Volume 6, Number 5 March 1994 $2.50

# MUUG Lines

Manitoba UNIX® User Group

Newsletter of the Manitoba UNIX® User Group

## The Power Switch

*By Andrew Trauzzi*

The year — 1980. The players — Zilog and Intel. Zilog is in the lead with the Z80, and Intel has just fumbled on its own 8085. 3rd down. Zilog snaps the Z8000. Intel counters with a massive 8086 advertising blitz. Zilog fades back, fades, fades, fades...they're gone! Intel becomes the undisputed chip champion, unbeaten for 13 years, but 13 seems to be Intel's unlucky number. This year, the ageing Intel will have to go head-to -head against the young and mighty Motorola PowerPC. Who will be the victor? Will it be a tie? A draw? Let the games begin!

At this month's meeting, Bruce Danforth of IBM Canada will be presenting the PowerPC. As you probably heard, IBM released the first PowerPC-based computer back in September in the form of an RS/6000. This year, both IBM and Apple have announced computers based on the PowerPC. Unfortunately, most of the hype around the release of the PowerPC deals with a certain popular OS. (See my Ramblings on page 2 for more.) The PowerPC is heralded as the next generation of computing power. Many, many journalists have written thousands of words on how the PowerPC will affect DOS and Mac users. But how will the PowerPC impact UNIX users?

The first thing that came to my mind was the big CISC vs. RISC war of the late 1980s. UNIX vendors saw the power in RISC technology and went with it. PC users, on the other hand, called RISC "a fad." The PowerPC is definitely a RISC chip (I don't care what Intel says, the 80x86 family is not RISC!), and its proliferation and power should put an end to the RISC vs. CISC war. The clear winners are the UNIX vendors and their users.

The PowerPC offers (or will offer) users a large choice of operating systems — a large percentage grouped under the PowerOpen Environment. PowerOpen was formed in 1991 by IBM, Apple, and Motorola, and placed in the hands of the PowerOpen Association. The POA was founded by 7 large corporations who, along with future recruits, will guide PowerOpen in the future. You can't buy PowerOpen, but rather, it's a term for compliant operating systems sold by the members of the association. It just so happens that many of the founding members are UNIX vendors, so the resultant PowerOpen compliant operating systems will be primarily UNIX-based.

Currently, IBM's PowerPC runs a derivative of AIX as its base OS. Now, or in the near future, the PowerPC will support: System V, BSD 4.3, Posix, TCP/IP, X/Open, X-Windows, OFS's DCE and DME, WABI, Solaris, OS/2, Macintosh, DOS, Windows NT, and its own base OS — Pink. Do you see a common UNIX thread there? If all that wasn't enough, you have the ability to run multiple virtual machines on the PowerPC, and cut and paste data between them! Maybe UNIX and DOS will become friends there!?

UNIX is an easy and natural extension of RISC technology, and with projected early sales in the millions, I think that millions will be introduced to the power of PowerPC through UNIX. I hope that Motorola really delivers a knock-out punch to the proprietary and overrated Intel x86 chipset.

### This Month's Meeting

**Meeting Location:**
Our next meeting is scheduled for Tuesday, March 8, 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

Printing Courtesy of Xerox Canada Ltd.

## RAMBLINGS

# The Cray PDA

***By Andrew Trauzzi***

**Los Angeles** (AP) — Today, Cray Computers, the supercomputer giant, released the preliminary specifications for its new hand-held personal digital assistant. Dubbed the Y-ME, this little wonder screams along at 1 bip, contains 64 MB of dual-port high speed RAM, a whopping 5GB of disk storage, photorealistic 24 bit colour, with equally impressive 16 bit neural surround sound, builtin satellite uplink to all major networks, a digital cellular fax and phone, runs for 24 hours on 2 AA batteries, and costs only $500.00!

We rushed to interview Al Kroans of the Gronko-Widget corporation of America — the largest buyer of computers on the planet.

AT — “So Mr. Kroans, what do you think about the Cray announcement?
AK — “It sounds great! Can it run Windows?”
AT — “Groaaaannnnnn!!!!”

