Volume 5, Number 9 September 1993 $2.50

MUUG Lines

Manitoba UNIX® User Group

Newsletter of the Manitoba UNIX® User Group

The Internet:

Communicating with the World

The Internet links millions of computers worldwide and is the backbone of modern computer communications. It is used to transfer electronic mail and news, distribute software, and access remote computer systems. It extends the local area network to the world.

The Manitoba UNIX User Group invites you to a special presentation on the Internet by Dr. Roger Taylor, of the National Science Foundation, in Washington, D.C. He is a founder of CA*net, the Canadian research network which is part of the Internet, and he is presently coordinating the future direction of internetworking in the United States and internationally. Dr. Taylor will explain what the Internet is, and why it has become a crucial resource around the world in business, government, and education.

Dr. Taylor's talk will be followed by a brief presentation by Mr. Bill Reid, of the University of Manitoba's Computer Services Department, describing the Internet connection services available locally through Manitoba's MBnet.

About Our Featured Speaker

After graduating from McMaster University in 1965, Dr. Taylor worked full-time for 21 years on theoretical physics research at the National Research Council of Canada (NRC), establishing a strong international reputation in condensed matter physics. His research focused mainly on first principles, calculations of the properties of the electron, gas, and of simple metals and alloys. In 1986 he was appointed Director General of the Division of Informatics, a position that gave him the opportunity to set new computing directions within NRC and to meet new and interesting challenges. During his four years in that position, he worked with members of the Division to broaden its activities from being almost exclusively a centralized mainframe shop to being able to offer services on a wide variety of different platforms. Additionally, they set up an NRC-wide network and, more significantly, two of them spear-headed the establishment of the national high-speed network, CA*net, which commenced operations in 1990. This latter activity required considerable entrepreneurship on Dr. Taylor's part, meeting with the networking community to agree on a vision and then selling that vision to obtain government funding when money was short and when networking was an unfamiliar concept at senior levels.

Dr. Taylor left the NRC at the end of 1990, and for the next year and a half pursued a consulting career both with Fairfax Materials Research, investigating the mechanical properties of metal-void composites, and also with DA Ford & Associates in Ottawa, providing consultation to the Canadian Government Telecommunications Agency regarding the CANARIE high speed network proposal, and management advice to CA*net.

In 1992, he took up his current position, Executive Officer for the Networking Division of the National Science Foundation (NSF), where he is very much involved with NSFNET and the NREN program. His position requires him to perform executive functions as well as program direction, and allows him to continue research in physics. In addition to scientists, his activities have involved him with inter-agency working groups and a wide variety of network uses, e.g. electronic publishing and digital libraries where the humanities are particularly active.

This Month's Meeting

Meeting Location:
Our next meeting is scheduled for Tuesday, September 14, 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 Ramblings 2
President's Corner 3
Elections: Nominations for the Board 3
Hands-on: Profile That Code! 4
Hands-on: MONA Time Quota 5
Feedback: Ask Monsieur Ex 6
SIG Sideline 7
Sept. 14th Meeting Agenda 7
MUUG/CIPS Fall Symposium 8

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

Winding Down...

By Gilbert Detillieux

Well, here we are back again after a very short break during a very short (and very wet) summer. And now, those of us involved in the board prepare for another kind of "flood" as things start up again for the fall. As you may know, and as you'll see later in the newsletter, October is when MUUG holds its annual elections for the board. Nominations are now open, and the elections committee has put together an initial list of nominees.

As you look at the list, you'll notice some new names, and also the absence of some names that have been on the list for quite some time. Stepping down as president is Susan Zuk, a founding member, who will still have a role on the board as past president. Also stepping down are founding members Gilles Detillieux, and myself.

In my case, I felt that I had been quite heavily involved in the group's operation for long enough. I need a bit of a break, and I think it's time for some new blood to get involved on the board. Certainly, the group has grown enough, and has a number of very keen new members, that the group will continue to do well without me.

But don't worry — you haven't seen the last of me yet! I do still plan to remain active as a member, attending meetings, helping out with special events, and acting as a system administrator on the MUUG Online system. You'll likely still run across my name in future issues of the newsletter as well.

Which brings me to an important point — I will be stepping down as newsletter editor after the October issue, which means the new board will have to fill the position in time for the November issue. (I am prepared to offer the new editor lots of assistance with getting that first couple issues out the door.)

