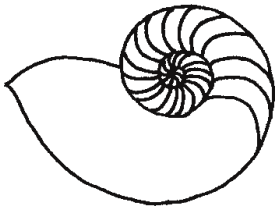
Volume 5, Number 1 November 1992 $2.50



MUUG Lines

Manitoba UNIX® User Group

Newsletter of the Manitoba UNIX® User Group

Tips for Downloading from MONA

by Gilles and Gilbert Detillieux

(with contributions from Andrew Chan and Andrew Trauzzi)

Many users have complained of difficulties with downloading files from the MONA system to their own machines. In fact, this is our most frequently asked question. The complications stem from the fact that there are several protocols to choose from, and the fact that the MONA system does not have direct serial links to the modem pool.

To download files from an interactive session, you have your choice of the Kermit, Xmodem and Zmodem protocols. To use Kermit, the command is simply "`kermit`". Type "`man kermit`" to get more information on this. To use Zmodem, use `rz` to receive (upload) files, and `sz` to send (download) them. Similarly, for Xmodem you use `rx` and `sx`. There are "`man`" pages for all of these.

The connection from UMnet to mona is via telnet, which is not an 8 bit transparent link by default. This is why protocols like Xmodem, Zmodem, and UUCP won't work without changing some of the connection parameters first (they expect a transparent link, so they can send arbitrary binary data).

With `sz/rz`, the simplest way around this is to specify the "`-e`" option when you run `sz` or `rz`. This will force it to escape certain special characters that might otherwise be intercepted, causing the transmission to fail. Some users have experienced problems even with the "`-e`" option. We believe that this is due to some half-baked implementations of Zmodem in some communications packages. We recommend TELIX or PROCOMM for PCs, or using the DSZ package with a communication program that supports external protocols. For the Macintosh, we recommend Zterm.

With Kermit, the telnet link should not pose a problem, as long as you configure it properly to work over a 7-bit link (this is usually the default setting). Another precaution is that Kermit has TEXT and IMAGE transfer modes, similar to FTP. Make sure the you select the right mode for the type of file you are transferring.

With other protocols (or to get `sz/rz` to work without the "`-e`"), you have to escape to the UMnet telnet command mode after having established the connection, type in a couple commands there, to change the connection to an 8 bit transparent one, then resume your session. You also have to dial in to 275-6150 (rather than 6100) to be able to use this. There are also a few C-Shell parameters that have to be set for this to work. The procedure is as follows.

Dial in to 275-6150. When connected, type "`muug`" and login as usual. At the C-Shell prompt, type:

```
set ignoreeof
set filec
```

Then, escape to telnet by hitting <Control-^>. At the ">" prompt, type:

```
set xb=on
set ecmc=dis
res
```

This should return you to your session, with an 8-bit transparent link. If your tty seems to be in a strange state (e.g. no response to a <Return>), try typing <Control-J>; if the system responds, type "`stty sane`" followed by another <Control-J> (not <Return>!). However, the `ignoreeof` and `filec` options above should prevent this from happening.

From this point on, until you log out, any communication protocol that requires 8-bit transparency should work correctly.

INSIDE THIS ISSUE

Newsletter Editor's Ramblings 2
President's Corner 3
Software Review: Patch 4
The Fortune File 5
MUUG Newsletter Deadlines 5
Industry: Memory Price Skyrockets 6
Favourite Commands: find 6
Finance: Year-end statements 7
Open Systems Fall Seminar 7
Oct. 13th Meeting Minutes 8
Nov. 10th Meeting Agenda 8

THIS MONTH'S MEETING

Meeting Location:
Our next meeting is scheduled for Tuesday, November 10, at 7:30 PM. The meeting will be held at the theatre of the St-Boniface Hospital Research Centre, just south of the hospital itself, at 351 Taché. You will likely have to sign in at the security desk at the entrance. The theatre is on the main floor; ask for directions at the security desk if you need them.