Unfortunately, half of North America is suffering from this Windows affliction. The only good cure is a strong dose of Motorola! Now where is my copy of WABI?

### Power to the People

Did you ever wonder what Power stands for? By this, I mean the Power in PowerPC. Did you even know it was an acronym? You should know by now that even the most innocuous computer terms are acronyms! Power stands for: “Performance Optimized With Enhanced RISC.” I am not making this up. (I’m saving that for the April issue!)

### Polymorphic Protocols

Boy does that sound good! In January’s Dr. Dobbs Journal, Bill Jolitz (the creator of 386BSD), examines a particularly clever way of addressing (groan!) the TCP/IP addressing dilemma. I found this to be an excellent and humorous article, that is well-written and well-referenced. If you can’t find a copy, send me an e-mail, and I will lend you mine.

### Book Reviews

Many of you have been asking me for information on UNIX books. It just so happens that I stumbled across a comprehensive list of UNIX books — from beginner to guru. For the next few months, I will be printing sections of the list. If you want to order any of the books, just give the ISBN number to the order clerk at any bookstore. ✒

## The 1993-1994 Executive

| | | |
|---|---|---|
| President: | Bary Finch | (W) 934-2723 |
| Vice-President: | Ramon Ayre | (W) 947-2669 |
| Treasurer: | Rick Horocholyn | (W) 474-4533 |
| Secretary: | Brad West | (W) 983-0336 |
| Membership Sec.: | Greg Moeller | (H) 786-6132 |
| Mailing List: | Roland Schneider | 1-482-5173 |
| Meeting Coordinator: | Roland Schneider | 1-482-5173 |
| Newsletter editor: | Andrew Trauzzi | (W) 986-3898 |
| Publicity Director | Rory Macleod | (W) 488-5168 |
| Past President | Susan Zuk | (W) 989-3530 |
| Information: | Bary Finch | (W) 934-2723<br>(FAX) 934-2620 |
| (or) | Andrew Trauzzi | (W) 986-3898<br>(FAX) 986-5966 |

## 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.

## Advertising Rates

| | |
|---|---|
| Quarter page | $50 |
| Half page | $75 |
| Full page | $100 |
| Insert (1-4 pages) | $100 |

Above prices are per issue. The first ad is charged at the full price; each successive month is 1/2 price.

Ad copy must be submitted by the final copy deadline for an issue (usually 3 weeks prior to the monthly meeting) in a format acceptable to the editor. (Please make arrangements with editor beforehand.)

**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 at any meeting, or by mail.

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

**Internet E-mail: membership@muug.mb.ca**

## PRESIDENT’S CORNER

# Power for Everyone

*By Bary Finch*

This month we are returning to the format of having a presentation from one of the industry providers. In this case it will be IBM presenting on the PowerPC.

This new chip technology, which is the focus of our cover story, is one of the most sensationalized items in the computer trade press these days. Especially with Apple’s recent introduction of its PowerPC models.

Our presenter this month is Bruce Danforth of IBM Canada. Bruce has been with IBM Canada for 24 years. He is currently the Team Leader, Technical Consulting Group, RISC System/6000 Division. The Technical Consulting Group’s main purpose is to transfer skills to IBM, Business Partners, and customers who work with AIX products. The Group also ensures technical professionals are skilled and focused on those areas of most importance to IBM’s customers.

Bruce will be talking about the phenomenon that is the PowerPC. This includes the PowerPC alliance between IBM, Apple and Motorola, and the developed chips, such as the 601 that is in the marketplace today. Bruce will also position where the RISC marketplace is from IBM’s view, and IBM’s trends, and directions for the PowerPC technology.

### Coffee Ammunition

This presentation should enlighten you on the current status of the PowerPC technology, and where it’s headed. You’ll definitely get more ammunition for those coffee discussions.

