, with vast potential for error. A nicer scheme, which we will be looking into, has been attributed to Mike Tilson, and involves mapping the text segments only (not the data as well). We also want to look into doing some system profiling, to attempt to identify bottlenecks.

In conclusion, we consider the changeover to level 7 to have been well worth the effort, and would not even contemplate going back to level 6. Though slow, level 7 has proven extremely stable and easy to modify. As far as most of our students are concerned, they neither know nor care if they are on our 11/70 or our VAX.

Although our changes are nowhere near complete, we are happy to send copies of our system to any interested sites (if allowed by our licence).

# UNIX Numerology

or

## A Summary of Magic Numbers

Dave Horsfall  
Computing Services Unit  
University of NSW

(dave:unswcsu)

To the best of my knowledge, the magic numbers used by UNIX have not been summarized in one place for easy reference, so once again I will get my name in print - this time with a learned discussion on numerology.

Magic numbers appear to be grouped into three classes of file: executable (including non-binary), library and "others". Executable files are not necessarily PDP machine code. PASCAL produces interpretive code, and SPITBOL only just comes under this classification since the magic number is the binary representation of "E." appearing at the start of every program. An interesting historical note is that the number "407" on the PDP-11 just happens to be an instruction that branches to beyond the 16 byte header. Perhaps bootstraps in those days were not smart enough to strip off this header.

Library files include non-loader archives such as CPIO. They are just a convenient way of grouping many items into one module.

The catch-all "others" consists of packed files, SCCS files and anything else that comes along.

This list is based on the FILE program at CSU, and so may be incomplete with respect to VAXii, V7 UNIX etc. It is presented in numerical (octal) order.

- 01           DEC (FORTRAN/MACRO) object module.
- 0404        PASCAL object.
- 0405        Text overlay.
- 0407        Ordinary executable.
- 0410        Pure executable.
- 0411        Separated I/D space.
- 0412        Shared data (R/W text).
- 017037     Packed file.
- 017437     Packed file (old format).
- 027105     SPITBOL program.

064001   SCCS file.  
070707   CPIO library.  
0177545   AR library.  
0177555   AR library (old format).



# Final Statement

Date	Particulars	Credit	Debit..	Total
20-11-79	Payin 1 (VolIII) \$174.00 receipt numbers 48201-48212, cash register numbers 134 & 135	174.00		174.00
6-12-79	Payin 2 (VolIII) \$34.00 receipt numbers 48213-48216, cash register number 739. (Receipt number 48215 accounted for below)	34.00		208.00
6-12-79	Internal transfer from account 10.130.3200 (receipt number 48215)	12.00		220.00
13-12-79	Mailing Vol II No. 1		46.40	173.60
8-1-80	Payin 3 (VolIII) \$120.00 receipt numbers 48217-48227, cash register numbers 93 & 94. (receipt number 48225 accounted for below)	120.00		293.60
12-1-80	Internal transfer from account 14.033.61 (receipt number 48225)	96.79		390.39
22-1-80	Xerox costs		108.75	281.64
8-2-80	Payin 4 (VolII) \$164.00 receipt numbers 48228-48238, cash register numbers 610 & 611.	164.00		445.64
2-2-80	Mailing of 5 items		9.40	436.24
2-2-80	Mailing of extra Vol II No. 1.		4.85	431.39
19-2-80	Xerox costs Vol II No. 2.		58.90	372.49
19-2-80	Mailing costs Vol II No. 2.		48.30	324.19
17-3-80	2 airmail letters to USENIX +Loginwest		1.00	323.19
27-3-80	Payin 5 (VolIII) \$214.00 receipt numbers 48239-48251.	214.00		537.19
24-4-80	Xerox costs 42*80 page \$53.71 + backissues		104.79	432.40
24-4-80	Vol 2 No. 3 + backissues mailing		57.70	374.70
7-5-80	Mailing 8 articles (7 * air + SAL)		7.45	367.25
20-5-80	Mailing 2 articles (1 airmail)		1.90	365.35
25-6-80	Mailing Vol II No. 4		61.25	304.10
25-6-80	Xerox backissues + Vol II No. 4		84.01	220.09
18-7-80	Payin 6 (VolII) \$202.00 receipt numbers 48252-48263, cash register numbers 573, 574.	202.00		422.09
18-7-80	Credit from receipt number 76 (21/4/80) for \$12.00 Subscription (Don McNeil)	12.00		434.09
18-8-80	Mailing Distributions of U-200 + Tape to St Peter's Lutheran College & UKUJG.		51.24	382.85