Meeting Agenda: See inside for details.

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

Fees, Renewal Dates, etc.

By Gilbert Detillieux

Welcome to another year. Hopefully, by now, you've sent in your renewal form(s) and are all paid up for the year. If not, your mailing label should clearly indicate if you are due for renewal. Now that we're going to staggered renewals, your renewal date will always be one year after you joined or renewed. (For many of you, this will still be October, as before.)

With all these changes, and with the confusion caused by our former pro-rated membership dues, it may not be clear to you when you have to renew. Again, check your mailing label – it will indicate your renewal date. Some of you who joined toward the end of summer, but paid for a full year, have had your renewal date moved up to next October, so you're paid up for the current year. Some of you also sent in your renewal payment, despite not being due for renewal yet – you will most likely receive a refund shortly.

Also please not that if you've recently paid for a full year's membership in MUUG, but haven't paid for your MONA account, that amount is now due. (You will likely receive an e-mail notice to that effect.) To renew your MONA account, fill in the MONA account application form (even if you filled one in initially), completing all of the personal information, and indicating your current userid. You can omit the password if your account is already in use, and has a secure password. Accounts that have not been renewed by the next meeting date, November 10, will be cancelled.

With that out of the way, I hope you enjoy this issue of the newsletter. Thanks to all the usual contributors. Thanks also to those who have offered (or will offer) to help by writing articles, selecting articles to reprint, and/or otherwise help out on the new editorial committee that will be set up. This committee will help make the task of putting the newsletter together more pleasant, and will hopefully result in more diversity and quality in the newsletter's contents. If you are planning to contribute to the newsletter or help out on the committee, please note the schedule of deadlines on page 5.

Also, don't forget about the upcoming Open Systems seminar, described on page 7. See you at the 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

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.

Welcome to a new MUUG Year

By Susan Zuk, President

Welcome to a new MUUG season complete with a new board with some new faces. I would like to extend a warm welcome to Bary Finch and Paul Hope. I look forward to working with you during the year. If this year is anything like the last one, you should find your involvement very fulfilling. For a list of the board and their positions, please read on.

At the November meeting, the board members will be providing you with an overview of their responsibilities and will be requesting volunteers to help them achieve their goals. Please think of what you would like to do this year to help the group. There will be sign-up sheets at the meeting.

A few weeks ago, I attended the National Council Meeting in Quebec City. This meeting, which is held twice a year, is comprised of representatives from the various UniForum Canada affiliates and also included some members of the UniForum Canada Board.

As chairperson and participant, I found the session very interesting and helpful. The day began with the affiliates providing an update on their current and future activities. Some affiliates have very stable groups while others are just starting out. Calgary, Ottawa, Montreal, Quebec City, and our group seem to be very well established and growing. New groups such as Toronto and London are progressing and appreciated much of the information we were able to share with them.

It is very interesting to see how the groups operate and the similarities and differences we have in program styles as well as membership makeup. For example, Calgary and Ottawa are very business oriented while Edmonton and Quebec City are more geared to technical users.We saw that all groups go through the same organizational growth stages as the group matures. This is where sharing information has a great effect on helping new affiliates.

One of the roles of the National Council is to help UniForum Canada fulfill its National objectives. As well, the National Council provides its view on what the affiliates require to meet their objectives both as a body and per affiliate. The National Council helps UniForum Canada see what the needs are on a country wide basis. To help UniForum Canada and the affiliates become a more cohesive group, a task force has been formed to review the way in which UniForum Canada will interact with the affiliates. A method of how each affiliate can provide assistance to the National Board will also be outlined. The document will be completed by the end of February. If members are interested in getting involved with the National programs, let me know and I will provide you with some of the areas where assistance is required.