I’d like to take this opportunity to thank all the Corporate Sponsors for attending our last meeting, and receiving their MUUG Corporate Sponsor plaques that signify their contribution. There were representatives from (in alphabetical order) Digital Equipment of Canada Ltd., Great-West Life Assurance Company, Hewlett-Packard (Canada) Ltd., IBM Canada Ltd., Online Business Systems, Tandem Computers of Canada Ltd., the University of Manitoba, and Xerox Canada Ltd.

### Laser-Etched Plaques!

The presenters of our Corporate Sponsorship plaques were Susan Zuk, Past President, and Ramon Ayre, Vice-President. This was our first presentation of plaques. These plaques are a laser etched walnut plaque, with room for five plates showing the years of the Corporate Sponsorship. We are also actively pursuing more Corporate Sponsors, and will have continuing presentations of their plaques as more of them commit to supporting us.

MUUG could not exist as it is without the continued support of our Corporate Sponsors. We thank them all for their recognition of the importance of the UNIX user community through their donations to MUUG.

Our last meeting also included our second Internet presentation. This time we had Smoot Carl-Mitchell of Texas Internet Consulting talking about why people use the Internet. We had a great attendance, and thankfully within the seating capacity of the St. Boniface Research Center.

I’d appreciate hearing your comments on Smoot Carl-Mitchell’s presentation. MUUG is trying to bring in the kind of speakers that YOU want to see. The only way we will know if we are being successful at doing that is to hear from you. So let us know! I can usually be reached by phone at 934-2723 or by email at `<bfinch@muug>`. See you at the March meeting!

## CORPORATE SPONSORS

# The Manitoba UNIX User Group

**gratefully acknowledges the generous support of the following**

# Corporate Sponsors

IBM

digital™

HEWLETT PACKARD

UM

XEROX The Document Company

Great-West Life Assurance Company

ONLINE BUSINESS SYSTEMS

TANDEM COMPUTERS

# C++ Q&A

## *By Marshall P. Cline*

*This month we look at inline functions, constructors and destructors, and operator overloading.*

**Question 14: What are inline functions? What are their advantages? How are they declared?**

An inline function is a function which gets textually inserted by the compiler, much like a macro. Like macros, performance is improved by avoiding the overhead of the call itself, and (especially!) by the compiler being able to optimize *through* the call (‘procedural integration’). Unlike macros, arguments to inline fns are always evaluated exactly once, so the ‘call’ is semantically like a regular function call only faster. Also unlike macros, argument types are checked and necessary conversions are performed correctly.

Beware that overuse of inline functions can cause code bloat, which can in turn have a negative performance impact in paging environments.

They are declared by using the ‘inline’ keyword when the function is defined:

```
inline void f(int i, char c) { /*...*/ }
```

or by including the function definition itself within a class:

```
class X {
public:
  void f(int i, char c) {/*...*/ }
};
```

or by defining the member function as ‘inline’ outside the class:

```
class X {
public:
  void f(int i, char c);
};
//...
inline void X::f(int i, char c) {/*...*/}
```

Generally speaking, a function cannot be defined as ‘inline’ after it has been called. Inline functions should be defined in a header file, with ‘outlined’ functions appearing in a ‘.C’ file (or .cpp, etc.)

**Question 15: What is a constructor? Why would I ever use one?**

Objects should establish and maintain their own internal coherence. The ‘maintaining’ part is done by ensuring that self-consistency is restored after any operation completes (ex: by incrementing the link count after adding a new link to a linked list). The part about ‘establishing coherence’ is the job of a constructor.

Constructors are like ‘init functions’; they build a valid object. The constructor turns a pile of incoherent arbitrary bits into a living object. Minimally, it initializes any internally used fields that are needed, but it may also allocate resources (memory, files, semaphores, sockets, ...).

A constructor is like a ‘factory’: it builds objects from dust. ‘ctor’ is a typical abbreviation for constructor.

**Question 16: How can I make a constructor call another constructor as a primitive?**

You can’t. Use an ‘init()’ member function instead (often ‘private:’).

**Question 17: What are destructors really for? Why would I ever use them?**

Destructors are used to release any resources allocated by the object’s constructor. Ex: a Lock class might lock a semaphore, and the destructor will release that semaphore. The usual ‘resource’ being acquired in a constructor (and subsequently released in a destructor) is dynamically allocated memory.

