Volume 5, Number 4 February 1993 $2.50

MUUG Lines

Manitoba UNIX® User Group

Newsletter of the Manitoba UNIX® User Group

Is There SIGnificant Interest?

By Bary Finch

Well, here's my first article, and it starts with a bad pun at that. I am working my way into my role as vice-president, and gaining a better understanding of MUUG as I go.

My first main task will be to assist in the formation of one or more Special Interest Groups (SIGs). The purpose of SIGs is to enable specific interests to be pursued to a greater depth than is allowed by MUUG's regular meeting schedule. I had requested ideas for SIGs that people would like to see formed, and so far two specific ideas for SIGs have been put forward.

Derek Hay has suggested forming a "System Administration" SIG. This group would exist to examine the "basics" of system administration in UNIX. Many of you may have exposure to UNIX, but have not had the opportunity to delve into system administration.

This kind of SIG may benefit greatly from the use of hardware and software to enable experimentation, if it is desired by the group. This would then become a main focus of the group to see if there are facilities available, or SIG members that have equipment available.

Another suggestion comes from Rennie Allen. He would like to form a SIG for "Real Time" systems. This SIG would explore aspects of UNIX as a real time operating system, as well as programming for real time applications.

As with the system administration SIG, equipment would probably be an integral part of the group. Rennie may have access to a version of the QNX operating system, as well as hardware to run it on. This will have to be explored further when the SIG forms.

For either of these SIGs we will need at least ten people to express a commitment to forming the SIG. This would be sufficient numbers to support any activities that the SIG may want to undertake.

Please let me know if you are interested in either of these SIGs. You can e-mail me at `bfinch@muug.mb.ca`, or at work: `finch@torvm3.vnet.ibm.com`, or call me at work: 934-2723 (warning: voice mail awaits!). If sufficient interest exists for either group, I will co-ordinate getting the first meeting arranged.

After that, the SIG will need to assign appropriate roles to its members to continue its activities, for example a group "coordinator," and then the real fun can begin! ✍

***Bary Finch** is a Systems Engineering Representative with IBM in Winnipeg. He is the current MUUG vice-president.*

THIS MONTH'S MEETING

Meeting Location:
Our next meeting is scheduled for Tuesday, February 9, 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

Living In Interesting Times

By Gilbert Detillieux

"May you live in interesting times," says the old Chinese curse. As I sit and write this on Chinese New Year's day, I can't help but think that this applies to our time. With all the changes we've seen in the political world in the last few years, how can we not find this "interesting"? Certainly, great tension and upheaval exists during such times, making things not only interesting, but stressful and a little frightening as well (hence the curse).

Of course, "interesting times" is also a good way to describe the rapid technological changes we see around us. These changes create lots of new opportunities, and interesting possibilities we wouldn't have even imagined. On the other hand, such changes, if not managed in a responsible way, can create lots of new problems and casualties. Whether or not we benefit and prosper depends on how we adapt to the changing times.

UNIX now seems positioned as an engine of change. The computer industry started in a very centralised way, with a few computer centres controlling information technology for a large number of people. The microcomputer then created a popular revolution, where individuals now were in control, and free to develop their own information technology solutions independently. This created information islands, with little communication (computer or human) between them. Now, with the advent of wide-spread networking, and the realisation that improved communication is key to effective use of computer technology, UNIX and open system technology provide the missing link. UNIX is becoming the glue that binds the various technologies, and allows greater interoperability.

Yet, there is great resistance to this change. Many in the industry see UNIX as a threat, since it means more learning, and abandoning a lot of the old ways. Since change in our industry is inevitable, we should accept the challenge it poses, rather than resisting. Whether UNIX or some other system wins out in the end is irrelevant – the point is things will change, and if we manage that change properly, we stand to gain.

Learning to cope with new technology, and facilitating that learning for others, is what computer user groups are all about. The recent interest in forming special interest groups in MUUG, and the large number of "hands-on" articles I've received for the newsletter are signs that our members are meeting the challenge of learning head-on.