UniForum Canada also has a newsletter which is sent out every 2 months. This newsletter is called README. If you would like to submit articles to be printed in this publication send your information to: README Article, Suite 3-46 Three Valleys Drive, Don Mills, Ontario, M3A 3B5; FAX to (416) 362-8324 (Greg Woods); or e-mail to <uniforum@robohack.UUCP>. The information will be much appreciated.

Note on your calendar the upcoming fall seminar called "Open Systems - Getting Past All the Hype" and co-sponsored by the Canadian Information Processing Society (CIPS). It is being held on Thursday, November 19 at the Winnipeg Convention Centre. This will give you an opportunity to see what different companies, both local and international, are doing when it comes to Open Systems. See details in the brochure you just received as well as in this newsletter. Students should also note the great price which is being offered to them. This seminar will help you see what organizations are doing and what type of things you may be involved with during your career. At $35.00 for students you just can't go wrong.

To take a look at future activities, we would like to tell you about a comedian who is coming to town in January. A fellow named Jeff Armstrong or "Saint Silicon" will provide you with his evangelistic views of the computer world. Jeff states that he is the only computer comedian in the world. This special engagement will be on January 20, 1993. Mark your calendar and stay tuned for more information!

This month we will be visited by Jim Baglot of Frame Technology Corporation, from Vancouver. Jim's presentation will include a live demonstration of their product. We look forward to seeing you on the 10th and helping us to give Jim a warm Winnipeg welcome.

MUUG Board

Meet the New Board for 1992-1993

Susan Zuk – President
Bary Finch – Vice President
Rick Horocholyn – Treasurer
Roland Schneider – Secretary
Gilbert Detillieux – Newsletter Editor
Gilles Detillieux – Publicity Director
Richard Kwiatkowski – Membership Director
Paul Hope – Meeting Co-ordinator

Patch

by Roland Schneider

One of the biggest problems with using public domain software is that it keeps changing. Every time the author adds a small enhancement, or fixes a bug, out comes another new version. Keeping up with the latest update quickly becomes a tiresome chore, especially if the entire distribution has to be downloaded, re-configured, re-compiled and re-installed each time a little bug is fixed. For large programs, the downloading itself becomes a major task.

Enter the *patch* program, written by Larry Wall. The UNIX utility *diff* compares two files and reports the minimal set of differences. *Patch* can take one of those files and the *diff* output and recreate the other file. If the author of the PD program you want to update ran *diff* to record the changes he (or she) made to the program, all you need is those changes and the *patch* program. This saves downloading the entire distribution again, and preserves your local configuration or other changes to the original distribution. Most changes to larger PD programs are distributed this way.

Time for an example. Let's say I wrote the following paragraph as part of a larger article:

```
This is a demonstration
of the public domain
patch program wrote
by Larry Wall. This must
rank as one of the most
useful utility programs
ever released for free.
```

After I had submitted the article, I noticed my grammatical error, corrected it, and ran *diff* to compare the original article to the corrected article:

```
diff article fixed_article > changes
```

The *diff* output is:

```
3c3
< patch program wrote
—
> patch program written
```

All the newsletter editor has to do now is type "`patch article changes`" producing the output

```
Hmm...  Looks like a normal diff to me...
Patching file note using Plan A...
Hunk #1 succeeded at 3.
done
```

This will apply the contents of `changes` to `article`, converting it into an exact replica of my file `new_article`. The original `article` is renamed `article.orig` just in case something goes wrong and it has to be retrieved.

Patch works even better if you use the `-c` option on *diff* to produce a diff file containing the context around all changes. Because the context diff contains the file names involved, patch can simply be run as "`patch < changes`". Patch can also use the context information to deal with cases where the file being patched has had some unrelated local modifications.

I have also used patch for a number of other purposes, including maintaining operating system configuration files. Whenever I change a file in `/etc` or somewhere else on my system, I first make a copy of it and name it *file*`.orig`. When I've made my changes, I use `diff -c` to record them and store them in a safe place. After a new version of the operating system has been installed, I can use *patch* to re-apply my changes, and allow its intelligent handling of changes in the files' contexts to deal with modifications to the configuration files made by the software vendor. At the very least, I have the human-readable output of *diff* to allow me to make the changes manually if *patch* can't manage on its own. After patch has run, I again have the original files named *file*`.orig`.