**Question 18: What is operator overloading?**

Operator overloading allows the basic C/C++ operators to have user-defined meanings on user-defined types (classes). They are syntactic sugar for equivalent function calls; ex:

```
class X {
  //...
public:
  //...
};
X add(X, X);      //fcn that adds tto X’s
mul(X, X);        //fcn that multiplies two X’s
X f(X a, X b, X c)
{
  return add(add(mul(a,b),mul(b,c)),mul(c,a));
}
```

Now merely replace ‘add’ with ‘operator+’ and ‘mul’ with ‘operator*’:

```
X operator+(X, X);    //fcn that adds two  X’s
operator*(X, X);      //fcn that multiplies two X’s
X f(X a, X b, X c)
{
  return a*b + b*c + c*a;
}
```

**Question 19: What operators can/cannot be overloaded?** Most can be overloaded. The only C operators that can’t be are ‘.’ and ‘?:’ (and ‘sizeof’, which is technically an operator). C++ adds a few of its own operators, most of which can be overloaded except ‘::’ and ‘.*’.

Here’s an example of the subscript operator (it returns a reference). First withOUT operator overloading:

```
class Vec {
  int data[100];
public:
  int& elem(unsigned i) {if (i>99) error();
  return data[i]; }
};
main() {
  Vec v;
  v.elem(10) = 42;
  v.elem(12) += v.elem(13);
}
```

Now simply replace ‘elem’ with ‘operator[]’:

```
class Vec {
  int data[100];
public:
  int& operator[](unsigned i) { if (i>99) error();
  return data[i]; } };
main() {
  Vec v;
  v[10] = 42;
  v[12] += v[13];
}
```

*Dr. Marshall P. Cline is the founder and president of Paradigm Shift, Inc., a firm that specializes in on-site training for C++, OOD, OOA, consulting, and reusable/extensible C++ class libraries. For more information, send e-mail to* <info@parashift.com> *or call 315-353-6100.*

# A Concise Guide to UNIX Books

***Compiled by: Samuel Ko (kko@sfu.ca, sko@wimsey.bc.ca)***

*Submitted by Andrew Trauzzi*

*This monthly column will briefly cover some good UNIX books available on the market today. This month we will look at General beginner UNIX texts.*

**Unix for Dummies**
John Levine and Margaret Young
1993 ISBN 0-878058-58-4

- A fairly informal (funny) and non-technical introduction to Unix.

**The Unix Operating System**
Kaare Christian
2nd ed. 1988 ISBN 0-471-84781-X

- A classic overview of Unix commands. Good in coverage.

**A Practical Guide to the Unix System V Release 4**
Mark Sobell
2nd ed. 1991 ISBN: 0-8053-7560-0

- A very good tutorial / reference book.

**The Waite Group's Unix System V Primer**
Mitchell Waite, Donald Martin and Stephen Prata
2nd ed. 1992 ISBN: 0-672-30194-6

- **Highly Recommended.** A very good hand-holding tutorial-type book for Unix/SVR4.

**Mastering SunOS**
Brent Heslop and David Angell
1990 ISBN: 0-89588-683

- A good, comprehensive hand-on text to SunOS and OpenWindows. BTW, the authors have also written a book on Solaris 2 — **Mastering Solaris 2** (ISBN: 0-7821-1072-X)

**Peter Norton's Guide to Unix**
Peter Norton and Harley Hahn
1991 ISBN: 0-553-35260-1

- Good coverage. A good introduction for beginners (especially those accustomed to DOS).

**A Student's Guide to Unix**
Harley Hahn
1993 ISBN: 0-07-025511-3

- Highly Recommended. A superb introduction to Unix. In a clear and lively language, the author tells the novice users everything they want to know about Unix and the Internet — covering Unix commands, utilities, shells, vi, X-Window, e-mail, netnews, ftp, gopher, etc. It should be an excellent textbook for any Unix introductory course. No experience necessary!

