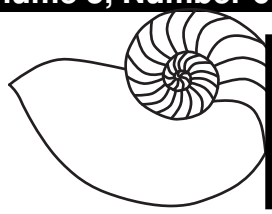
Volume 5, Number 6 April 1993 $2.50



MUUG Lite

Manitoba UNIX® User Group *Same Great Reading, With Only Half the News*

Newsletter of the Manitoba UNIX® User Group

The Intel Pantheon

Belief-Based Systems

By Doug Shewfelt and Arne Grimstrup

After Japan's massive Fifth Generation projects fizzled into oblivion, the computer industry lost interest in pursuing knowledge based systems. Leading players launched projects researching belief-based systems. MUUG Labs has just viewed a new (and still top secret) architecture based entirely on the belief-based paradigm: the Intel Pantheon.

Two of the earliest belief-based systems were, of course, Windows and OS/2. IBM claimed that OS/2 would run Windows faster than Windows, while Microsoft claimed that Windows NT would run OS/2 faster than OS/2. Naive users who truly believed these claims soon found that by loading several successive copies of OS/2 and Windows, they could make their machines run at any speed desired!

The next few years will see a battle between knowledge and belief-based systems for supremacy, with belief-based systems holding the performance edge. Researchers now feel that with belief-based systems a truly artificially intelligent machine is only 20 years away.

Hardware

In the Pantheon, Intel breaks from the dated ISA/EISA buses and introduces its new 32 bit Atlas bus. Theoretically, the new Olympus mount for the bus management processor makes this the fastest bus on the market. With all the advances made, Atlas still is a bottleneck. It's always holding up everything else in the system.

Atlas supports two highly asymmetric processors, Castor and Pollux. Castor is a littleendian CISC chip clocking in at around 150 Mhz. Pollux is a bigendian RISC chip of 100 MHz. We don't really know what Intel was thinking when they married these two processors — sometimes they act like the Bickersons — but somehow they manage to get the work done.

The system comes with a Benchmark coprocessor. Every benchmark suite we tried went right off the scale. We've never seen Whetstone, Dhrystone and Rhollingstone numbers like that before in our lives. We had to see what made this chip eat up our benchmarks. Shocked were we when it was discovered that this chip consisted of a large lookup table of precalculated results for each benchmark. Sheepish grins filled the labs — apparently there was a breakout in the Animal Science labs.

After sending the sheep back to the big house, we continued our evaluation. The base system includes the Medusa WORM drive system which simplifies the backup process. Almost. We never really completed our evaluation, because it turned the system to stone when we tried to browse the file hierarchy. Medusa also has some problems with Perseus, but we'll discuss that later.

We had more success with the RAID 9 Hydra disk drive. Unfortunately, whenever you remove one drive for repair, the controller spawns two more in the control table.

Continued on page 4

This Month's Meeting

Meeting Location:
Our next meeting is scheduled for Tuesday, April 13, at 7:30 PM. Once again, the meeting will be held in the auditorium of the St-Boniface Hospital Research Centre, just south of the hospital itself, at 351 Taché. You don't have to sign in at the security desk – just say you're attending the meeting of the Manitoba UNIX User Group. The auditorium is on the main floor, and is easily found from the entrance.

Meeting Agenda: See inside for details.

Inside This Issue

Newsletter Editor's Rantings 2
President's Corner 3
Feedbag: Ask Monsieur Ed 3
Review: The Pantheon (continued) 4
Hands-on: Software Portability, part 2 ... 5
Hardware: DL-232 – A New Standard 6
The Misfortune File 7
UniForum Superuser Award 8
April 13th Meeting Agenda 8

UNIX is a registered trademark of UNIX System Laboratories, Inc.

Groping With Change

(Name and address withheld)

Last month, and the month before that, we talked about dealing with change in our industry. We could start on a completely new topic this month, but why change now?

Working in the computer industry, we see so much change, we tend to become jaded about it. But change is not unique to our industry. Oh no, it's everywhere you go. You can't walk into any store these days without seeing it, and walking out with pockets full of it!