Facts in brief

Runs on:	almost any UNIX machine
How to get it:	lots of ftp sites, MONA `/usr/local/src/patch.tar.Z` (older version)
Disk space:	159K for the source
Usefulness:	High
Learning:	Easy
Installation effort:	Low (highly automated self-configuration)
Cost:	Free

UNIX TRAINING

UNIX Shell Programming (System V. Release 4.0)

This course teaches the skills necessary to code, debug and use shell programs. Shell programs enable users, programmers, or system administrators to automate system tasks and develop prototype program solutions. This course combines lecture with laboratory time to allow the students to explore the capabilities of shell programming.

This course is being held November 30 to December 4, at the Unisys Winnipeg office, 1000-1661 Portage Avenue. Contact Susan Zuk or Debbie Harapiak, at 788-7400, for details.

The Toaster

Submitted by Ken Stewart (original source unknown)

Once upon a time, in a kingdom not far from here, a king summoned two of his advisors for a test. He showed them both a shiny metal box with two slots in the top, a control knob, and a lever. "What do you think this is?"

One advisor, an engineer, answered first. "It is a toaster," he said. The king asked, "How would you design an embedded computer for it?" The engineer replied, "Using a four-bit microcontroller, I would write a simple program that reads the darkness knob and quantizes its position to one of 16 shades of darkness, from snow white to coal black. The program would use that darkness level as the index to a 16-element table of initial timer values. Then it would turn on the heating elements and start the timer with the initial value selected from the table. At the end of the time delay, it would turn off the heat and pop up the toast. Come back next week, and I'll show you a working prototype."

The second advisor, a computer scientist, immediately recognized the danger of such short-sighted thinking. He said, "Toasters don't just turn bread into toast, they are also used to warm frozen waffles. What you see before you is really a breakfast food cooker. As the subjects of your kingdom become more sophisticated, they will demand more capabilities. They will need a breakfast food cooker that can also cook sausage, fry bacon, and make scrambled eggs. A toaster that only makes toast will soon be obsolete. If we don't look to the future, we will have to completely redesign the toaster in just a few years."

"With this in mind, we can formulate a more intelligent solution to the problem. First, create a class of breakfast foods. Specialize this class into subclasses: grains, pork, and poultry. The specialization process should be repeated with grains divided into toast, muffins, pancakes, and waffles; pork divided into sausage, links, and bacon; and poultry divided into scrambled eggs, hard-boiled eggs, poached eggs, fried eggs, and various omelet classes."

"The ham and cheese omelet class is worth special attention because it must inherit characteristics from the pork, dairy, and poultry classes. Thus, we see that the problem cannot be properly solved without multiple inheritance. At run time, the program must create the proper object and send a message to the object that says, 'Cook yourself.' The semantics of this message depend, of course, on the kind of object, so they have a different meaning to a piece of toast than to scrambled eggs."

"Reviewing the process so far, we see that the analysis phase has revealed that the primary requirement is to cook any kind of breakfast food. In the design phase, we have discovered some derived requirements. Specifically, we need an object-oriented language with multiple inheritance. Of course, users don't want the eggs to get cold while the bacon is frying, so concurrent processing is required, too."

"We must not forget the user interface. The lever that lowers the food lacks versatility, and the darkness knob is confusing. Users won't buy the product unless it has a user-friendly, graphical interface. When the breakfast cooker is plugged in, users should see a cowboy boot on the screen. Users click on it, and the message 'Booting UNIX v. 8.3' appears on the screen. (UNIX 8.3 should be out by the time the product gets to the market.) Users can pull down a menu and click on the foods they want to cook."