Also, as we are now winding down the fifth year of newsletter issues, I would like to have a comprehensive article index in the next issue. I have all the back issues; do I have a volunteer to go through these and compile a list?

Finally, I would like to say that editing the newsletter for the past two years has been a great experience. Sure, it's meant a few hectic rushes to meet deadlines, but it has also been quite enjoyable. Anyone up for the job?

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

Welcome Back!

By Susan Zuk, President

I hope everyone had a restful summer and you are all ready for a new MUUG season. We have a really exciting year planned, beginning with our September 14th meeting. Dr. Roger Taylor is paying us a visit to tell us about the Internet. Dr. Taylor is traveling all the way from Washington D.C. and the National Science Foundation. This will be a wonderful start to the 93-94 schedule of events. From all indications, the Internet is a hot topic and we are expecting a full house. Come early to ensure yourself a good seat.

During the summer, the board members did not stop to take a break. We did take a break from the city to head to the cottage for the annual July weekend planning session. It was called "really roughing it" when we surprisingly received beautiful weather and had to hold the meeting outside with the sun shining upon us. We even escaped the big mosquito onslaught until night. The weekend was a great way for us to brainstorm and prepare for the year as well as relax and socialize. The plans resulting from the meeting will be presented at up coming meetings and newsletters.

We have started preparing for the MUUG elections. October is the official start for MUUG's new year. Included in this newsletter are nominations for the 93-94 board. If you are interested in being a part of the board or have questions about the duties/positions please contact either myself at 631-2530, Gilles Detillieux at 789-3766, or Gilbert Detillieux at 474-8161. Elections will be held on Tuesday, October 12th. See the election article for more details.

Another exciting event is coming up on Wednesday, October 27th. This event is called "Survival: Riding the Technology Wave" and MUUG has teamed with CIPS (Canadian Information Processing Society) again to offer this seminar to you. Plans began in May and we have a great range of speakers and topics discussing Today's Management Issues, Application Development Environments and Tools, as well as New Technologies. To accompany the speaker sessions, we have planned an exhibit area where attendees can see current hardware and software applications. Keep watching your mailbox for the brochure describing more on the seminar and for registration information.

That's all for now, there'll be more news at the September 14th meeting.

I'm looking forward to seeing you there.

ELECTIONS

Nominations for MUUG Board Elections

By Gilbert Detillieux

Well, October and the MUUG Annual Meeting are fast approaching. One of the most important events at the annual meeting is the election of a new board of directors. The choice of directors is very important, because they plan group meetings, make policies, raise and spend group money, and generally set the direction for the group as a whole. Almost every event, project, or initiative, from the monthly newsletter to the fall symposium originates with the board of directors. All this can be a lot of work, so it's important to elect people with the energy, enthusiasm, and commitment to do a good job.

Note that, in keeping with the new constitution, nominees for the board are not voted into specific positions, they are just elected to the board. The board then divides up the positions of president, vice president, secretary, treasurer, newsletter editor, membership secretary, publicity director, and meeting coordinator. This gives the board the flexibility to reassign duties if someone can't continue to do their job part way through the year.

The election committee does two things to help facilitate the process: 1) it nominates people for election to the board; and 2) it runs the election and counts the ballots. The list of people nominated by the committee is just a starting point – any voting member of the group can be nominated simply by getting the support of one other member. If you feel you would like to contribute to the group by running for a board position, please don't hesitate to do so. Below is the current list of nominees, followed by instructions on nominating others.

Ramon Ayre	Partner	CSB Systems
Barry Finch	Systems Engineering Rep.	IBM
Paul Hope	Coordinator of A/V and Info. Services	St-Boniface Research Centre
Rick Horocholyn	Sr. Financial Consultant	Manitoba Hydro
Kirk Marat	NMR Supervisor	Chemistry, UofM
Rory McLeod	Integrated Solutions Consultant	Xerox
Greg Moeller		MTS
Roland Schneider	Computer Consultant	Niche Technology
Brad West		Freshwater Fish Marketing Corp.

A total of eight people will be elected. If you want to be nominated, or nominate someone else, send a letter to the group's mail-box or deliver it in person to a member of the election committee. The letter must contain the name, title, and employer of the nominee, along with a short (100 word) biography, and must contain the signatures of the nominee and one other member. The letter must be received no later than September 28, 1993. This is 14 days prior to the annual meeting. If you have any questions about the election, please give me, Gilbert Detillieux, a call at 474-8161 during business hours.