In fact, I'm getting pretty darn tired of dealing with all this change. And you know who's fault it is that we have to put up with so much change? Brian Mulroney's, of course! What? You don't believe me? Well, consider this.

What have been the two changes in Canadian monetary policy in the last few years that have resulted in an increase in the amount of coins we have to carry around? First, there was the loonie, that awkward, heavy coin that replaced the good-old dollar bill. Then, there was the GST, which ensured that nothing would ever again have a price that was a round number. So, not only do we have to shell out more money to the government than ever before, but now all we get as a *thank-you* is a pocket full of loonies and pennies!

I think I'm finally starting to understand the logic of trickle-down economics. We end up with so much extra coinage in our pockets that the pocket material eventually wears away. As a result of this, coins start to fall through the holes in our pockets, *trickle down* our pant legs, and roll down the street, right to the feet of some poor homeless person. Hey, what do you know? It could work out after all!

Of course, it wouldn't be fair to blame it all on Brian Mulroney. Let's see... There's Michael Wilson too! Besides, who said life was fair? Anyway, my point is that blaming others won't necessarily solve anything. It will just give us the satisfaction of knowing that all of this mess is someone else's fault, and that whatever happens, we won't be held responsible.

And in the end, that's all that matters, since who knows what will happen tomorrow? OK, Jeanne Dixon knows. But for all we know, we could be hit by a meteor, or space aliens could take over, or... What?... What are you looking at?... Hey! Stop staring at me! Ahhh! ➽

The 1992-1993 Executive

President:	Susan Zuk	(W) 631-2530
Vice-President:	Bary Finch	(W) 934-2723
Treasurer:	Rick Horocholyn	(W) 474-4533
Secretary:	Roland Schneider	1-482-5173
Membership Sec.:	Richard Kwiatkowski	589-4857
Mailing List:	Roland Schneider	1-482-5173
Meeting Coordinator:	Paul Hope	(W) 237-2361
Newsletter editor:	Gilbert Detillieux	489-7016
Publicity Director	Gilles Detillieux	489-7016
Information:	Susan Zuk	(W) 631-2530
		(FAX) 694-3901
(or)	Gilbert Detillieux	(H) 489-7016
		(FAX) 269-9178

Copyright Policy and Disclaimer

This newsletter is ©opyrighted by the Manitoba UNIX User Group. Articles may be reprinted without permission, for non-profit use, as long as the article is reprinted in its entirety and both the original author and the Manitoba UNIX User Group are given credit.

The Manitoba UNIX User Group, the editor, and contributors of this newsletter do not assume any liability for any damages that may occur as a result of information published in this newsletter.

Our Address

Manitoba UNIX User Group
P.O. Box 130
Saint-Boniface, Manitoba
R2H 3B4

Internet E-mail:
editor@muug.mb.ca

Group Information

The Manitoba UNIX User Group meets at 7:30 PM the second Tuesday of every month, except July and August. Meeting locations vary. The newsletter is mailed to all paid-up members one week prior to the meeting. Membership dues are $25 annually and are due as indicated by the renewal date on your newsletter's mailing label. Membership dues are accepted by mail, or at any meeting.

PRESIDENT'S CORNER

But Seriously, Folks...

By Susan Zuk, President

Welcome to MUUG's April Newsletter. This is the month where our editor likes to poke fun at the computer industry and give "April Fool's Day" a monthly feel. Hope you are having fun reading this edition.

We are in the process of preparing for ELSY Group's Computer Expo '93, which is to be held on May 8th and 9th. We have arranged for booth space and expect just as good, if not better, exposure than at last year's event. If any of you happened to attend, you will remember that there was a tremendous number of people flowing through the exhibit area and standing in lines waiting to enter. They have made this event bigger in order to accommodate more people and exhibitors. At our April meeting there will be a sign-up sheet requesting your help in manning the booth for a couple of hours on either day. Check your schedules so that you can sign-up and help us promote the group. Your help will be very much appreciated and will also allow you access to the show. If you cannot attend the meeting and would be available please e-mail or phone either myself (zuksue, 631-2530) or Gilles Detillieux (grdetil, 788-6766) with dates and times.