May the year be an interesting one for you. I hope to see all of you at this month's meeting. ✒

The 1992-1993 Executive

President:	Susan Zuk	(W) 788-7312
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) 788-7312
		(FAX) 788-7450
(or)	Gilbert Detillieux	(H) 489-7016
		(FAX) 269-9178

Copyright Policy and Disclaimer

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

Events Past, Present, and Future

By Susan Zuk, President

January proved to be very busy for the Manitoba UNIX User Group. On top of holding our regular monthly meeting we co-hosted a second event with CIPS (Canadian Information Processing Society) and the Winnipeg PC Users Group. I’ll first give you a synopsis of our meeting held on January 12th.

Mr. Jim Baglot, of Frame Technology Corporation, was our guest for last month’s meeting. He flew in from Vancouver to provide us with a presentation on FrameMaker, FrameViewer and FrameBuilder.

FrameMaker is more than a desktop publishing system, it is described as a document publishing system. What this means is that you use it for more than just word processing and graphics work. This system is one that you can base your whole organization around. Networked systems can hold documents, like letterhead, memos and user manuals. These can be shared and accessed at any time. Information can be retrieved in a query-like fashioned by clicking on pictures, linked words or phrases (using hyper-text) or documents may be used as templates with input stops. These documents can in turn be printed out or mailed electronically between office users internally or externally by various means.

This product and presentation was very interesting in a number of ways. It provided us with a look at some “shrink-wrapped” software written for many platforms, including DOS and UNIX, and it showed us the type of power and flexibility there is for those users and organizations who really need a product to handle their internal document woes. It also showed us how the office of the future may look. Documents like memos and letterhead being held electronically, not in storage rooms. Changes being made and received, instantly not at the next shipment of paper. Things like training manuals, specification sheets being on-line for those who require up-to-date and many other automation requirements.

At the end of the presentation, Jim drew for a free copy of FrameMaker. The lucky winner was Michael Doob of the University of Manitoba. Congratulations to Michael. Maybe he can provide us with a first-hand review of this interesting and timely product. Thank you Jim for your presentation. Another special thanks goes to Roland Schneider for bringing in his huge monitor for the event. It’s called weight training, right Roland? Also thanks to Paul for setting up the viewing environment.

Our second event of the month was an evening with Jeffrey Armstrong, otherwise known as Saint Silicon. This was a fun evening with, as he states, the world’s first computer comedian. He came to us from Santa Cruz, California, and entertained us with his computer humour. A thank you goes out to the CIPS organization, which funded this presentation and allowed us to take advantage of his visit to Winnipeg. It was interesting to see how the Saint showed us how computers run our lives and not the other way around!

Please take note of Bary Finch’s column on SIG’s. We have had a number of people requesting the creation of certain special interest groups. If you have more suggestions or would like to help out, please give Bary a call.

On the last page of the newsletter there is a request for nominations for an award. This award is called the Canadian Open Systems Superuser and is sponsored by UniForum Canada. Take a look at this article to find out more about the award. If you have suggestions or nomination(s) please call me.

I shall close for now, and look forward to seeing you at our February meeting. ✒

FEEDBACK

Ask Monsieur Ex

A column in which our resident UNIX expert answers questions submitted by members, or discussed at round table sessions.

By Gilbert Detillieux

Dear Faithful Readers,

Last month, you might recall, we left you with the following question, from Roland Schneider, as a quiz.

What is the significance of 6:00 pm, December 31, 1969?

I then added a couple more dates for which you may want to figure out the significance:

```
Mon Jan 18 21:14:07 2038
Fri Dec 13 14:45:52 1901
```

(For all of these, you were to assume Central Standard Time.)

Eh bien, mes amis, Monsieur Ex will leave you in suspense no longer. The solution to Roland’s question first, since this is a rather easy one. That given date and time, when converted to Greenwich Mean Time (GMT) or Universal Coordinated Time as it is now more accurately called, is January 1, 1970 at exactly midnight. This is what is known, in UNIX folklore, as the *epoch*. That is the time “when time began” on UNIX, which maintains time as the number of seconds elapsed since the epoch.