"Having made the wise decision of specifying the software first in the design phase, all that remains is to pick an adequate hardware platform for the implementation phase. An Intel 80386 with 8MB of memory, a 30MB hard disk, and a VGA monitor should be sufficient. If you select a multitasking, object oriented language that supports multiple inheritance and has a built-in GUI, writing the program will be a snap. (Imagine the difficulty we would have had if we had foolishly allowed a hardware-first design strategy to lock us into a four-bit microcontroller!)."

The king had the computer scientist thrown in the moat, and they all lived happily ever after.

MUUG Lines Newsletter Deadlines for 1992-1993

			Deadlines		MUUG Monthly
Issue Number	**Month**	**Articles**	**Final Copy**	**Mailing**	**Meeting Date**
Volume 5, No. 01	November	October 17	October 24	October 31	November 10, 1992
Volume 5, No. 02	December	November 14	November 21	November 28	December 8, 1992
Volume 5, No. 03	January	December 19	December 26	January 2	January 12, 1993
Volume 5, No. 04	February	January 16	January 23	January 30	February 9, 1993
Volume 5, No. 05	March	February 13	February 20	February 27	March 9, 1993
Volume 5, No. 06	April	March 20	March 27	April 3	April 13, 1993
Volume 5, No. 07	May	April 17	April 24	May 1	May 11, 1993
Volume 5, No. 08	June	May 15	May 22	May 29	June 8, 1993
Volume 5, No. 09	September	August 21	August 28	September 4	September 14, 1993
Volume 5, No. 10	October	September 18	September 25	October 2	October 12, 1993

Mailing deadline is always 10 days prior to meeting date, to ensure sufficient advance notice of motions, etc. All copy (ads, columns, meeting notices, etc.) must be submitted by noon on the final copy deadline date, which is one week prior to the mailing deadline. Articles and any other material that isn't time sensitive must be submitted by the articles deadline, which is one week prior to the final copy deadline.

INDUSTRY

Memory Price Skyrockets!

By Mark H. Anbinder & Adam C. Engst

Reprinted from TidBITS. Copyright 1990-1992 Adam & Tonya Engst.

Elephants beware! The price of memory is shooting up! This is largely due to a tariff levied on Korean-imported memory chips, such as from Hyundai and Samsung. The US Commerce Department just ruled in favor of US memory manufacturer Micron Technology's complaint that the South Korean manufacturers were selling chips for less than it cost to make them. Samsung and others have posted a bond to continue importing chips, reportedly at the same price as before the ruling. As a result of the ruling, Japanese chip prices are going up as distributors bid more and promise quicker payments for materials to build SIMMs.

What this means for users is that SIMM prices will rise and supply will become tight. As with hard drives and other components, major manufacturers like Apple tend to get "first dibs" on parts used in building their computers, so other vendors may be left with the scraps. End-user prices on computers are unlikely to rise as an immediate result of this shortage, especially since Apple buys memory from many different sources around the world, but prices that might have come down in the near future probably won't.

Memory industry insiders are estimating that this backlog, and resulting price increases, will last anywhere from several weeks to several months, with a common estimate of about four months. Chances are fairly good that, for at least the next few weeks, prices will increase steadily. Paul McGraw of APS feels prices will rise quickly, level off for a while, and then gradually descend to perhaps $30US/MB, although probably not as far as the $25US/MB range of last week. MacWEEK quoted Mike Frost, president of TechWorks, as saying " This could create a shortage like back in 1988 when prices shot through the roof. The savvy corporate buyer will buy supplies now to cover the next several months."

This situation may serve to shake out some of the cut-rate memory vendors, who will be unable to retain customer loyalty as their prices increase dramatically and delays mount. One possible result is that, even when things settle down, the final memory prices may be substantially higher than they are now. This will be due only partly to the increased taxes, and partly to a reduction in competition. Although it may already be too late, don't put off investigating RAM prices if you're thinking about buying memory in the next few months.