MUUG Election Committee:
Susan Zuk, Chairperson
Gilbert Detillieux
Gilles Detillieux

Performance Problems? ... Profile That Code!

By Gilbert Detillieux, The University of Manitoba

So, you've just finished writing that big C program, and after a fair bit of testing and debugging, it all seems to work as it should. Problem is, it runs so slowly you'll either have to get a faster machine, or get used to long coffee breaks! Sound familiar? Well, don't despair — you may not have to junk the old hardware just yet, and your caffeine-fried nerves can use a break.

There's probably some simple changes you can make to optimise the performance, but it can be hard to know what part of the code to attack. You certainly don't want to start rewriting and optimising everything —that's more trouble than it's worth, and if the code isn't used a lot, it can be a waste of effort. Before you start optimising, you need to know where the bottlenecks in the code are.

One of the rules of thumb in programming is that a program will spend at least 90% of its time in about 10% of the code. Now, this isn't very scientific, but it's surprising to find how often it is true. Once we know what 10% of the code we need to work on, the job is much easier. That's where profiling comes in.

Execution time profilers work by periodically checking what part of the code is being executed, and updating counters based on that. Typically, they just check the program counter's address on every "tick" of a timer, and increment a counter that corresponds to that section of memory. If you have enough counters, and let the program run for enough ticks of the clock, you end up with a fairly representative sample of how much time was spent in each area of memory. By matching this up against the program's symbol table, you can find out what functions correspond to those memory areas.

Another common profiling technique is to compile or link in special code that will count all the calls to any function. This count may not directly correlate to execution time, but it can still be a very useful piece of information.

Profiling on UNIX uses a combination of the two techniques above. You specify an option when you compile and link your program, such as this:

```
cc -p program.c
```

This will compile in extra code, as well as link with a special version of the C library and run-time start-up code, to enable profiling. (Unfortunately, we don't have the required version of the library on `mona`, so the profiling option won't work there. However, the C compilers on most other UNIX systems should provide support for profiling.)

You can then run your program as usual. The program might be slightly slower due to the profiling, but it shouldn't make a big difference. At the end of execution, the profiling data are written out to a binary file (typically called "`mon.out`" in the current directory). This file can then be analysed with the "`prof`" command, to get a readable report. The default format of the report will look something like this:

```
 %time  cumsecs  #call  ms/call  name
  50.1   22.00 18640953     0.00  _strncmp
  30.6   35.43                   mcount
  16.6   42.71      1  7280.00  _wtmp
   2.5   43.80     72    15.14  _read
   0.2   43.90  36803     0.00  _want
   0.0   43.91     18     0.56  __doprnt
   0.0   43.92     71     0.14  _lseek
   0.0   43.93    523     0.00  _malloc
   0.0   43.93     18     0.00  _printf
   0.0   43.93    178     0.00  _strlen
   0.0   43.93      1     0.00  _time
   0.0   43.93     18     0.00  _write
```

(Of course, it will typically contain many more lines of output than this, but you get the idea.)

The above is actually part of the output I got by profiling the "`last`" command that comes with the current versions of BSD. I had planned on adding some options to the program, so I fetched the source and compiled it. To my surprise, it ran about 17 times slower that the SunOS "`last`" command. After profiling, the reason became more clear — on a test "`wtmp`" file of about 72,000 records, the `strncmp`function was called 18.6 million times! This accounted for a significant part of the execution time. (The "`mcount`" function in the report is part of the profiling code, which counts function calls. Due to the unusually large number of calls to `strncmp`, "`mcount`" itself actually chalked up a significant amount of time as well.)

OK, now that we know where the program is spending its time, we still have to figure out why, and whether or not we can reduce this. Other debugging techniques can help there, by letting you see from where the functions are being called, and with what arguments. Sometimes, a few well-placed trace prints will really help make things clear.