On most UNIX systems today, this counter is a 32 bit signed long integer. At some point, this counter will exceed the largest positive value that can be represented, and wrap around to negative value, giving the following times:

```
0x7FFFFFFFL = Tue Jan 19 03:14:07 2038 (GMT)
0x80000000L  = Fri Dec 13 20:45:52 1901 (GMT)
```

These dates and times, when converted to CST, correspond to the other dates and times in the quiz. ✒

***Monsieur Ex**, a mysterious Frenchman who claims to be an old editor and an expert in UNIX, will return again next time he’s asked questions, so don’t forget to write in, kids!* ***Gilbert Detillieux**, a French-Canadian of non-mysterious origins, is mysteriously still the MUUG newsletter editor.*

Mail Aliases on the MUUG Online System

By Gilbert Detillieux

UNIX systems that use *sendmail* as the mail transfer agent, such as the MUUG Online system (*mona*), provide a means of addressing mail to a mail *alias*, rather than only to a user's actual user ID on that system. This is a very useful feature, since it allows for mail to be redirected to the appropriate user or users, whether they are on the same system or not. It also allows for mail to be sent to fixed, easily remembered names, without worrying about the actual users' addresses.

On the MUUG Online system, we've made good use of this feature, and have set up several aliases for your convenience. These are stored in a file called `/etc/aliases`, if you want to take a look at them. However, this file only indicates what the aliases expand to, and not necessarily what they mean and when it's appropriate to use them.

Since many users aren't aware of these yet, or may be unclear on what the purpose of these aliases is, here is a summary of the important ones that we've set up. Please look over the list, so you'll know where to send e-mail when you need to. If you're logged onto *mona*, you just have to give the alias name as a mail address; on other systems, specify <*alias*`@muug.mb.ca`> as the address, where *alias* is the name of the mail alias to which you want to send.

postmaster
This is a standard alias on all systems that conform to the Internet mail standards. It refers to the local user that is responsible for administration of e-mail on this system. You can send mail to this person if you experience problems with delivery of mail, such as when a message bounces back to you for no obvious reason. (Check the message carefully first, to determine whether the local postmaster or the one at the destination address should be notified, if there is a problem, or to determine if it's something you can correct yourself.) The postmaster at a particular site can also be contacted to determine the e-mail address to use to send to a particular person at that site.

MAILER-DAEMON
This alias, also a standard on most Internet mail systems, is used by the mail software when it must generate its own messages. For example, it is used as the return address on messages that bounce back. This is usually the same person as *postmaster*. You will likely never have to actually send mail to this alias.

muug
This is essentially MUUG's "general delivery" address, and should only be used when there isn't a more appropriate address that you're aware of. This address goes to the postmaster, who can then forward the message as required.

info
This is for general group information enquiries. This might be used, for example, by someone who wants to know what MUUG is about, or where and when the next meeting is.

membership
This is for enquiries specifically about membership. For example, if a non-member wants to be added to the mailing list to receive application forms, if someone want to know about dues, or if someone has a question about his/her own membership status.

board
This is for communicating with all board members. This is used extensively by board members to keep each other informed, but can be used by anyone who'd like to address the MUUG board. Discussions about by-laws, policy, special events, or ideas for new initiatives are encouraged.

cuc
This is for the MUUG Online computer use committee. Again, this is used mainly by committee members, but is there for anyone who'd like to address the committee on issues of computer usage policy and guidelines.

admin
This is the small, busy group of users on mona that act as system administrators. Use this alias to report problems with the system, or other things about the system that should be brought to the attention of the administrators. You can also use this alias when you need to ask a question about using the system, but this should be one of your last recourses. Make sure you've tried other ways of getting help, such as the "help" and "man" commands, as well as the "news" and "gopher" commands. You might also want to try posting a question to one of the local network news groups, such as "muug.general" or "muug.help" (the latter would be better). That way, others can help as well, and you'll likely get the answer you want more quickly.

uucpadmin
This is the person responsible for administration of UUCP on mona. Questions or problems regarding setup and configuration of a UUCP connection to mona can be addressed to this alias.