Date	Particulars	Credit	Debit	Total
27-8-80	Payin 7 (Vol III) \$222.00 (\$US25.00 accounted for later receipt number 48266) receipt numbers 48264-48267, cash register numbers 885 & 886.	222.00		604.85
3-9-80	\$US25.00 converted to \$A21.32 for receipt number 48266, cash register 418.	21.32		626.17
7-9-80	Xerox for Vol II No. 5 etc		12.19	613.98
27-9-80	Mailing for Vol II No. 5 etc		35.40	578.58
7-10-80	Payin 1 (Vol III) \$312.00 (\$US16.07 accounted for later receipt number 48277) receipt numbers 48268-48288, cash register numbers 596 & 597.	312.00		890.58
22-10-80	receipt number 48277 \$US16.07 converted to \$A13.57	13.57		904.15
30-10-80	Xeroxing for many backissues		104.80	799.35
26-11-80	Mailing for Vol II No. 6		68.30	731.05
18-12-80	Payin 2 (Vol III) \$649.00 (CAN\$12.00 accounted for later receipt number 48298) receipt numbers 48289-48317, cash register numbers 851 & 852.	649.00		1380.05
21-1-80	\$CAN12.00 converted to \$A8.45 for receipt number 48298	8.45		1388.50
21-1-81	Xeroxing for Vol III No. 1		94.14	1294.36
22-12-80	Mailing for Vol III No. 1		46.34	1248.02
23-1-81	Mailing of various items		10.70	1237.32
23-1-81	Xeroxing for Vol III No. 2		83.75	1153.57
23-2-81	Microfiche Vol I & II + Vol III No. 1 Microsystems See Order.		158.65	994.92
24-2-81	Mailing for Vol III No. 2		110.80	884.12
24-2-81	Mailing of package to USUUGN editor		11.10	873.02
8-4-81	Payin 3 (Vol III) receipt numbers 48318-48350, cash register numbers 157 & 159.	495.55		1368.57
28-4-81	receipt number 48351 \$US30.00, cash register number 013	25.73		1394.30
9-4-81	Mailing of Adrian Freed Australian Summary for UK meet		5.45	1388.85
22-4-81	Mailing Vol III No. 3		138.05	1250.80
22-4-81	Copying Vol III No. 3		103.47	1147.33
21-1-81	Credit from receipt number 619 9/1/81 CSIRO	30.00		1177.33

Date	Particulars	Credit	Debit	Total
9-2-81	Recovered the lost \$12.00 from Rob Freeth Uni of WA	12.00		1189.33
11-2-81	Transfer \$12.00 for receipt number 48339	12.00		1201.33
20-5-81	Mailing tape to Adrian Freed		6.65	1194.68
22-5-81	US. Bell Labs letter		1.65	1193.03
20-7-81	Mailing of Vol III No. 4		166.00	1027.03
30-7-81	Xeroxing of Vol III No. 4		116.01	911.02
12-8-81	Mailing of tape to Adrian Freed		7.45	903.57
28-9-81	Xeroxing of Vol III No. 5		67.40	836.17
28-9-81	Mailing of Vol III No. 5		97.30	738.87
28-9-81	Payin 4 (VolIII) receipt numbers 48352-48382, cash register numbers 913 & 914.	556.70		1295.57
7-10-81	Purchase of summer US UNIX conference, Software Tools tape and catalogue.		200.00	1095.57

Income & Xerox Costs.

—

The mailing costs are not  
computerised.....

Account 34-006-01

Name	Item	Amount	Receipt
Dave Parry	Subscription + set of backissues	22-00	48201
George William Gerrity	Subscription	12-00	48202
Geoff Cole	Subscription	12-00	48203
Piers Lauder	Subscription	12-00	48204
John Holden	Subscription + set of backissues	22-00	48205
Ross Gayler	Subscription	12-00	48206
Rod Bilson	Subscription	12-00	48207
Craig McGregor	Subscription	12-00	48208
Colin Boswell	Subscription	12-00	48209
Dr Peter Collinson	Subscription	12-00	48210
Joseph Longo	Subscription	12-00	48211
Richard Wolff	Subscription + set of backissues	22-00	48212

All subscriptions are for 6 issues from November 1979 to October 1980.

There are 5 backissues.

Receipt from cashier numbered 134 and 135. Payed in 20-11-79.

Australian Unix Users Group Newsletter

Account 34-006-01

Name	Item	Amount	Receipt
Rod Bilson	set of backissues	10-00	48213
John Lambert	Subscription	12-00	48214
Vincent Laurence	Subscription	12-00	48215
The Librarian	Subscription	12-00	48216

All subscriptions are for 6 issues from November 1979 to October 1980.

There are 5 backissues.

Receipt from cashier numbered 739. Payed in 6-12-79.