In continuing with the theme of promoting the group, we have started selling MUUG mugs. These mugs have our logo on one side and the caption "Wake up and smell the coffee, UNIX is here to stay" on the opposite side. We are selling these mugs to members for $3.00. These items are quite the hit and have been selling so quickly that we have placed another order. If you would like to purchase one of these collectors items call the 1-800 number and ask for operator 1523 or come to the April meeting.

Bary Finch is in the midst of forming a SIG (special interest group) which will deal with a combination of LINUX and system administration topics. He will probably be calling a meeting for the beginning of May (tentatively set for Tuesday, May 4). If you are interested in becoming involved or being placed on a list to be contacted of upcoming meetings either e-mail or call Bary (bfinch, 934-2723).

Remember the request for nominations for UniForum Canada's Open Systems Superuser Award. The final date for nominations is May 10, 1993. Please read the last page of this issue and let me know if you have a person that could be Canada's Superuser. What a great way to promote our Manitoba talent.

I would like to thank Greg Moeller for his presentation last month on LINUX. His presentation provided us with some interesting information about LINUX as well as how it is being developed. This is a new flavour of UNIX for 386 PC's which is available free of charge. If you would like to know how to obtain a copy you can ask Greg via e-mail at <gkm@muug.mb.ca>. I feel that one of the interesting parts of this operating system is the way in which it is being developed. This is a truly international project, with the base being in Finland and contributors being from around the world. The development is being accelerated by LINUX users around the world providing patches and new code via the Internet. This really shows us what an amazing network the Internet has really become. It also shows how very quickly things can evolve. LINUX being about one and a half years old is growing quite rapidly and has many features and standards built-in. We will have to see if it can move into working with newer processors. Greg, please keep us informed as to the progress of this operating system. Thank you again for the interesting presentation.

This month we will hear about Windows NT from Mr. Greg Brezinski. He feels that he is entering the lion's pit by coming to our meeting. Please help us to give him a warm UNIX welcome. We can then leave him with a MUUG mug to remember us by. Don't worry Greg, we'll take good care of you!

Just one note before I go, mark in your calendars that the May topic and presentation is on Sybase.

Oh, Gilbert, the board has just nominated you to be newsletter editor for the next 5 year's. April Fool!! Take care, I look forward to seeing you at the meeting on April 13th. ➸

FEEDBAG

Ask Monsieur Ed

A column in which our flatulent UNIX expert answers questions made up to look as if they were submitted by members, or discussed at round table sessions.

Tolerated by Gilbert Detillieux

Dear Monsieur Ed,

After staring at my computer's screen all day, my eyes go kind-of buggy, and I get pains in various parts of my body, especially in the neck and shoulders. What causes this to happen? Should I be concerned about it?

Are there any simple solutions to this problem? What can I do to avoid it, or to correct it when it happens?

Yours achingly,
Arnold P. Geekneck
Programmer/Analyst
Couch Potato Programming, Inc.

Dear Arnold,

I'm glad you wrote me now, before it's too late. Yes, you should be concerned about this. Be afraid, be very afraid! I have received reports from various reliable sources such as the *Weekly Inquirer* and the *National Tattletale*, that indicate that *space aliens* have been living on Earth for several years, disguised as various pieces of high-tech equipment. *Computer monitors* are their favorite disguise, since this allows them to infiltrate an increasingly large number of homes and secretly watch you, without you even realizing it. Your buggy eyes are due to their *hypnotic X-rays*, which they use to probe your mind and control you. The muscle pain is due to experiments they perform on your body, while you're in a *computer-induced trance*.

I can't say any more... My monitor is starting to act up. **Hey! Stop staring at me! Ahhh!** ➸

***Monsieur Ed**, a mysterious Frenchman who claims to be a* very old *editor (of course, of course), will return again next time he's accidentally set free. Consider yourselves warned.* ***Gilbert Detillieux**, a French-Canadian of non-mysterious origins, is mysteriously still the MUUG newsletter editor.*

The Intel Pantheon

Continued From Page 1

For those of you who believe that working with RAID technology is like playing with fire, Intel has the Prometheus SCSI controller which can be chained with up to 6 Eagle drives.