editor
This is the current MUUG Lines newsletter editor-in-chief. You can mail enquiries about the newsletter, or (better yet) articles you'd like to submit, to this address.

muuglines
Same as above.

edcom
This is the group of people on the MUUG Lines editorial committee. This includes the editor, as well as several helpful assistants. The alias can be used by anyone, but is used primarily by committee members so they can easily communicate to each other. (Actually, it's used mainly by the editor, when he wants to gripe about the lack of submitted articles, or the looming deadlines.)

mailing
This is the group of people who help with stuffing envelopes and mailing out the newsletter. The alias is used mostly to remind volunteers of upcoming mailings, and to discuss scheduling.

m-ex
This is the alias for our mysterious old French friend, Monsieur Ex. You can mail him questions, or bits of trivia, about UNIX. These will be answered either by e-mail, or in the "Ask Monsieur Ex" column in the newsletter, or both. (He likes mail, so don't be shy.) ✒

Configuring UUCP, Part One

By Gord Tulloch <gordt@cyberspc.muug.mb.ca>

Introduction

UUCP means *UNIX to UNIX Copy Program*, but it's also a whole subsystem of programs intended to provide an entire range of services on UNIX and other systems. While UUCP is a standard on all UNIX derived systems, it is also available on other systems such as VMS, MS-DOS, and OS/2. I run stock (referred to as HoneyDanber after the authors) UUCP on my IBM RISC System/6000 Model 220 at the office and the OS/2 based UUPC on my 386 at home.

I've always wanted to work with UUCP, but have never worked with more than one system at a time — *mona*'s emergence has allowed me a base point on which to base an entire network of systems. This is fortunate as I've since begun to work setting up UUCP with a number of other UNIX systems owned by my clients.

The UUCP system is also part of a standard package on most modern UNIX systems named BNU, for Basic Networking Utilities. UUCP is the first and simplest part of BNU, which also includes TCP/IP based networking and SLIP, Serial Line Internet Protocol, an asynchronous dial-up method of linking computers via TCP/IP into a Wide Area Network (WAN).

Once I realized that I now had all the tools to set up a WAN encompassing my home and office machines as well as customer machines, I began to link up the systems for which I am responsible one-by-one. *Mona*, of course, provides a link into the Internet for my personal systems. These articles, three in all, cover what I experienced while setting the systems up. This first article talks about setting up basic UUCP links between UNIX systems (with a little bit about my home OS/2 system for good measure) with subsequent articles describing more hairy operations — setting up for properly routed electronic mail (with domain based addresses) and setting up to receive USENET news groups.

UUCP — What Does It Do For Me?

UUCP consists of a number of tightly interwoven programs which provide various functions for the system. They can be split into three basic classes — file transfer, mail handling, and remote execution of commands. The first two are in most cases one and the same since what is mail besides a file which receives special handling at either end of the transfer?

The actual `uucp` command (note the lower case) simply allows the user to do a `cp` command across multiple systems. Using UUCP allows you to:

- Transfer files from your system to another with a simple command line — as opposed to sending files via *kermit* or *Zmodem*, UUCP allows you to “batch” files for transmission and have them sent to your system automatically when convenient. For example, you might wish to set up UUCP to poll *mona* hourly to pick up files and mail. The files will be sent in the next UUCP transfer.
- Send and receive mail from your system to other systems, either through basic UUCP or, if you are connected to an Internet machine such as *mona*, through the Internet.
- Send and receive USENET newsgroups, a special class of mail.

Configuring UUCP

To configure UUCP, you must first edit a number of standard files on your system. In AIX (and I assume most recent UNIX systems that are moving to OSF/1 compliance) these files are located in the `/usr/lib/uucp` directory. Let's look at each file in depth and see what has to be done. I'll use a link with `mona` in all cases since that's what most people will be doing first anyway!

/usr/lib/uucp/Devices

The `Devices` file is used to inform the UUCP system what devices are available in a pool at what baud rates and so on. When any program in the UUCP system (`uucico` or `cu`, for example) requires a modem, they consult this file, lock the device, and start operations. Here's an example of the entries in this file:

```
ACU tty1 - 2400 hayes
Direct tty2 - 19200 direct
```

I won't go through these entries in detail due to space limitations; consult your manual for details.