**Unix System V Release 4: An Introduction**
Kenneth Rosen, Richard Rosinski and James Farber
1990 ISBN: 0-07-881552-5

- A very comprehensive text targeted to novice users. BTW, the authors have written a new book — **1001 Unix Tips** — (ISBN: 0-07-881924-5).

**Guide to the Unix Desktop**
Chris Negus and Larry Schumer
1992 ISBN: 1-56205-114-8

- A fine tutorial / reference text on SVR4.2.

**Learning Unix**
James Gardner
1991 ISBN: 0-672-30001-X

- With disks containing MSDOS simulation of Unix (MKS Tools). A good tutorial / reference book for those without constant access to Unix.

**Portable Unix**
Douglas Topham
1992 ISBN: 0-471-57926-2

- A pretty good task-oriented quick reference.

**Learning the UNIX Operating System**
Grace Todino, John Strang and Jerry Peek
3rd ed. 1993 ISBN: 1-56592-060-0

- A fairly concise introduction for (Internet) users new to Unix.

**Life with Unix — A Guide for Everyone**
Don Libes and Sandy Ressler
1990 ISBN: 0-13-536657-7

- Highly Recommended. An everything-you-want-to-know-about-Unix book. It includes info you might not find elsewhere. " This book is the "other" book about Unix ... a study in reading between the lines — which is very much what learning UNIX is like. "

**The Unix Industry**
Ed Dunphy
1991 ISBN: 0-89435-390-X

- Covering Unix technology and the structure of the Unix marketplace.

*Next month, we'll look at general intermediate/advanced UNIX texts.*

## The Fortune File

### Submitted by Andrew Trauzzi

A computer engineer, systems analyst, and computer programmer were driving through the mountains on the way to the Mega-Widget annual convention. They had just levelled off from a steep climb, when a large downward slope appeared. The engineer (the driver) was beginning to apply the brakes to slow down their decent, but realized that the brakes were beginning to fail. Panicked, the engineer pulled on the handbrake and dragged the car against the guardrail. The car began to slow down, but it looked like it wasn't going to be enough. Miraculously, the car slid to a halt just meters from the edge of the cliff. The engineer jumped out and exclaimed, "I know exactly what happened, and I can fix the problem right here on the spot!" The systems analyst disagreed and said, "I think we should have the car towed into town where a skilled professional can fix the problem." The programmer disagreed with both and stated, "I think we should first drive it back down the hill to see if it does that again!"

# UNIX Q&A

## *Originally Compiled by Ted Timar*

*Submitted by Andrew Trauzzi*

*UNIX Q&A is a monthly column that will hopefully answer some commonly asked UNIX questions. If you have any specific questions, please submit them to Monsieur Ex via e-mail <m-ex@muug.mb.ca>.*

**Q1. How do I rename “`*.foo`” to “`*.bar`”, or change file names to lowercase?**

Why doesn’t “`mv *.foo *.bar`” work? Think about how the shell expands wildcards. “`*.foo`” and “`*.bar`” are expanded before the `mv` command ever sees the arguments. Depending on your shell, this can fail in a couple of ways. `CSH` prints “`No match.`” because it can’t match “`*.bar`”. `SH` executes “`mv a.foo b.foo c.foo *.bar`”, which will only succeed if you happen to have a single directory named “*.bar”, which is very unlikely and almost certainly not what you had in mind.

Depending on your shell, you can do it with a loop to “mv” each file individually. If your system has “`basename`”, you can use:

C Shell:

```
foreach f ( *.foo )
  set base=`basename $f .foo`
  mv $f $base.bar
end
```

Bourne Shell:

```
for f in *.foo; do
  base=`basename $f .foo`
  mv $f $base.bar
done
```

Some shells have their own variable substitution features, so instead of using “basename”, you can use simpler loops like:

C Shell:

```
foreach f ( *.foo )
  mv $f $f:r.bar
end
```

Korn Shell:

```
for f in *.foo;
  do mv $f ${f%foo}bar
done
```

If you don’t have “`basename`” or want to do something like renaming `foo.*` to `bar.*`, you can use something like “`sed`” to strip apart the original file name in other ways, but the general looping idea is the same. You can also convert file names into “`mv`” commands with ‘`sed`’, and hand the commands off to “`sh`” for execution. Try

```
ls -d *.foo | sed -e 's/.*/mv & &/' -e 's/foo$/bar/' | sh
```