In the case of the "`last`" command, it turns out that most of the calls are due to a search routine which looks up a *tty* name in a list, so that it can find out the logout time on that *tty* or update the entry. By using a few simple techniques to improve the search, such as deleting entries that are no longer needed, reordering the list, and comparing the first character of a string before calling `strncmp`, I was able to greatly improve the performance of the program:

```
 %time  cumsecs  #call  ms/call  name
  33.1    0.44     72     6.11  _read
  27.1    0.80      1   360.00  _wtmp
  18.8    1.05 154453     0.00  _strncmp
  10.5    1.19  36803     0.00  _want
   8.3    1.30                   mcount
   1.5    1.32  36803     0.00  _chkwant
   0.8    1.33     18     0.56  _write
   0.0    1.33     18     0.00  __doprnt
   0.0    1.33    504     0.00  _free
   0.0    1.33     71     0.00  _lseek
   0.0    1.33    533     0.00  _malloc
   0.0    1.33     18     0.00  _printf
   0.0    1.33    178     0.00  _strlen
   0.0    1.33      1     0.00  _time
```

With these optimisations, the program is now about 30 times faster than before, and better than twice as fast as the SunOS "`last`" command. (Incidentally, the SunOS version is based on the source from 4.2 BSD, which has a bug in the *tty* search list. This bug is what resulted in the apparently better performance, although its output could be incorrect.)

Note that the extra calls to `malloc`and `free`, due to the deletion of unneeded entries from the search list, resulted in almost no performance cost, and permitted a very significant drop in the number of `strncmp`calls. Also note that the `strncmp`function is still called more than any other function in the program. However, its total execution time is now less than that of a few less frequently called system calls. At this point, further optimisations would result in only negligible performance gains. So we decide to call it quits, satisfied with our 30-fold speed-up. (Of course, your mileage may vary.)

Profiling is a simple, and often very effective, way to find bottlenecks in a slow program. Sometimes, it can even help you find bugs in the code. It's a shame that it seems to have been forgotten by many of the developers of modern compilers and whiz-bang graphical debuggers. If you write serious, production code, making the "lost art" of profiling part of your debugging routine will help you to produce better code.

The MUUG Online System and Your Connect Time Quota

By Gilles Detillieux

In our last issue, I discussed the disk space quota system on the MUUG Online system. Along the same lines, I will now discuss the newly-implemented connect time quota system. The time quotas have been the subject of several questions by our users, and have even caused some alarm among a few. This article will hopefully answer most of your questions, and put your mind at ease with regards to time quotas.

Time quotas are a very recent addition to the MUUG Online system. They were put in place August 13, initially only to inform users of their connect time. Full enforcement of time quotas will begin in September, about the time you receive this newsletter.

We've implemented time quotas as a solution to a growing problem. Our user base has expanded considerably since our system was put in place, leading to much greater usage of the University of Manitoba's modem pool. Excessively high connect times by some users has aggravated the problem. From the start, we recommended that users restrict themselves to an average of about an hour a day, and most users have been very reasonable about it, often remaining well below the recommended time. A few users have chosen to ignore our guidelines however. This hasn't been a serious problem during the summer, but come September the University's staff and students will need access to the modem pool in much greater numbers.

The way the time quota system works is quite simple. Every morning, the system tallies the connection times for active users, based on their login and logout times for the last thirty days. The login and logout times are recorded in a system log file. Totals are tallied only once a day because the process takes a little too long to do more often than that. Since we can't actually detect whether the user is tying up a modem or not, we count all UMnet logins, virtually all of which do indeed originate from the modem pool. Users logging in on campus will usually do so via some other network node, and this will not be counted against their quota.

When you login to the system, it checks your total and shows it to you. Because the totals are not continually updated during the course of the day, what you are shown is your total connect time for the previous thirty days, which does not include the current day's time. If your total is above the recommended 30 hours per 30-day period, you will also get a warning about this, and we ask you to limit your usage for a while to get your total down. If your total is above the maximum limit (currently 60 hours), the system will not allow you to login until the total goes down below that limit.

We feel that the current quota of 60 hours per month is more than enough, and will not interfere with users who have been reasonable. It remains to be seen whether it will decrease usage of the modems enough so that our users don't conflict with the University's. If there's a lot of contention for UMnet modems this fall, and we see lots of users on at the same time, we might have to impose stricter limits. Remember that the 60 hour limit is the maximum we allow — we still recommend that you limit yourself to no more than 30 hours per month, or one hour per day on average.

We chose to tally up login times over a 30-day period, rather than by day or by week, to give users more flexibility. Shorter time periods would have meant smaller quotas. With a long time period, and a correspondingly large quota, users can choose whether to spread their usage evenly over time, or spend it all in a few days if they're prepared to go without the system for the rest of the month. With a large user base, things should more-or-less average out over time.

For the first few weeks of operation, the time quota was not strictly enforced. Users above the limit would get a warning, but were still allowed to login to the system. This served to give them advanced notice of the quota enforcement, and to give them time to change their usage patterns without being denied access altogether.