/usr/lib/uucp/Systems

The `Systems` file lists the systems that you are able to connect with directly. They can include systems dialed into via modem, direct serial link-ups as well as TCP/IP networked systems. For example, the following entry is required for `mona` (on a single line):

```
mona Any ACU 2400 275-6150 "" \d\d\r\c classn
\p\pmuug ogin: \036\c > set\sxb=on >
set\secmc=dis > res resumed \pUxxxxxx\n\c
assword: xxxxxxxx\n\c
```

The first field gives the system name, the second the times that the system can be called, the third specifies what kind of connection is required to link to the system (ACU means Automatic Calling Unit, or some such!), fourth the baud rate, next the phone number (if an ACU setup) and the rest is a “chat script,” which is in the “get this/send this” pair format. When the calling UUCP program (`/usr/lib/uucp/uucico`) receives the first text string it will send the second. Since UUCP uses the same login people do, but with special accounts, this is simply how to log into the system.

You will note that due to the 7 bit Telnet link that must be reconfigured on `mona` to do 8 bit transfers the `mona` sequence is far more complicated than normal — here's one for a UNIX system that's a direct dial to a tty port:

```
systemx Any ACU 2400 555-1212 ogin:--ogin:
nuucp assword: ThisPass
```

This script waits for the login prompt (sending a CR if it doesn't get it in a few seconds and waiting again), logs in under the `nuucp` account (commonly used for “anonymous,” non system-specific UUCP logins) and upon receipt of the password prompt sends the password. The called system then invokes the `uucico` program which initiates the machine to machine transfers. More on `uucico` later.

/usr/lib/uucp/Permissions

This file defines what permissions the calling (or called!) system has on your machine. Two sections are required, MACHINE and LOGNAME, which define what the system can write to on your disk, what commands it can execute (via the uux remote execution command) and other functions. Here's my permissions for mona:

```
    MACHINE=mona REQUEST=yes READ=/ WRITE=/ \
        COMMANDS=rmail:uucp:rnews:uux
    LOGNAME=mona REQUEST=yes SENDFILES=yes \
        READ=/usr/spool/uucppublic \
        WRITE=/usr/spool/uucppublic \
        COMMANDS=rmail:rnews:uux:uucp:news \
        CALLBACK=no
```

/usr/lib/uucp/Poll

This file is simply a list of systems with times they can be polled. UNIX systems normally have a cron job that runs a batch file to look at this file and run uucico on the system if a poll is required. Look at the crontab for the uucp user on your system — the command

```
    su uucp -c "crontab -l"
```

should list it for you (if you are the root user anyway!).

The Poll file consists of entries such as:

```
    mona        0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
    15 16 17 18 19 20 21 22 23
```

AIX requires a tab between system name and the hours to poll. As you can see, my RS/6000 polls mona every hour on the hour daily.

Testing UUCP

You should first send mail on mona to the uucpadmin alias to request a userid and password for your system — this information must go into your Systems file.

While you await our busy Sysop's reply, test your installation by entering

```
    cu mona
```

This should invoke the simple cu comm program and dial into the named system (in this case, mona). The cu program can also be used as a terminal program for your modem by entering speed and port on the command line. See your manual for details. If this doesn't work, uucp will fail also.

If and when this works and you have your system userid, you can now try a UUCP poll:

```
    /usr/lib/uucp/uucico -smona -r1 -x9
```

which calls the system specified in the -s parameter, with your system acting as the "master," with a debugging level of 9 so you can see what's happening (or not happening!).