Account 34-006-01

Name	Item	Amount	Receipt
Wayne Harris	Subscription	12-00	48217
Chris Doney	Subscription	12-00	48218
Chris Maltby	Subscription	12-00	48219
Civ. Eng. Batch Station	Subscription	12-00	48220
Bruce Cheek	Subscription	12-00	48221
Dr. Terence Beed	Subscription	12-00	48222
Clarke Gerber	Subscription	12-00	48223
Colin Webb	Subscription	12-00	48224
Chris Rowles	Subscription	12-00	48225
Kevin Hill	Subscription + set of backissues	22-00	48225
Andrew Hume	Subscription	12-00	48225
Dr. Ian Johnstone	Subscription + set of backissues	22-00	48225
A.G.S.M.	Donation	28-79	48225
Robert Elz	Subscription	12-00	48226
Brian Rowswell	Subscription	12-00	48227

All subscriptions are for 6 issues from November 1979 to October 1980.

There are 5 backissues.

Receipt from cashier numbered 93 and 94. Payed in 3-1-80.



Account 34-006-01

Name	Item	Amount	Receipt
Mrs L. Dickinson	Subscription	12-00	48228
Christopher Barter	Subscription	12-00	48229
Juris Reinfelds	Subscription	12-00	48230
Kelvin Nicolle	Subscription	12-00	48231
Richard Miller	Subscription	12-00	48232
Ian Perry	Subscription + set of backissues	22-00	48233
Alistair Kilgour	Subscription	12-00	48234
John Hine	Subscription	12-00	48235
Gabi Steinberg	Subscription + set of backissues	12-00	48236
Ken Robinson	Subscription	12-00	48237
Jeff Ferguson	Subscription + set of backissues	22-00	48238

All subscriptions are for 6 issues from November 1979 to October 1980.

There are 5 backissues.

Receipt from cashier numbered 610 and 611. Payed in 8-2-80.

Australian Unix Users Group Newsletter

Account 34-006-01

Name	Item	Amount	Receipt
Ken J McDonnell	Subscription	12-00	48239
Ivan Fris	Subscription + set of backissues	22-00	48240
School of Building	Subscription	12-00	48241
Dr David W E Blatt	Subscription	12-00	48242
Ian Jackson	Subscription + set of backissues	22-00	48243
Kent Instruments	Donation	30-00	48244
C Stubbs	Subscription	12-00	48245
U.N.S.W. Library	Subscription	12-00	48246
Roger Brown	Subscription	12-00	48247
A.A.E.C. Librarian	Subscription	12-00	48248
Gregory Hill	Subscription + set of backissues	22-00	48249
Rodney Harries	Subscription	12-00	48250
Jean Williams	Subscription + set of backissues	22-00	48251

All subscriptions are for 6 issues from November 1979 to October 1980.

There are 5 backissues.

Receipt from cashier numbered 190,191. Payed in 27-3-80.

Account 34-006-01

Name	Item	Amount	Receipt
C.K. Fong	Subscription + set of backissues	22-00	48252
Peter Mason	Subscription + set of backissues	24-00	48253
David Carrington	Subscription	12-00	48254
Dr. John Lions	Subscription	12-00	48255
Stephen Pozgaj	Subscription + set of backissues	22-00	48256
D.C.I.E.M.	Subscription to vol 2 and 3	24-00	48257
The Librarian	3 various issues	6-00	48258
Clary Harridge	Subscription	12-00	48259
Don McNeil	Subscription	12-00	48260
Gershon Shamay	Subscription + set of backissues	30-00	48261
Shlomo Goldberg	Subscription + set of backissues	30-00	48262
Perry Brown	Donation	8-00	48263

All subscriptions are for 6 issues from November 1979 to October 1980.

There are 5 backissues.

Receipt from cashier numbered 573 and 574. Payed in 18/07/80.

Australian Unix Users Group Newsletter

Account 34-006-01

Name	Item	Amount	Receipt
Greg Kable	Subscription + Vol I backissues	30-00	48264
David Woodrow	Subscription + set of backissues	30-00	48265
Richard L. Lawhorn Jr.	Donation	21-32	48266
Rob Freeth	Subscription + software cat. contrib.	162-00	48267

All subscriptions are for 6 issues from November 1979 to October 1980.

Backissues are \$18-00 per volume, or 3 dollars each.

Receipt from cashier numbered 885, 886 and 418. Payed in 27-8-80 and 3-9-80(418).