What about multimedia support? The Pan audio card sounds much too reedy, while the Themis video card was judged and found to be wanting. No matter what video drivers we tried, we could never get a proper colour balance. For those of you who will buy this system, we do NOT recommend the Terminus drivers. They make the system perform as if it were planted in the ground.

Hermes, the networking card, supports all of the IEEE Lan standards and includes a built-in V.fast modem. This is truly an outstanding piece of hardware and Intel should be commended. What we still haven't figured out is why we keep seeing pizza delivery people on our doorstep carrying anchovy, black olive, double onion and garlic pizzas 30 minutes after we use the modem.

Overall the hardware has numerous bells and whistles, not to mention a few Sirens.

Software

Following the landmark decision of "Microsoft vs The Rest of the World", the new operating system must be called "Windows NT 3.1" Version 1.0, thus allowing Microsoft a share in all profits. Currently, it is in the early stages of the later pre-beta beta beta spam beta beta test, where every user gets a slightly different version. As part of Intel's generous upgrade policy, everyone will be receiving a copy of the FINAL version of the operating system for free.

The hardware directly supports all standard data types except for booleans. However, booleans can be emulated in software by equating TRUTH and BEAUTY. Boolean values may thus be represented in the Aphrodite Love Unit (ALU), which supports user defined constants of up to 513 millihelens. *(Ed. note: A helen is the unit of BEAUTY needed to launch one thousand chips.)*

To compete in the tough workstation market, the Pantheon machine comes bundled with many software tools. The Icarus emulation package was probably the best we'd ever seen, but it was pulled from the system when Sun threatened a look-and-feel lawsuit. Unnamed sources at Intel have been overheard saying "we went down in flames when Sun put the heat on us." Sun systems are now handled by the Apollo processor emulator.

The C compiler, Poseidon, is fully ANSI compliant, but really shines in its enhanced mode. Our tests of the trident() call, for example, show an almost 3 fold increase in speed over the standard fork(). We were shocked, however, when we saw the results from the tutorial programs for this particular system. Instead of the pleasant "Hello, world", Poseidon's output was "WHAT FOOLISH MORTAL DARES TO DISTURB ME?" Fishy, no? In fact, a number of the sample programs would go nonlinear and attack Andromeda (the network file system). Fortunately, the compiler and operating system teams provided Perseus (the memory manager) to deal with these errant C monsters.

Cassandra is a fully Oracle compatible RDBMS. It out does Oracle in a number of areas. Cassandra is fully peer-to-peer capable and is capable of running on a distributed system when coupled with Andromeda and Heracles. For some odd reason, Cassandra is at her best when Apollo is up in the background. Although Cassandra is as gifted as Oracle, we found ourselves disbelieving the reports that were generated.

Heracles, one of the stronger points of the system, does a heroic job of handling communications tasks like FTP, Telnet, and mail. Fault-tolerance of Heracles does leave something to be desired. We noticed that if the Iphitus layer dies, Heracles reacts oddly. Andromeda runs acceptably, but the installation process is very straining.

Task management in the operating system is done by the Hades and Charon daemons. Inactive processes are collected by Charon in a similar fashion to lazy-writing. (Charon would be better described in this context as a faerie process.) Those scavenged process are handed off to Hades for long duration storage. Perseus interacts well with Charon and Hades, but has a very nasty habit of killing the Medusa WORM drive.

As impressed as we were with the other tools, the Cerebus security system was a let down. On the surface, the triple-threaded package should out perform all others on the market. Unfortunately, our tests showed some serious flaws. "It's a real dog," one dissatisfied tester proclaimed, "I went through hell to get some files that slipped past it."

If you are thinking about double-clicking the Pandora icon, DON'T! Just trust us on this one.