Sending Files Via UUCP

Once you are up and running, you can test uucp on mona by entering

```
    uucp .cshrc cyberspc!~/
```

where cyberspc is the name of my system. Note that the destination is in bang (!) path format, which specifies the destination system, an exclamation point (bang) and the target directory. The characters ~/ are shorthand for the default spool directory on the destination system. This command instructs mona to create a batch entry to send your .cshrc file to the public directory on the cyberspc system (normally /usr/spool/uucp/uucpppublic, but on my OS/2 system it's D:\UUPC\SPOOL\PUBLIC). You may have to set the rwx bits on your .cshrc file to make sure uucp can access it — the command

```
    chmod 755 .cshrc
```

does this, giving rwxr-xr-x access to the file. You may wish to have uucp make a spool copy of your file so you can reset these bits immediately — the command would look like this:

```
    uucp -C .cshrc cyberspc!~/
```

The administrators would probably appreciate it if you didn't do this with 3 meg files!

Once you have this working, you are ready to implement mail. We'll look at this next time.

Sybase Database Objects

By Scott Balneaves

This article is somewhat specialized; however, since there are some people here at the group who use Sybase at their workplace, I hope that some of you will be able to use this.

This was one of my first forays into C++ several months ago. I'm still learning but I've done some slightly more complex things than this. However, this small program fragment was useful to me in solving a quick and dirty problem, and it demonstrates some of the fundamental aspects of C++, and Sybase.

Basically, what has been done here is to create a simple object class that allows you to hide some of the nastier bits of code that goes on in a Sybase query deep in the object. This results in a much cleaner API to Sybase. I'll step through the object a piece at a time, and comment on what it's doing.

How the Object Works

Because of the fact that the Sybase include files aren't in C++ (they aren't even in ANSI C — ugh!), we have to enclose them in the construct that you see at the top of the file:

```
    extern "C" {
    ...
    }
```

You will notice that our DBPROCESS and LOGINREC structures are in the private part of the object. This hides them from being seen from anything but this object. No need to worry when you have multiple threads about keeping the DBPROCESS pointers from clobbering each other.

I have two constructors. One is the constructor with no arguments, which simply does the dbinit() The second constructor accepts your userid and password for Sybase as arguments. You can either create the object with no args, and use the login() function, or pass the userid and password as part of the creation.

The bind and exec are fairly straightforward. If you are familiar with the Sybase API, these make sense. I've just eliminated some of the tedium.

The destructor at the end will automatically do the dbexit() for you when you pass out of the scope that the object was instantiated in. Or, you can explicitly call the logout if you like.

So what does it all look like? I've created a small mainline (and Makefile) to go along with this object. Just follow the comments, and I think you'll agree: this looks a lot less complicated than the normal C Sybase hoops you have to jump through.

I hope to use this as a prelude to an introduction to a series of C++ articles, much along the lines of the RPC series I did last year. I hope people will be interested in this series of articles. If you'd like to drop me a line, you can from mona. Simply mail <sbalneav@silver.cs.umanitoba.ca>, and I'll do the best I can to help you.

The source code referred to in this article is available via anonymous FTP on mona.muug.mb.ca, *under the directory* pub/muuglines/source/sybase-obj.

THE FORTUNE FILE CLASSIC

Selecting a Programming Language Made Easy

By Daniel Salomon and David Rosenblueth

Departments of Computer Science, University of Manitoba and Acadia University (resp.)

Originally appeared in the September 1986 issue of SIGPLAN Notices, *Volume 21, Number 9. (Reprinted with author's permission.)*

With such a large selection of programming languages it can be difficult to choose one for a particular project. Reading the manuals to evaluate the languages is a time-consuming process. On the other hand, most people already have a fairly good idea of how various automobiles compare. So in order to assist those trying to choose a language, we have prepared a chart that matches programming languages with comparable automobiles.

Assembler
A Formula I race car. Very fast, but difficult to drive and maintain.

FORTRAN II
A Model T Ford. Once it was the king of the road.

FORTRAN IV
A Model A Ford.

FORTRAN 77
A six-cylinder Ford Fairlaine with standard transmission and no seat belts.

COBOL
A delivery van. It's bulky and ugly, but it does the work.

BASIC
A second-hand Rambler with a rebuilt engine and patched upholstery. Your dad bought it for you to learn to drive. You'll ditch it as soon as you can afford a new one.