Account 34-006-01

Name	Item	Amount	Receipt
Chris Rowles	Subscription to vol 3	12-00	48268
David Horsfall	Subscription to vol 3+ vol 2 nos 5&6	12-00	48269
David Woodrow	Subscription to vol 3	12-00	48270
Rob Freeth	Subscription to vol 3 and 4	24-00	48271
Craig McGregor	Subscription to vol 3	12-00	48272
Richard Wolff	Subscription to vol 3	12-00	48273
Dave Parry	Subscription to vol 3	12-00	48274
Geoff Cole	Subscription to vol 3	12-00	48275
George William Gerrity	Subscription to vol 3	12-00	48276
Steve Daniel	Payment for mag tape and postage	13-45	48277
Kim Malafant	Subscription to vol 3	12-00	48278
Rod Bilson	Subscription to vol 3	12-00	48279
Ian Perry	Subscription to vol 3	12-00	48280
Graham Smith	Subscription to vol 3 + Vol 2 No 6	14-00	48281
David Abulafia	Subscription to vol 3 + Vol 2 Nos 1-6	24-00	48282
Andrew S. Tanenbaum	Subscription to vol 3 + Vol 1 and 2	42-00	48283
Ivan Fris	Subscription to vol 3	12-00	48284
Munro Saunders	Subscription to vol 3	12-00	48285
Ken J McDonell	Subscription to vol 3	12-00	48286
U.N.S.W. Library	Subscription to vol 3	12-00	48287
Juris Reinfelds	Subscription to vol 3	12-00	48288

All subscriptions are for 6 issues from November 1980 to October 1981.

Backissues are \$18-00 per volume, or 3 dollars each.

Receipt from cashier numbered 596-597 and 944. Payed in 7-10-80 and 13-10-80.

Name	Item	Amount	Receipt
Christopher Barter	Subscription to vol 3	12-00	48289
Robert Elz	Subscription to vol 3	12-00	48290
Colin Boswell	Subscription to vol 3	12-00	48291
John Hine	Subscription to vol 3	12-00	48292
Ross Gayler	Subscription to vol 3	12-00	48293
C Stubbs	Subscription to vol 3	12-00	48294
Clary Harridge	Subscription to vol 3	12-00	48295
The Librarian	Subscription to vol 3	12-00	48296
Mr. Yong Hiong Leong	Subscription to vol 2 + vol 1	22-00	48297
Jerzy Manitius	Subscription to vol 3	12-00	48298
Richard Miller	Subscription to vol 3	12-00	48299
Dr Peter Collinson	Subscription to vol 3	12-00	48300
Alistair Kilgour	Subscription to vol 3	12-00	48301
Dr. R B Newell	Subscription to vol 3 + Vols 1 and 2	48-00	48302
Fawray Pty. Ltd.	Subscription to vol 3 + Vols 1 and 2	42-00	48303
Gershon Shamay	Subscription to vol 3	12-00	48304
Perry Brown	Donation	9-00	48305
Phil Chadwick	Subscription to vol 3	12-00	48306
Stephen Pozgaj	Subscription to vol 3	12-00	48307
A.A.E.C. Librarian	Subscription to vol 3	12-00	48308
Rick Stevenson	Subscription to vol 3 + backissues 1&2	34-00	48309
R. E. Kelly	Subscription to vol 3 + backissues 1&2	42-00	48310
Dept. Computer Science	Donation to Software Catalogue	200-00	48311
Paddy Hodgson	Subscription to vol 3	12-00	48312
Janet M Rybak	Subscription to vol 3	12-00	48313
David Carrington	Subscription to vol 3	12-00	48314
Dr. John Lions	Subscription to vol 3	12-00	48315
Kevin Hill	Subscription to vol 3	12-00	48316
Andrew Hume	Subscription to vol 3	12-00	48317

All subscriptions are for 6 issues from November 1980 to October 1981.

Backissues are \$18-00 per volume, or 3 dollars each.

Receipt from cashier numbered 851-852. Paid in 18-12-80.

Australian Unix Users Group Newsletter

Account 34-006-01

Name	Item	Amount	Receipt
Rod Curtin	Subscription to vol 3 + 1 and 2	48-00	48318
Daniel Braniss	Subscription to vol 3	12-00	48319
The Librarian	Backissues of vol 1 and vol 2	30-00	48320
J.D. Smith	Backissues of vol 2	18-00	48321
John Holden	Subscription to vol 3	12-00	48322
Dr. D. Davey	Subscription to vol 3	12-00	48323
The Director	Subscription to vol 3	12-00	48324
L.H. Leong	Subscription to vol 3	12-00	48325
Mark Ross	Subscription to vol 3	12-00	48326
Kevin Dawson	Subscription to vol 3 + 1 and 2	48-00	48327
Dr David W E Blatt	Subscription to vol 3	12-00	48328
Rodney Harries	Subscription to vol 3	12-00	48329
Tel-Aviv University	Supply of mag tape and mailing charges	14-15	48330
Brian Rowswell	Subscription	12-00	48331
Kelvin Delbarre	Subscription to vol 3 + backissues 1,2	48-00	48332
Keith Titmuss	Donation	9-00	48333
Shlomo Goldberg	Subscription to vol 3	12-00	48334
Glynn W. Peady	Subscription to vol 3	12-00	48335
Bob Kummerfeld	Subscription to vol 3	12-00	48336
Mr. M. Blake-Knox	Subscription to vol 3	12-00	48337
Mrs L. Dickinson	Subscription to vol 3	12-00	48338
Peter Cohn	Subscription to vol 3	12-00	48339
Ken Robinson	Subscription to vol 3	12-00	48340
Peter Mason	Subscription to vol 3	12-00	48341
Vince Lawrence	Subscription to vol 3	12-00	48342
Vince Lawrence	Backissue vol 1 no 6	3-00	48343
Albert Nymeyer	Subscription to vol 3, mag tape mailing	13-40	48344
G.F. Nelson	Subscription to vol 3	12-00	48345
Mr. B Durnota	Subscription to vol 3	12-00	48346