Shell loops like the above can also be used to translate file names from upper to lower case or vice versa. You could use something like this to rename uppercase files to lowercase:

C Shell:

```
foreach f ( * )
  mv $f `echo $f | tr '[A-Z]' '[a-z]'`
end
```

Bourne Shell:

```
for f in *; do
  mv $f `echo $f | tr '[A-Z]' '[a-z]'`
done
```

Korn Shell:

```
typeset -l l
for f in *; do
  l="$f"
  mv $f $l
done
```

If you wanted to be really thorough and handle files with ‘funny’ names (embedded blanks or whatever) you’d need to use:

Bourne Shell:

```
for f in *; do
  g=`expr "xxx$f" : 'xxx\(.*\)' | tr '[A-Z]' '[a-z]'`
  mv "$f" "$g"
done
```

The ‘`expr`’ command will always print the filename, even if it equals ‘`-n`’ or if it contains a System V escape sequence like ‘`\c`’.

Some versions of “`tr`” require the `[` and `]`, some don’t. It happens to be harmless to include them in this particular example; versions of `tr` that don’t want the `[]` will conveniently think they are supposed to translate ‘`[`‘ to ‘`[`‘ and ‘`]`’ to ‘`]`’.

If you have the “`perl`” language installed, you may find this rename script by Larry Wall very useful. It can be used to accomplish a wide variety of filename changes.

```
#!/usr/bin/perl
# rename script examples from lwall:
#       rename ’s/\.orig$//’ *.orig
#       rename ’y/A-Z/a-z/ unless /^Make/’ *
#       rename ’$_ .= “.bad”’ *.f
#       rename ’print “$_: “; s/foo/bar/ if
#       <stdin> =~ /^y/i’ *
$op = shift;
for (@ARGV) {
  $was = $_;
  eval $op;
  die $@ if $@;
  rename($was,$_) unless $was eq $_; }
```

**Question 2: When someone refers to ‘`rn(1)`’ or ‘`ctime(3)`’, what does the number in parentheses mean?**

It looks like some sort of function call, but it isn’t. These numbers refer to the section of the “Unix manual” where the appropriate documentation can be found. You could type “`man 3 ctime`” to look up the manual page for “ctime” in section 3 of the manual. The traditional manual sections are:

1 User-level commands
2 System calls
3 Library functions
4 Devices and device drivers
5 File formats
6 Games
7 Various miscellaneous stuff — macro packages etc.
8 System maintenance and operation commands

Each section has an introduction, which you can read with “`man # intro`” where # is the section number.

Sometimes the number is necessary to differentiate between a command and a library routine or system call of the same name. For instance, your system may have “`time(1)`”, a manual page about the ‘`time`’ command for timing programs, and also “`time(3)`”, a manual page about the ‘`time`’ subroutine for determining the current time. You can use “`man 1 time`” or “`man 3 time`” to specify which “`time`” man page you’re interested in.

You’ll often find other sections for local programs or even subsections of the sections above — Ultrix has sections 3m, 3n, 3x and 3yp among others.

# GNU Review

## *By Peter Graham*

Hi again. I'm back writing on the deadline date for submission. Just can't seem to break the habit of being deadline driven.

### Correction

Anyway, before we start this month's review, I should clarify something regarding last month's column on the mtools package. Stephen Keeling was kind enough to e-mail me and point out that some things I said (or did not say) about mtools were somewhat misleading. This was unintentional and I had best clear it up right away. (Egads, my first published correction already!!!) Firstly, I intentionally limited myself to the capabilities of the mtools package alone including its annoying filename restrictions. Other packages, some of which may be included with various Unix systems, correct these deficiencies.