Winding Down

Shortly after our investigation of the system, we contacted Thomas Bulfinch, supposedly the leader of the Pantheon development team. We expressed our concerns about the lack of orthogonality of the system — subsystems are drawn from a variety of historical pantheons. Dr. Bulfinch, rather confused, demanded to know was "orthogonality" meant. When we explained, he told us that architecture consistency was against written company policy at Intel. "In any case," he continued, "Intel believes that its 'Pantheon' trademark provides sweeping claims to participants of all mythologies, past and present." Finally, as an afterthought, he denied the existence of the Pantheon project.

Intel has its work cut out for it. However, removal of the rather major bugs from this system will allow this neo-classical architecture to be appreciated by everyone, even if it's all Greek to you. It is a bold entry into the belief-based arena. Intel's head start may prove to be unbeatable; the Pantheon improves as more people believe in it.

A reliable source has told us that several students at the M.I.T. Center for Industrial Theology are praying for quick application ports. ➽

Writing Software for Portability, Part 2

By Gilbert Detillieux, Info West Inc.

Last time, we looked at how to structure a program to improve portability. By breaking it up into small, independent modules, and isolating system specific code to just a few low-level modules, you make it more portable. Even more important to assure portability is the way we use data structures and data types. We will now look at certain considerations in using data in memory, and more importantly, how to use them in data files.

Data Types and Structures in Memory

Abstract data types can be used in many languages, to help clarify or restrict the processing we do on that data. We also can make use of this on data whose representations may change, to simplify such changes. In C, system header files define many such data types (`time_t`, `uid_t`, etc.), so we don't need to worry about the internal representation, which may change from system to system. Likewise, we also can define our own data types (e.g., `typedef long int acctnum_t`).

You should never make assumptions about the size of data types and objects in memory, as these are machine (and compiler) specific. It may not even be safe to assume that a character takes one byte, if your code is to support international character sets. Also use named constants for array sizes and counts. If the system headers define some of these (buffer sizes, maximum number of files), use them; otherwise, define your own. In C, always use "sizeof" to specify sizes, when you need the size of a variable. For example, use `malloc(NCHRS*sizeof(char))` rather than `malloc(80)` – hard-coded constants for sizes are always a sign of trouble.

For structured data types, there are some extra considerations. Element sizes, alignment of elements and padding can all vary with machine or compiler. Don't assume padding is done or not, or how it's done. With unions (variant records in Pascal), don't assume a particular alignment of the various members. Of course, you can't easily determine the overall size of a data structure in a portable way (other than using "sizeof" in C). If you're concerned about keeping the size of structures down to a minimum, you can assume that the best way to do that is to put all the largest elements first, then work your way down. This will reduce the amount of padding that might be introduced, and shouldn't hinder portability in any way. (In C, it's probably safe to assume the following size order: `doubles`, `floats`, pointer types, `longs`, `ints`, `shorts`, then `chars`.)

There might be restrictions on how you can allocate data, and the size that you can allocate. Avoid allocating data in a way that isn't supported by all compilers (for example, initialized automatic arrays are supported by some compilers but aren't standard, and register types supported can vary with compilers). Some architectures have restrictions on the size of particular objects – automatic variables may be restricted by the size of a stack frame, or static or dynamic data may be restricted by the maximum size of a data segment.

Data File Formats

All this data has to come from somewhere, and may have to go somewhere else. This usually means data files. You may have to design your own data file formats, and these should be designed with portability in mind too. This should be based on usage – decide what data must be kept, for what purpose, and for how long, then define your format accordingly. Consider using text files for permanent data storage, as this is not only far more portable, but more versatile, flexible, and maintainable. There are lots of existing programs that you can use to edit, filter, search, and sort these files, and their format is less tied to program code changes.

Use binary files only if you have to, due to space or time constraints (i.e. if you need a more compact representation, or if conversion to/from internal representation would be too slow). Clearly define the data types you are using, their size, representation, and even bit and byte ordering. Unlike for data types in memory, where it was better not to make assumptions about internal representation, here we must know exactly how everything is stored, so the data can be treated consistently, regardless of which program, machine, or operating system is involved.

For data structures in files, use explicit data types. (In C, use short int, or long int, rather than just int, which is more machine specific.) Avoid automatic padding between elements by ordering them carefully. Or you might want to introduce explicit pad bytes to force a particular alignment; this also allow for growth if you have to add new elements later. Define clearly any assumptions you make about element sizes and offsets, and test this out with the various compilers you will use.