The Librarian	Subscription to vol 3	12-00	48347
Dr. G. Mohay	Subscription to vol 3 + backissues	42-00	48348
Jean Williams	Subscription to vol 3	12-00	48349
Mr. J. McKie	Subscription to vol 3 + backissues	48-00	48350
Arturo Puente	Subscription to vol 3 + backissues	30-00US	48351

Receipt from cashier numbered 157 & 159, 013. Payed in 8-4-81 and 28-4-81.



## Account 34-006-01

Name	Item	Amount	Receipt
Maria De Nicola	Donation	6-50	48352
Perry Brown	Donation	11-00	48353
C.S.I.R.O	Subscription to vol 3	12-00	48354
Greg James	Subscription to vol 3	12-00	48355
Greg James	Subscription backissues vol 1 and 2	36-00	48356
Kevin Broughan	Subscription to vol 3 + V2 nos 3 and 4	18-00	48357
Terry Gallagher	Subscription to vol 3 + backissues 1&2	48-00	48358
The Librarian	Subscription to vol 3	12-00	48359
Roy R. Rankin	Subscription to vol 3	12-00	48360
David Hunt	Subscription to vol 3	12-00	48361
P. Zwart	Subscription to vol 3	12-00	48362
H.C. Lucas & B. Martin	Subscription to vol 3	12-00	48363
Stephane Zadri	Subscription to vol 3	12-00	48364
Lindsay Harris	Subscription to vol 3	12-00	48365
Tektronics Pty. Ltd.	Subscription to vol 3	12-00	48366
J. R. Bilson	Payment for magnetic tape	10-00	48367
Kevin Dawson	Subscription to vol 4	12-00	48368
Jeffrey Tobias	Subscription to vol 3	12-00	48369
David Horsfall	Subscription to vol 4	12-00	48370
Michael Cullinan	Subscription to vol 3 + voll and vol2	42-00	48371
Andrew Fullford	Subscription to vol 3	12-00	48372
Jeff Freeman	Subscription to vol 3	12-00	48373
B. J. Simpson	Subscription to vol 3	12-00	48374
Munro Sauders	Donation	1-20	48375
Paddy Hodgson	Backissues of voll and vol2	36-00	48376
Dr. R.J. Lobb	Subscription to vol 3 + backissues 1,2	48-00	48377
Trevor Norman	Subscription to vol 3 + backissues 1,2	48-00	48378
Richard Wolff	Subscription to vol 4	12-00	48379

Robert Elz	Subscription to vol 4	12-00	48380
George William Gerrity	Subscription to vol 4	12-00	48381
The Librarian	Subscription to vol 4	12-00	48382
The Librarian	Subscription to vol 4	12-00	48383

All subscriptions are for 6 issues from November 1980 to October 1981.

Backissues are \$18-00 per volume, or 3 dollars each.

Receipt from cashier numbered 913 & 914. Payed in 28-9-81.



Total Xerox copying account balance to date:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
10/12/79	3600	1	65	SS	1-09.9	1-09.9
11/12/79	3600	19	40	DS	15-18.1	16-28.0
12/12/79	4000	130	12	DS	80-73.0	97-01.0
12/12/79	4000	2	12	SS	00-62.6	97-63.6
21/01/80	4000	19	10	DS	11-11.5	108-75.1

Total Xerox copying account since last transfer:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
14/02/80	4000	18	5	DS	5-31.0	5-31.0
14/02/80	4000	15	3	DS	2-68.5	7-99.5
14/02/80	4000	1	3	SS	0-09.2	8-08.7
14/02/80	4000	46	1	DS	2-89.8	10-98.5
14/02/80	3600	40	60	DS	41-56.0	52-54.5
14/02/80	3600	2	60	SS	1-04.4	53-58.9
15/02/80	4000	18	5	DS	5-31.0	58-89.9