PL/I
A 1968 Chevrolet Impala convertible with automatic transmission, a somewhat peeling two-tone paint job, tires with 4-inch whitewall trim, chrome exhaust pipes, BIG tailfins, and fuzzy dice hanging in the windshield.

C
A black Camaro, the HE-man car. Comes with optional seat belts (lint and dbx), and required fuzz-buster (escape to the assembler).

C++
A pink Camaro with lace ruffled curtains in the windows and five training wheels.

ALGOL 60
An Austin Mini. Boy, that's a small car!

Pascal
A Swiss hand-made Volkswagen Beetle. It's small but sturdy. Was once popular with intellectuals and other quiche eaters.

ALGOL 68
An Aston Martin. An impressive car, but not just anyone can drive it.

SmallTALK
John Lennon's psychedelic Rolls-Royce. There is no question that the car is good, but not everyone has the guts to drive something flashy around. Besides, what do you do with the photocopier in the trunk, anyway?

LISP
A BMW 735i. Becoming trendier and trendier, faster and faster, larger and larger, but still hasn't quite got it there. So many models, colors, and options are available that you simply try to choose what's "in" this week, and sometimes miss. Seat belts not available with most models, and some of the Japanese versions seem to be missing the engine.

T
A pickup truck with an ergonomic cabin and a Honda Civic engine. Very easy to learn, but then what?

PROLOG
A Jaguar. "The" car on the Dom Perignon-caviar circuit, but the casual drivers can neither afford to buy, nor can they afford to maintain it, though they all wish they could one day get their hands on it. Comes in a lot of colors and with so many options that it takes books just to list some of the capabilities.

FLAVORS
An ice-cream truck that has just been in an accident. You never really know all the mixins you are getting with every flavor, but you can break a tooth on some of those nuts.

MACSYMA
A Jeep-based limousine. Absolutely ridiculous to look at, guzzles the gas like crazy, but will go through anything given the proper driving techniques.

FORTH
A go-cart.

LOGO
A kiddie's replica of a Rolls-Royce. Comes with a real-looking engine and a working horn.

APL
A double-decker bus always painted deep BLUE. It takes rows an columns of passengers to the same place at the same time; but, drives only in reverse gear, and is instrumented in Greek.

LOOPS
A top-of-the-line Winnebago. The car has got so many amenities that you could probably live in it, but boy does it guzzle the gas!..

Ada
An army-green Mercedes-Benz staff car. Power steering, power brakes and automatic transmission are all standard. No other colors or options are available, except by passing a Bill through Congress. If it's good enough for the generals, it's good enough for you. Manufacturing delays due to difficulties reading the design specifications are starting to clear up.

Submitted by Tom Dubinski

From: leech@cs.unc.edu (Jon Leech). Source: rec.humor.funny

After the 3rd or 4th consecutive daily crash of one of our critical (and incredibly flaky) DECservers, the following was posted to a local newsgroup by a frustrated user:

As a friend once said, "If the car industry behaved like the computer industry over the last 30 years, a Rolls-Royce would cost $5, get 300 miles per gallon, and blow up once a year killing all passengers inside."

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 March 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 UniForum Canada's Annual UNIX Show which is to be held this coming April.

Agenda

for
Tuesday, February 9, 1992, 7:30 PM
St-Boniface Hospital Research Centre
Theatre, Main Floor, 351 Taché

1.	Round Table	7:30
2.	Business Meeting a) President's Welcome b) New Business	8:00
3.	Break	8:15
5.	Presented Topic Electronic Publishing with TEX and LATEX By Michael Doob, University of Manitoba *As a counterpoint to last month's presentation on FrameMaker, this month, we will look at a very popular batch-oriented tool for electronic type-setting. This system is widely used by many academics, particularly those in mathematics and other branches of science and engineering, due to its powerful equation layout language. Michael Doob uses this system, among other things, to edit the Canadian Math Society Journal. 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, March 9, at 7:30 PM. Meeting location will be given in the March newsletter. The meeting topic is also to be announced.

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 in the pipeline, and a few more that should be on the way, including a review of *The Whole Internet User's Guide & Catalog*. Of course, I can always use some more material, especially shorter articles – half a page to one page 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.