A Little Bit of Magic

It's a good idea to start each file with a "magic number" or "magic string" which can be used to identify uniquely the type of a file. The value you pick is arbitrary, but should be long enough (four bytes or more), and unique. If you use an integer magic number, you can use the bit pattern in your code to determine the bit and byte ordering in the file automatically (e.g. 1234 v.s. 4321). A magic string is perhaps more portable, and can be used for both text files (e.g. "%!" in PostScript files) or binary files (e.g. GIF87 in GIF files).

Version numbers in files are useful as well, to identify variations in a file's format. A major version number change could indicate a significant format change (or you might want to change the magic number if the files are

incompatible), while a minor version number change would indicate a small change that doesn't affect the format (e.g. new data stored in former pad bytes). Version numbers also could indicate the presence or absence of sections, allowing changes and additions to the file's contents while still maintaining compatibility.

Avoiding Obsolescence

As much as possible, your programs should handle all file versions, so old data files aren't made obsolete by program changes. If it's feasible, data files can be converted on the fly when your program loads an older file, or one from a different system. Failing that, conversion programs should be provided, to support file formats for all versions, systems, and machines. Remember that data files are likely to outlive the programs that created them, and it may be important to continue to support them.

In a way, file formats are like protocols – you should stick to a standard, define it clearly, and plan for future growth. Use industry standards where possible (e.g. ASCII or ISO character sets, IEEE floating point format, Sun's XDR standards), or define your own if you have to, but do so carefully. And most importantly, document everything you do (data types, structures, file formats, versions supported, conversions, defaults for missing data, etc.), keep these documents consistent with your code, and make them available to everyone who needs to know.

There is no simple formula to assure portability of software and data. Choosing languages and systems that are portable is an important step, but that isn't enough. You have to plan for future growth in machines, systems, and in your own software. Think about portability from the start. Define and stick to standards, but also be flexible – use an open-ended design to allow for unforeseen changes. And document, document, document! ➧

Gilbert Detillieux *would like to apologize to any readers who scanned this article looking for a punch line.* :-)

DL-232 — A New Standard

By Dave Lyons

Submitted by Tom Dubinski (From: Rupa Schomaker, newsgroup: rec.humor.funny)

I may never understand how the designers of the RS-232 "standard" for serial communication managed to use 25 wires where only 3 are really necessary. Maybe they made a deal with the companies that make cables, connectors, and switch boxes...I just don't know.

Well, I thought of a few things that the RS-232 standard lacks, and since there are already so many extra signals, a few more can't hurt anybody, right? Heck, let's go for 50-pin connectors and cables and add the following new signals. (Just to make sure this isn't compatible with any old equipment, all OLD signals are moved up one pin number (Carrier Detect becomes 9 instead of 8, etc., and pin 25 goes to pin 1).