Total Xerox copying account since last transfer:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
27/03/80	4000	53	4	DS	12-56.1	12-56.1
27/03/80	4000	2	4	SS	0-24.2	12-80.3
27/03/80	4000	43	5	DS	12-68.5	25-48.8
27/03/80	4000	1	5	SS	0-15.0	25-63.8
09/04/80	4000	44	4	DS	10-42.8	36-06.6
09/04/80	4000	33	5	DS	9-73.5	45-80.1
09/04/80	4000	18	3	DS	3-22.2	49-02.3
09/04/80	4000	53	1	SS	1-80.2	50-82.5
09/04/80	4000	2	4	SS	24.2	51-06.7
24/04/80	3600	42	80	DS	53-71.8	104-78.5

Total Xerox copying account since last transfer:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
17/06/80	4000	55	6	SS	7-92.0	7-92.0
18/06/80	4000	29	2	DS	3-50.9	11-42.9
18/06/80	4000	2	2	SS	0-12.6	11-55.5
19/06/80	4000	158	2	DS	19-11.8	30-67.3
20/06/80	4000	4	2	SS	0-25.2	30-92.5
25/06/80	3600	41	80	DS	52-43.9	83-36.4
25/06/80	3600	1	80	SS	0-64.2	84-00.6

Total Xerox copying account since last transfer:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
13/08/80	3600	2	100	SS	1-52.4	1-52.4
13/08/80	3600	5	1	SS	0-15.5	1-67.9
23/08/80	3600	14	80	DS	8-98.8	10-66.7
25/08/80	3600	2	100	SS	1-52.4	12-19.1



Total Xerox copying account since last transfer:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
29/10/80	4000	40	12	DS	24-84.00	24-84.00
29/10/80	4000	2	12	SS	62.60	25-46.60
29/10/80	4000	18	12	DS	11-17.80	36-64.40
29/10/80	4000	33	10	DS	19-30.50	55-94.90
29/10/80	4000	29	10	DS	16-96.50	72-91.40
29/10/80	4000	14	10	DS	8-19.00	81-10.40
29/10/80	4000	1	10	SS	29.50	81-39.90
29/10/80	4000	24	10	DS	5-85.00	87-24.90
29/10/80	4000	1	10	SS	29.50	87-54.40
29/10/80	4000	15	10	DS	8-77.50	96-31.90
29/10/80	4000	1	10	SS	29.50	96-61.40
29/10/80	4000	14	10	DS	8-19.00	104-80.40

Total Xerox copying account since last transfer:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
November	4000	3	90	SS	3-04.5	3-04.5
	4000	2	20	DS	1-53.0	4-57.5
	4000	1	70	SS	83.5	5-41.0
	4000	1	82	SS	94.3	6-35.3
	4000	1	72	SS	85.3	7-20.6
	4000	1	10	DS	58.5	7-79.1
	4000	1	8	DS	46.9	8-26.0
	4000	1	18	DS	72.9	8-98.9
	4000	1	46	DS	1-23.3	10-22.2
	3600	2	70	DS	3-31.8	13-54.0
	3600	1	44	DS	84.7	14-38.7
	3600	1	72	DS	1-18.3	15-57.0
	3600	1	82	DS	1-30.3	16-87.3
	3600	1	80	DS	1-27.9	18-15.2
	3600	30	90	DS	41-97.0	60-12.2
December	4000	145	2	DS	17-54.5	77-66.7
	3600	1	100	SS	76.2	78-42.9
	3600	10	100	DS	15-19.0	93-61.9
	3600	2	100	SS	1-52.4	95-14.3

Total Xerox copying account since last transfer:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
February	3600	50	100	DS	75-95.0	75-95.0
	3600	3	100	SS	2-28.6	78-23.6
	4000	2	100	DS	4-41	82-64.6
	4000	1	100	SS	1-10.5	83-75.1

Total Xerox copying account since last transfer:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
April	3600	114	1	DS	6-49.8	6-49.8
	4000	198	1	SS	6-73.2	13-23.0
	3600	64	1	DS	3-64.8	16-87.8
	3600	56	100	DS	85-06.4	101-94.2
	3600	2	100	SS	1-52.4	103-46.6

Total Xerox copying account since last transfer:						
Date	Xerox	Orig	Copies	SS/DS	Cost	Total
May	3600	1	20	SS	0-27.0	0-27.0
	3600	2	2	SS	0-11.4	0-38.4
	3600	1	50	SS	0-46.2	0-84.6
	3600	9	2	DS	0-98.1	1-82.7
	3600	11	50	DS	10-10.9	11-93.6
July	3600	1	110	SS	81.2	12-74.8
	3600	63	110	DS	103-25.7	116-00.5

Total Xerox copying account since last transfer:

Date	Xerox	Orig	Copies	SS/DS	Cost	Total
September	3600	24	120	DS	42-21.6	42-21.6
	3600	54	4	DS	11-50.2	53-71.8
	3600	56	4	DS	11-92.8	65-64.6
	3600	11	3	DS	1-76.0	67-40.6

From mhlsa!pwbcc!andrew Thu Sep 17 12:21:49 1981 remote from usa netmail from basser40  
some trivia on lawrence livermore laboratories:

8,230 employees  
in 375 buildings and trailers  
on a 640 acre site.