Secondly, I made a statement "...restricted to the disk size/density on your workstation's floppy." In hindsight, the meaning of this was very unclear. My intent was to point out that non PC-based Unix systems typically only have 3.5" floppies and you are therefore restricted to the densities provided by that single drive. Unfortunately, it came out sounding like you could only support one density on your drive. This is not correct — during installation, you edit a '`devices.h`' file to describe and identify the floppy drive(s) on your machine. Each drive can be configured to support multiple densities.

Also regarding last month's column, my thanks to Andrew for the spiffy 3D table effect. Lets see if you can outdo yourself this month. (Aside: I guess this also checks how carefully Andrew proof reads the text I send him :-> ).
*(Sorry Peter, nothing like that this month, but watch for the 3D glasses included in the next issue of MUUGLines! — ed.)*

### oleo spreadsheet

This month I decided to knuckle down and install the Gnu spreadsheet product "oleo" which I have been planning to do for some time. Oleo supports both text-based and X displays and appears to be based on an older, text-only program called 'sc' (Spreadsheet Calculator). Overall I was pleasantly surprised by oleo. It is by no means as slick as some of the MAC and Windoze based products in terms of its user interface (e.g. you can't sweep to select cell ranges) but it offers some impressive computational abilities. For someone who is using Unix and has a casual need for a spreadsheet program (e.g. for quick and dirty data analysis), it is ideal.

### oleo installation

The basic installation is a standard Gnu install, but Oleo also comes with a couple of special instructions in the file '`INSTALL.OLEO`' which tells you how to compile for unusual X11 clients. For my machine, the installation procedure was just '`./configure`' followed by a '`make`' and a '`make install`' as root. Pretty easy, eh?

### oleo features

Oleo provides the basic spreadsheet features including absolute, relative, and mixed cell references, the expected builtin arithmetic operations (e.g. +,-,*,/, etc.) as well as boolean, logarithmic, and trigonometric functions. The expected functions which operate on cell ranges are also present (sum, prod, max, min, cnt) and are augmented by basic stats functions such as sum, standard deviation, and variance.

Oleo supports special "packages" for particular applications. These include a string package for string-intense applications, and a business package which includes functions for computing mortgage payments, depreciation, and the like.

Oleo also provides customization options for the more advanced/frequent user including user-defined key mappings, variables, macros, and display formats. Finally, printing of arbitrary spreadsheet sections on PostScript printers is supported by commands which generate encapsulated PostScript files for "offline" printing.

Obviously, all the capabilities of any reasonable spreadsheet package are far too extensive to enumerate in a couple of paragraphs. There is much more in Oleo than I have discussed here. The best way to find out about oleo is to grab and install a copy and try it out. The basic usage instructions are in a file called "`USING`" which is included in the distribution. Online help is also available.

### X11 support

If, during the installation process, '`libX11`' is found, then oleo is automatically built with X11 support. When using oleo with X11 you have the obvious advantage of the far larger spreadsheet size (as compared to a 24x80 ASCII terminal). Standard functions are also bound to the mouse buttons ('Select Cell', 'Mark Cell', 'Select and Mark Cell' by default). Default fonts used may also be changed. In release 1.5, oleo does not read your `.Xdefault` file for resource-based customization, but the designers promise to remedy this in the next version.

### oleo Summary

| | |
|---|---|
| Name | oleo |
| Description | Text and/or X based spreadsheet program. |
| Archive Loc'n | prep.ai.mit.edu: /pub/gnu/oleo-1.5.tar.gz |
| Archive size | 380906 bytes |
| Approx Space to Install | 1MB |
| Time to Install (Sparc-1) | 11 minutes |
| Pros | • Free, small, and easy to install.<br>• All the standard spreadsheet features.<br>• Nice set of builtin functions.<br>• Input based on emacs => powerful and familiar.<br>• Supports user-definable key maps for personalization.<br>• Since source code is available (and readable) extension for special purpose application domains is both possible and relatively easy. |
| Cons | • Somewhat limited user interface when compared to PC products.<br>• No 'man' page — I had to whip one up myself but there is online help within oleo (CTRL-H).<br>• Documentation for usage is ad-hoc (just a simple text file complete with typos) but there is also a wall chart of key bindings which is useful. |

See you all next time. Remember, you can send any requests to <`pgraham@cs.umanitoba.ca`>.