Pin	Name	Description
26	XCAT	Should be connected to chassis of devices. Used with the next two signals, this provides protection against cats who haven't learned not to walk on floppy disks or serial equipment.
27	CATGND	Cat ground. Used with pin 26. This signal should be connected to another part of the chassis or the tabletop.
28	CTD	Cat detect.
29	SD	Self-destruct. This signals causes the device to destroy itself.
30	SDACK	Self-destruct acknowledge. Acknowledges that the device has destroyed itself.
31	VADIC	This signal indicates to a computer that the device on the other end is a modem that uses VADIC protocol. (Note: CompuCenter Iowa users should jumper this signal to SD and then buy a decent modem.)
32	STBIT1	Stutter bits. With pin 33, sets the number of "stutter bits" (0 to 3 of them) to be included before each byte transmitted. This may reduce the number of people who feel inferior to computer equipment by showing them that computers have problems communicating with each other.
33	STBIT2	(See above.)
34	CABR	Cable ready. It's not enough to know that the Data Set is ready (DSR) and the Data Terminal is read (DTR). We also need to know that the CABLE connecting them is ready.
35	GRR	Gremlins ready. Not everybody knows it, but there are little green guys inside most modern computer equipment. Most of the time they sleep, but other times they cause trouble. The next 6 signals are for dealing with gremlins.
36	220A	Used with pin 37, supplies 220 volt power for the gremlins' air conditioning. On hot days when gremlins can't sleep, applying power to these pins may solve your problems.
37	220B	(See above.)
38	110H	110 volts, hot side. When the 220 volt power doesn't help and gremlin problems persist, use this with pin 39 to supply 110 volts for the gremlins' TV and video game center.
39	110N	110 volts, neutral side.
40	MOON	Indicates the phase of the moon. Sometimes solves mysterious problems.
41	LHI	Pins 41 through 45 can be used to implement the "like" protocol when the normal RTS/CTS protocol isn't enough. This one means "Like HI" and is used to establish a connection.
42	LHTY2	Like HI to You Too. Acknowledges pin 41.
43	LLTT	Like Listen To This. Requests permission to send data.
44	LOK	Like OK. Grants permission to transmit data.
45	LWOW	Like WOW. Acknowledges receipt of data.
46	HEY	Pins 46 to 50 may be used to implement the "Eighties" protocol when RTS/CTS and "Like" protocols won't do the job. This signal is similar to RTS (Request to Send).
47	NP	No Problem. Acknowledges HEY.
48	HUH?	Signals that data was not received correctly (possibly wrong number of stutter bits).
49	YEAH	Acknowledges data received.
50	KMG365	Like YEAH, but for avid Emergency One fans.

That makes 50! Let's hear your suggestions for MORE serial signals. Maybe we can get 100 and REALLY make the cable manufacturers happy. ➧

The Perils of Hardware Installation

Submitted by Daniel John Little <danl@aeris.muug.mb.ca>

This story has appeared in various Usenet news groups and in print in several publications.

Greetings, dear reader. Today we bring you a hardware installer's accident report that was printed in the newspaper of the English equivalent of the Worker's Compensation Board.

Dear Sirs,

I am writing in response to your request for additional information in Block 3 of the accident reporting form. I put "Poor Planning" as the cause of my accident. You said in your letter that I should explain more fully and I trust that the following details will be sufficient.

I am a computer hardware contractor by profession. My latest assignment involved replacing a client's old mini-computer hardware. Due to technological advancement, the new hardware was compact and light-weight which allowed me to install it on my client's premises with no difficulties. However, removing the old CPU presented a few problems. Since the installation of the old CPU 12 years ago, the building had been totally remodelled and the box could no longer be removed via the stairs or elevator. My client did have access to the roof and I decided to lower the CPU by using a pulley which, fortunately, was attached to the building at the 10th floor.

Securing the rope at ground level, I went up to the roof and swung the CPU out. Then I went back to the ground and untied the rope, holding it tightly to insure a slow descent of the 500 pound piece of hardware. You will note in Block 11 of the accident reporting form that my weight is 135 pounds.

Due to my surprise at being jerked off the ground so suddenly, I lost my presence of mind and forgot to let go of the rope. Needless to say, I proceeded at a rather rapid rate up the side of the building. In the vicinity of the fifth floor, I met the CPU which was now proceeding in a downward direction at an equally impressive rate of speed. This explains the fractured skull, minor abrasions and the broken collarbone, as listed in Section 111 of the accident reporting form.

Slowed only slightly, I continued my rapid ascent, not stopping until the fingers of my right hand were two knuckles deep into the pulley, which I mentioned in paragraph 2 of this correspondence. Fortunately, by this time I had regained my presence of mind and was able to hold tightly to the rope in spite of the excruciating pain I was now beginning to experience.

At approximately the same time, the CPU hit the ground - and the guts fell out of the box. Now void of the weight of its internals, the box weighed approximately 50 pounds. I refer you again to my weight in Block 11. As you might imagine, I began a rapid descent down the side of the building.

Again, in the vicinity of the fifth floor, I met the box coming up. This accounts for the two fractured ankles, broken tooth, and severe lacerations to my legs and lower body.