880 physicists  
360 computer scientists  
1170 engineers  
1700 admin support

12 large computers (7600s, STAR, CRAY)  
1500 terminals  
300 minis (VAX 780s, 750s, 11/70)  
1000 micros (LSI/11 or equiv)  
700 WPs  
1500 typewriters

micro-computers are a stores item!!

largest computer site in the world devoid of IBM equipment

andrew

From mhlsa!pwbcc!andrew Tue Sep 29 11:17:25 1981 remote from usa netmail from basser40  
Pete:

re you long-forgotten request for pricing of UNIXes. The new licences for UNIX 3.0 are about \$43,000 commercial (\$16k for each additional CPU) and the normal 3-400 for educational. WeCo are not yet thinking of an administrative license for UNIX 3.0. The other prices are old hat and I'm getting them in a day or two in the mail. Which prices do you want in particular?

andrew

From peteri Wed Oct 21 09:09:50 1981 netmail from elec70  
>From root Wed Oct 21 01:26:13 1981 forwarded by peteri netmail from elec70  
To: peteri  
Subject: 11/70 crash yesterday

Well very interesting indeed - it got a red zone stack violation for the very simple reason that the stack did indeed go too far. However, hidden in the stack was the fact that it had panic'd, due to the fact that an unused proc slot (containing junk) was in the runnable processes queue - this is undoubtedly related to the occasional "proc on q" message i.e. the whole thing has not been well thought out. I will continue investigating.....

kev

From auugn Wed Oct 21 11:03:57 1981 netmail from elec70  
To: gurus  
Subject: network host names

I have of late been considering changing the local machine names to something a little less formal.

eg

elecvox to peter  
elec70 to kevin (or just kev)  
elec35 to murray  
elec40 to david (or just dave)  
dsl to jeff  
etc etc etc.....

As I have thought of these names first, I would naturally fight to keep them should such a change be made. What do you think..?

Seriously though, what do you think of names like murgatriod and fandoogle instead of these insipid csu750s and things..... puke.....       pete

From kev Thu Oct 22 01:24:19 1981  
To: auugn  
Subject: network host names

I rather thought "zaphod beeblebrox" had a nice ring to it for the 70.....

kev

From lindsay Wed Oct 21 14:09:04 1981 forwarded by peteri netmail from agsm  
To: peteri  
Subject: network host names

We are not amused!

At least, not unless there is some sort of alias facility (i.e. elecvox is also peter etc). Alternatively, some sort of linking of names with location - e.g. all machines at elec have names starting with e, such as ernie, eryl etc.

Lindsay

From mhtsa!duke!decvox!peter Thu Oct 22 16:35:52 1981 remote from usa netmail from basser40

subject: return from europe and things

g`day, g`day:

i got back from europe a couple of weeks ago with a couple of items of interest for you and auugn readers (so feel free to publish the news, if you want to).

i have a friend who works for l'institut national pour la recherche dans l'informatique et l'automatique (the national research institute for computer science and automation). he and two colleagues designed a memory management unit for the motorola 68000 that only causes an 80 nanosecond effective delay for memory access. he and his colleagues hold the patent along with the french government. if you or anyone else is interested, his address and phone number is:

m. ciaran o'donnell  
i.n.r.i.a.  
domaine de voluceau  
roquencourt b.p. 105



78150 le chesnay  
france

telephone: 33 1 954 9020 (you have to ask for  
his extension ... in french)  
( monsieur oh dun ell, s'il vous  
plait ... the french are  
lousy at pronouncing english  
( irish, scottish ) names ).

in other news, a friend of a friend works at siemens in munich.  
he single-handedly wrote a c-compiler (that generated object code, not  
assembler) for the mc68000, ported unix v7 to the processor (with there  
own mmu ... a poor one that made access times effectively 2 to 3 micro  
seconds). he then proceeded to modify the kernel to run on a two  
processor 68000 (extendible to more processors) system. then he  
ported the remaining non-critical utilities, tool, and language  
processors (sans f77) to the motorola system. each stage took approximately  
four months to complete each stage of the project until september  
of this year, when the whole effort was cancelled, since siemens was  
committed to intel hardware.

two days ago i spoke with adrian freed on the phone  
and he said that his joining the uucp net in the u.s.  
has been delayed because of neglect due to being too busy.  
we got the april-may auugn last friday and as usual you  
beat login for a summary of the austin usenix meeting.

since you were here in june, a new product sporting a  
v7 (xenix hence unix) based o/s has come on the market.  
it is sold by micro da sys of los angeles and they claim  
to have a virtual memory o/s available soon (october 81?).  
they use a 6809 to replace pages and handle i/o generally ...  
such that when a page fault occurs, the 68000 is halted in  
mid-instruction, and the 6809 does its thing, thus holding  
the 68000 for a few milliseconds ... kludgy.