Information from: Paul McGraw, APS vice-president
Related articles: MacWEEK — 26-Oct-92, Vol. 6, #38, pg. 1

MY FAVOURITE COMMAND

find – Find Files by Name, or by Other Characteristics

by Roland Schneider

If you're like me, and have been using the same computer system for a number of years, you probably have hundreds, if not thousands, of files in your account. (I have over 6000 at the moment) How do you find that one file you **know** you saved three months ago, under a nice descriptive name, but somewhere deep in an obscure directory?

The "`*`" wildcard character works *within* the current directory, but does little when you want to recursively search an entire directory tree. Other operating systems have a recursive wildcard character, UNIX has the `find` command.

Let's say I remember that the word "`internet`" was part of the name of the file I was trying to find. I would change to my root directory and type:

```
find . -name '*internet*' -print
```

The first argument to find is the name of the base directory to search from — in this case the *current* directory, indicated by a dot. The rest of the arguments are logically *and*ed together to form an expression. Parts of the expression may have side effects, like the "`-print`" in the example. Remember that the wildcard characters are special to the *shell,* and must be quoted so that the find program receives them undisturbed.

The find command will now print the name of every file in my directory hierarchy containing the string "`internet`". Unfortunately, I must have had the Internet on my mind recently, since I got a list of over 100 files. Luckily, I know that the file I want was over 100K long, so I insert the condition "`-size +100000c`" and try again. The plus sign means *greater than* in this case.

The arguments to find can get pretty hairy. Let's say I want to clean up my directories, and delete all backup files ending in "`.bak`" or "`~`" which have not been modified in the last month, and print the names of the files as they are deleted. I would use

```
find . \( -name '*.bak' -o -name '*~' \) -mtime
+30 -print -exec rm '{}' \;
```

Ugly, isn't it? The `\( \)` pair is needed to make sure that the *or* of the two name conditions, separated by "`-o`", is interpreted in the correct order. "`-mtime +30`" selects files whose modification date is more than 30 days ago. The "`-exec`" obviously executes a command, in this case "`rm`", and the "`{}`" is converted to whatever filename passed the other tests. The execution arguments are terminated by the semi-colon, which must be escaped by the backslash to keep it from being interpreted by the shell.

There are lots of other conditions which can be added to the find expression. The online UNIX manual gives a complete, although brief, description along with a few examples. By the way, unless you're unusually brave, it's always a good idea to run find with a "`-print`" to test a complicated expression before using something like "`-exec rm`"!

FINANCE

Manitoba UNIX User Group

Year-end Financial Statements

By Rick Horocholyn, Treasurer

Balance Sheet
As of 1992/09/30

Assets	
Cash	304.00
Chequing	900.92
Investments	8000.00
Equipment	100.32
Other Assets	27.36
Total Assets	**9332.60**
Liabilities	
Accounts Payable	276.56
Total Liabilities	276.56
Equity	
Net income (this year)	8544.71
Retained Earnings (previous year)	511.33
Total Equity	9056.04
Total Liabilities and Equity	**9332.60**

Profit and Loss Statement
For the 12 months ending 1992/09/30

Revenues	
Membership Fees	2056.35
Other Revenue (Fall Symposium)	8000.00
Total Revenues	10056.35
Expenses	
Newsletter (paper & postage)	880.32
Advertising	63.00
Entertainment (Symposium wind-up)	59.55
Speaker Fees & Expenses	109.44
Bank Charges	4.83
Postal Box Rental	80.25
Miscellaneous Expenses	314.25
Total Expenses	1511.64
Net Income	**8544.71**

***Rick Horocholyn** works for Manitoba Hydro. He's been a member of the group for several years, and has been the group's treasurer since October, 1991.*

Fall Seminar Announcement

A CIPS/MUUG SEMINAR

Open Systems – Getting Past the Hype

Thursday, November 19, 1992
Winnipeg Convention Centre
8:30 AM - 4:00 PM

What does "*Open Systems*" really mean? How may it fit into your organization? Is "*Open Systems*" a reality and is it right for everyone? How does UNIX fit or does it not fit into this scenario? These are some of the questions which will be discussed at the joint Canadian Information Processing Society / Manitoba UNIX User Group fall seminar on Thursday, November 19th, 1992.