Here my luck began to change slightly. The encounter with the box seemed to slow me enough to lessen my injuries when I fell into the pile of boards and chips, and fortunately, only three vertebrae were cracked.

I am sorry to report, however, that as I lay there in pain I again lost my composure and presence of mind and let go of the rope.

Sincerely,

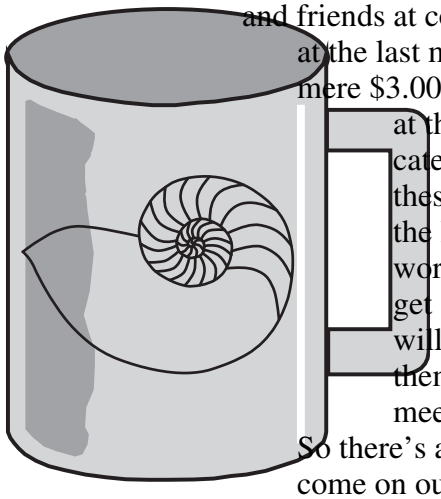
Policy XYZ 23456789. ➸

Announcement

MUUG Mugs Are Here

The Manitoba UNIX User Group has ordered an additional 96 custom designed coffee mugs, which bear the group's name and logo on one side. These mugs will be used as thank-you gifts for group members and other local speakers who give presentations at our meetings. They will also be used as promotional items at computer shows such as the upcoming MWCS Computer Show in May.

In addition to that, we will be offering them to MUUG members and friends at cost, as was mentioned at the last meeting. Cost is still a mere $3.00 per mug. Many people at the last meeting indicated an interest in buying these mugs. If you missed the last meeting, don't worry – it's not too late to get a mug for yourself. We will have a quantity of them on hand at the March meeting.



So there's another good reason to come on out to the meeting!

Canadian Open Systems Superuser

Sponsored by UniForum Canada

UniForum Canada is currently calling for nominations for its annual Canadian Open Systems Superuser award. This award is given each year to individuals who have made extraordinary contributions, in any field, to the Canadian UNIX and Open Systems Community. The award need not recognize achievements for a single calendar year, but can be more broad in scope. An award recipient need not be a member of UniForum Canada or an affiliate body. Organizations or incorporated bodies are not eligible.

Nominations for this award must be made, and seconded by members of UniForum Canada or an affiliate group. Nominations must be received by the Award Coordinator, Susan Zuk, by May 10, 1993. Anyone who would like to submit a nomination (or more) or would like more information, please call Susan at 788-7312.

The award will be presented to the recipient(s) at a future date, most likely at one of UniForum Canada's annual UNIX shows.

Agenda

for
Tuesday, April 13, 1993, 7:30 PM
St-Boniface Hospital Research Centre
Theatre, Main Floor, 351 Taché

1.	President's Welcome	7:30
2.	Round Table	7:40
3.	Business Meeting a) Old Business b) New Business	8:00
4.	Break	8:15
5.	Presented Topic Microsoft's Windows NT By Greg Brezinski *Microsoft's new operating system, although not UNIX based, does fit into the big picture of open systems technology. This first look at Windows NT will help us see how it will fit in, and how it compares to other systems. The demonstration will be on a live system, with the display connected (hopefully) to a large projection TV unit.*	8:30
6.	Adjourn	9:30

Note: Please try to arrive at the meeting between 7:15 and 7:30 pm. Thank You.

Coming Up

Meeting:
Next month's meeting is scheduled for Tuesday, May 11, at 7:30 PM. Meeting location will likely be the same as this month. The meeting topic will be a presentation on the Sybase relational database system.

Got any ideas for meeting topics? Any particular speaker or company you'd like to see at one of our meetings? Just let our meeting coordinator, Paul Hope, know. You can e-mail him at <phope@muug.mb.ca>.

Newsletter:
There are a few articles that should be on the way, including a book review on another book about the Internet. Of course, I can always use some more material, especially shorter articles – half a page to one page (400 to 1000 words) would be fine. Monsieur Ex has also let me know that his mail-box is empty once again – please submit your questions to the old guy via e-mail to <m-ex@muug.mb.ca> or by FAX to the MUUG Lines editor.