Time quotas will not interfere with UUCP connections to our system. As well, we will not "charge" UUCP connect time to the account owner's own login ID. If this becomes a problem in the future, we will try to find a way to deal with it, such as limiting UUCP traffic at peak times. Even our busiest UUCP host, which forwards mail and news to a half-dozen other sites, is not using the modems excessively during prime evening hours.

For the time being, the system implements one quota for everyone. If anyone can make a convincing case to us that they will need more than 60 hours of connect time in any given 30 day period, we can look into implementing a more flexible quota system, where we can increase a user's quota as required. (If a user's work benefits the group in some tangible way, that helps his/her case, just as for disk quotas.) We can also look into making alternate arrangements for these users, such as other file transfer methods or other network connection options, as appropriate.

If you require any such arrangements, or have any further questions or concerns about the time quota policy (or any other policies, for that matter), you should address your requests to the Computer Use Committee, <cuc@muug.mb.ca>.

We will do our best to accommodate you.

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 Monsieur Ex,

I have often wanted to do something that is easy in DOS, but not in UNIX:

```
ren *.txt *.prn
```

This renames all files with the `.txt` extension to the same name, but having a `.prn` extension. How would I write a script that `mv`'s or `cp`'s (etc.) a group of files *en masse*?
Rick Horocholyn

Dear Rick,

Ah oui! The old wildcard matching problem. The differences in the way UNIX and DOS handle windcards causes *beaucoup de confusion* when switching between these systems. The REN command in DOS can do what it does due to two things about filenames that are different than in UNIX: 1) each command gets the arguments with the wildcard characters intact, and it's up to it to decide how to interpret them (whereas on UNIX, the shell expands wildcard arguments into the corresponding list of file names before calling the command), and 2) file names in DOS have a fixed internal length – always 11 characters, and the "." is just used to seperate the first 8 from the last 3 (whereas in UNIX, file names are of a variable length, and the "." is just another character).

The first point is not the real obstacle, though, since the arguments could be quoted to prevent wildcard expansion. The second point is perhaps not as obvious, but it is important to the renaming process, since it allows REN to easily map characters from the first pattern to the corresponding characters in the second, since the character positions can easily be determined. But if you think about how a pattern such as "`*.txt`" (or "`abc*.txt`", or "`*abc*.txt`") would map on UNIX, you realize that there is no way of knowing where the corresponding replacement characters in a pattern such as "`*.prn`" would fall. (In DOS it's easy, since the file extension always starts at a fixed position.)

Fortunately, UNIX does provide some nice features of its own for dealing with file names. In the C shell, a loop such as this could be used:

```
foreach file (*.txt)
  mv $file `basename $file .txt`.prn
end
```

(A similar construct can be set up for the Bourne shell too.)

If you understand how "`for`" loops work, the only tricky part here is how we generate the target file name. The "`basename`" command is used in this case not to strip off a leading directory name, but to strip off the "`.txt`" suffix. We then add the "`.prn`" suffix to that result to get the target name. More complicated renaming could be accomplished by using something like "`sed`" to substitute a new file name for the old one.

As an exercise to the more masochistic readers, consider how you might rename files of the form "`99-xxx.yyy`" to the form "`yyy99.xxx`" (where `99` represents any two digits, and `xxx` and `yyy` are any three letters) in both UNIX and DOS. (And using the MKS Toolkit for DOS is cheating!)
Salut,
M. Ex.

***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.*

The Fortune File

BABY(1) **USER COMMANDS** **BABY(1)**

NAME

baby — create new process from two parent precesses

SYNOPSIS

baby [**–***sex*] [**–***name*]

DESCRIPTION

Baby is initiated when one parent process polls another server process through a socket connection (BSD) or through pipes in the System V implementation. *Baby* runs at a low priority for approximately 40 weeks, then terminates with heavy system load. Most systems require constant monitoring when *baby* reaches its final stages of execution.

Older implementations of *baby* required that the initiating process not be present at the time of completion. In these versions, the initiating process is awakened and notified of the results upon completion. Modern versions allow both parent processes to be active during the final stages of *baby*.

OPTIONS

–*sex*	Option indicating type of process created.
–*name*	Process ID to be attached to the new process.

RESULT

Successful execution of the *baby* results in a new process being created and named. Parent processes then typically broadcast messages to all other processes informing them of their new status in the system.

BUGS

The *sleep*(1) command may not work on either parent proceses for some time afterward, as new *baby* processes contantly send interrupts which must be handled by one or more parents.