An industry specialist and author of *The Open Systems Handbook*, Mr. Tom Wheeler, will provide a current and future view on the subject followed by four real life case studies. These case studies will provide you with the opportunity to see how Open Systems are being implemented in organizations such as: The Canadian Wheat Board, Bell Corporation and DSC (Data Switch Corporation). You will see if they are vanilla Open Systems environments or a mixture and why this is so.

We invite you to come and hear what people are really thinking and doing with their computer systems, to make sure they stay competitive and productive in today's quick paced world.

This is a unique opportunity for you to hear a diverse array of speakers, both local and international, discuss their experiences. You will also have an opportunity to ask them questions you have been wrestling with.

For a registration form please FAX your request to Susan Zuk at 788-7450, or if you have questions, call her at 788-7312.

MUUG Meeting Minutes

Annual General Meeting – 1992
Tuesday, October 13, 1992, 7:30 PM
Room 234B, Engineering Bldg.
University of Manitoba
Ft. Garry Campus

1. New By-laws Adopted
Pursuant to the notice of motion in the Annual Meeting notification contained in the October issue of MUUG Lines, the motion to adopt Draft #6 (Sep. 28, 1992) of the Manitoba UNIX User Group by-laws was tabled. The motion was carried by a show of hands. Official copies of the by-laws are available from the MUUG Secretary.

2. Election of the MUUG Board of Directors
The candidates for the eight positions on the MUUG Board were given a few moments to introduce themselves. The MUUG members present at the meeting then cast their ballots. Ballots were counted by three volunteers. The candidates who were elected are (in alphabetical order):

Gilbert Detillieux
Gilles Detillieux
Bary Finch
Paul Hope
Rick Horocholyn
Richard Kwiatkowski
Roland Schneider
Susan Zuk.

3. MUUG Fees
Membership fees for 1992-93 are $25 per person. MUUG Online fees (discounted for active participation in the group) are $50 per account.

Agenda

for
Tuesday, November 10, 1992, 7:30 PM
St-Boniface Hospital Research Centre
Theatre, Main Floor, 351 Taché

1. Round Table 7:30

2. Business Meeting 8:00
   a) President's Welcome
   b) MUUG Board Introductions
   c) Objectives for the Year
   d) New Business

3. Break 8:20

5. Presented Topic 8:30
   FrameMaker for UNIX
   By Jim Baglot, Frame Technology
   A look at FrameMaker on a UNIX workstation. Jim will describe some of the features of FrameMaker as well as a new product called Frame Builder. The demonstration will be on a live system, with the display connected (hopefully) to a large projection TV unit.

6. Adjourn 9:30

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

Coming Up

Meeting:
Our next meeting is scheduled for Tuesday, December 8, at 7:30 PM. Meeting location will be given in December's newsletter. The meeting will be the traditional December wine and cheese party. To give the evening a computer-related theme, while still keeping things festive, we are considering having a computer demo area set up with interesting/fun applications, such as computer music, graphics, and so on. If you have any suggestions for things you'd like to see displayed (or would like to display), let us know ASAP. Thanks.

Newsletter:
There are no more articles in the pipeline. I could use some more material, especially shorter articles – half a page to one page in length would be fine. I would like to see some more book reviews – have you read any good books lately? How about some handy tips for MONA users, or any UNIX users, for that matter? Monsieur Ex has also let me know that his mail-box is empty lately – please submit your questions to the old guy via e-mail to <m-ex@muug.mb.ca> or by FAX to the editor.