the address is: micro da sys  
2811 wilshire blvd  
santa monica, ca  
usa 90403

in more personal news, i haven't got my trip down under  
set up yet, but my brother tells me he's interested in  
tagging along, so i am looking at stopping in new zealand  
and tahiti somewhere along the way.

greetings to all .... and what's new at unsw and with  
the ivanovs?

peter stevens

tel. 1 213 829 6781

PLEASE POST

PLEASE POST

## Pre-Announcement

### 1982 Winter USENIX Meetings

The 1982 Winter USENIX Meetings will be held in Santa Monica, California, January 26 through January 29, at the Miramar Sheraton Hotel. This announcement provides early information about the dates of events, as well as persons to contact for further information. A Pre-registration Packet will be mailed before Thanksgiving. It will include both registration and hotel information and forms.

#### ☛ USENIX Technical Sessions – January 27-29, 1982

Session topics include Conversion, Networks, Graphics, Applications, Languages, Hardware, Word Processing, etc. A session is planned for vendor presentations. Copies of the Call for Papers may be obtained from the USENIX Association at the address given below.

#### ☛ Software Tools Users Group Meeting – January 26, 1982

The emphasis at this meeting will be standardization and consolidation of all the various implementations into a compatible system.

#### ☛ UNIX\* Tutorials – January 26, 1982

Two all day tutorials will be offered:

- 1) *Overview of UNIX*, intended for programmers without prior UNIX experience.
- 2) *Text Processing and Document Preparation*, intended for clerical staff users of UNIX.

For further information, please contact:

John L. Donnelly  
NCAR  
P.O. Box 3000  
Boulder, Colorado 80307  
(303) 494-5151

#### ☛ Vendor Exhibition – January 26 (afternoon) - January 29, 1982

An exhibit by vendors will be held at the Miramar.

Vendors wishing to exhibit should contact:

Mr. Harry Kerman, President  
Conventions West, Inc.  
9301 Wilshire Boulevard  
Beverly Hills, California 90210  
(213) 278-2326

If you did not receive this Announcement directly and wish to be on the mailing list for receipt of the Pre-registration Packet, please contact:

USENIX Association  
Box 8  
Rockefeller University  
1230 York Avenue  
New York, New York 10021

(212) 360-1039

Note: The above number will be changed shortly after 10/30/81 to (212) 570-8934

\*-UNIX is a trademark of Bell Laboratories

PLEASE POST

PLEASE POST

## 1982 Winter USENIX Conference

### Call for Papers

The 1982 Winter USENIX Conference will be held in Santa Monica, California, January 27 through January 29, at the Miramar Sheraton Hotel. Abstracts are now being accepted from individuals wishing to make presentations at the conference. Potential speakers must submit a 250-word abstract to Mike O'Brien (address below) by October 20, 1981. Abstracts may be sent by any available means; electronic mail is preferred.

It is planned that one session be available for vendor presentations. Presentations at this session should consist of factual descriptions of currently available products. More technical presentations should be reserved for the other, more technical sessions, at which vendors are also welcome.

Possible session topics include:

<b>VAX</b>	Further developments, 750 experiences, system extensions
<b>V7</b>	Rumor is that it's slow. Why? Tuning, extensions, and major and minor customization efforts.
<b>V6</b>	Is there a case for retaining it? Do some people prefer it?
<b>Conversion</b>	How to ease the pain of version conversion.
<b>Networks</b>	Local nets and long-haul nets. Distributed systems. UUCP installation and modifications. Building the USENET. Gateways. RJE.
<b>Graphics</b>	Core standards. Necessary system extensions. Exchange of available packages.
<b>Applications</b>	Mail systems, editors, database systems, games, etc.
<b>Languages</b>	New language systems and debussers. Changes, fixes, and extensions to C.
<b>Hardware</b>	Portability to new processors. News and rumors about new processors.
<b>Word Processing</b>	Newer and better tools and applications packages.

Abstracts must contain the following information:

- Full name of author
- Installation name and U.S. mailing address
- Electronic address, if available (Arpanet, or UUCP net relative to a "well-known" host such as ucgvax or decvax)
- Telephone number and hours available
- Audio-visual requirements

Authors will be notified of accepted presentations in advance of the conference. Talks already presented at previous conferences are discouraged unless the previous conference was one not likely to attract the USENIX community, or unless *significant* new developments are described.

Abstracts should be mailed to:

Michael T. O'Brien  
The Rand Corporation  
1700 Main Street  
Santa Monica, CA 90406

obrien@rand-unix (Arpanet)  
decvax!randvax!obrien (UUCP)