Baby processes, upon being created, may frequently dump in **/tmp**, requiring **/tmp** to be cleaned out frequently by one of the parent processes.

SEE ALSO

sleep(1), dump(8), cry(3)

SIG Sideline

By Bary Finch, SIG Coordinator

Welcome back after the summer break.

This column will be used to share information on the Special Interest Groups (SIGs) that have been formed within the MUUG membership.

Our first SIG, which combines interests in LINUX, as well as System Administration, has continued in popularity. We held meetings on the Tuesdays of June 15 and July 20. These meetings continued in the roundtable discussion format.

There was a large turnout to the June meeting, with a smaller assembly at the July meeting. As a result, we cancelled the August meeting to let everyone enjoy summer (or whatever it was that we had during July and August). Our trend in the SIG is likely to follow the MUUG meeting schedule, and skip July and August. This will be reviewed again next summer.

The discussions in the roundtable sessions covered a wide spectrum of topics, from large SCSI disks to favorite shells to use. Any and all discussions on Linux and System Administration are welcome, so quite a variety of topics come up.

The next meeting is scheduled for Tuesday, September 21, at 7:30 p.m. This meeting will again be held at ISM, 400 Ellice Avenue, behind Portage Place. Our host is Wolfgang von Thuelen. He will be waiting in the lobby as of 7:15 p.m. to let everyone in.

I am still in the process of trying to locate a PC with Linux that can be transported to the meeting. I am also still trying to get a presenter for a topic such as uucp for this meeting. So for now I will have to leave the agenda as TBA. We will of course still have the roundtable discussions on Linux and System Administration.

If anyone has an interest in bringing a PC to any of the SIG meetings, or if you'd rather present a topic at one of the SIG meetings, please let me know. You can email me at bfinch@muug, or call me at work: 934-2723.

Hope to see you at the September meeting!

Agenda

for

Tuesday, September 14, 1993, 7:30 PM
Samual N. Cohen Auditorium
St-Boniface Hospital Research Centre
Main Floor, 351 Taché

1. President's Welcome 7:30
   a) Old Business
   b) New Business
   c) MUUG Board Nominations

2. Presented Topic 7:45
   The Internet: Communicating with the World
   By Dr. Roger Taylor
   See cover story for details.

3. MBnet Connection Options 9:00
   By Mr. Bill Reid

4. MUUG Online 9:15
   By Gilbert Detillieux

6. Coffee and Informal Discussion 9:30

Note: Please try to arrive at the meeting by 7:15, since we are expecting a large turnout.

Coming Up

Meeting:
Next month's meeting is scheduled for Tuesday, October 12, at 7:30 PM. Meeting location will be the St-Boniface Research Centre, as usual. The October meeting topic is still to be announced. This is also the annual MUUG meeting, where the elections are held.

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:
Once again, I'm just about out of articles in the pipeline, so 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 has room for more of your wonderful queries 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.

CIPS/MUUG ANNUAL FALL SEMINAR

INTERNATIONAL INN

WEDNESDAY, OCTOBER 27, 7:30AM-5:00PM

Keynote speaker

Roger Burlton is the founder of the Process Renewal Consulting Group, an industry leader, and an internationally renowned speaker.
Mr. Burlton will discuss the dramatic changes occurring in information technology, in methods, and in management. He will advise on how to cope with organizational challenges while transforming to improved business performance.

Three seminar tracks

Today's Technologies

This track will explore modern computer communications, high availability systems, and security, capped with a lively panel discussion on the merits and pitfalls of today's hottest operating systems.

Application Development

In addition to a panel session with representatives from today's major database vendors, this track features presentations on rapid application development (RAD), object oriented systems, and graphical user interfaces (GUI's).

Enterprise and Management Issues

The management track will begin with a case study of a local success: UGG's transition to client-server computing. This will be followed by presentations which explore the business and management sides of down-sizing.

Vendor Exhibits

The focus this year is on software. Major vendors of database systems, communications software, interface builder tools, and operating systems will be on hand to demonstrate their wares and answer your questions.

Canadian Information Processing Society

Manitoba UNIX User Group

Prices

(includes lunch)

CIPS Members: $200 MUUG: $250
Students: $30 Others: $295
(late registration – add $50 after Oct. 8)

For further information

Don Gillespie	(204) 983-6725
Susan Zuk	(204) 631-2530
FAX	(204) 694-3901