# SIG Sideline

### By Brad West, SIG Coordinator

We had a good turnout at the last SIG meeting on Tuesday Feb 15th. Unfortunately, the sendmail presentation was delayed to the next meeting, so we continued with the round table format. The discussing lent its way mostly to Linux. One topic was the upcoming release of 1.00 of Linux — expected in the next month or so. Another point of interest was the new development of ipx and win socket into the Linux kernel.

Greg Moeller arrived with the hot new release of the Slackware distribution for Linux — all 47 3.5" HD diskettes! This distribution includes everything but the kitchen sink! If anyone is interested in borrowing the set of diskettes, you can contact Greg Moeller <gkm@muug.mb.ca>, or come to the next SIG meeting. Greg had his luggable PC at the meeting and was demonstrating Linux running, complete with X-windows. The configurations of Linux Greg had on the PC took only 60 MB of disk space showing that with Linux you can have a functional system without a large amount of disk space. The rest of the meeting focus mostly on what is included in the Linux distribution. Since there seems to be a lot of interest on this, I have included an overview of the Linux Slackware distribution.

| **Disk(s)** | **Description** |
|---|---|
| ./bootdisk | Boot / install disks for 1.44M and 1.2M floppy drives. |
| ./a1 — ./a3 | The base system. Enough to get up and running and have elvis and comm programs available. Based around the 0.99pl14 Linux kernel, and the new file system standard (FSSTND). |
| ./ap1 — ./ap4 | Various applications and add-ons, such as the manual pages, groff, ispell (GNU and international versions), term, joe, jove, ghostscript, sc, bc and the quota patches. |
| ./d1 — ./d6 | Program development. Gcc/G++/Objective C 2.4.5, make (GNU and BSD), byacc and GNU bison, flex, and the 4.4.4 C libraries, gdb, kernel source for 0.99pl14, SVGAlib, ncurses, clisp, f2c, p2c, m4, perl, rcs. |
| ./e1 — ./e5 | GNU Emacs 19.22. |
| ./f1 | A collection of FAQs and other documentation. |
| ./n1 — ./n3 | Networking. TCP/IP, UUCP, mailx, dip, deliver, elm, pine, smail, cnews, nn, tin, trn. |
| ./oop1 | Object Oriented Programming. GNU Smalltalk 1.1.1, and the Smalltalk Interface to X. (STIX). |
| ./q1 — ./q4 | Alternate kernels and kernel source. Contains precompiled 0.99pl13r and 0.99pl14a kernels and kernel source for 0.99pl13, 0.99pl13r, and 0.99pl14a. |
| ./tcl1 — ./tcl2 | Tcl, Tk, Tclx, bit, itcl. |
| ./y1 | Games. The BSD games collection, and Tetris for terminals ( The all-important Disk). |
| **X Windows Disks:** | |
| ./x1 — ./x5 | The base XFree86 2.0 system, with libXpm, fvwm 1.11, and xlock added. |
| ./xap1—./xap2 | X applications: X11 ghostscript, libgr13, seyon, workman, xfilemanager, xv 3.00, GNU chess and xboard, xfm 1.2, ghostview and various X games. |
| ./xd1 — ./xd3 | X11 program development. X11 libraries, server linkkit, PEX support. |
| ./xv1 — ./xv2 | Xview 3.2 release 5. XView libraries, and the Open Look virtual and non-virtual window managers. |
| ./iv1 — ./iv2 | Interview Libraries, include files, and the doc and idraw apps. |
| ./oi1 — ./oi3 | ParcPlace's Object Builder 2.0 and Object Interface Library 4.0. |
| ./t1 — ./t3 | Tex support. |

If anyone is interested in being a guest speaker at a SIG meeting or if you have a specific topic of interest, please let me know. I can be reach by email at <bwest@muug.mb.ca> or my work phone is 983-0336. We are planning an overview of Sendmail at our next meeting — scheduled for Tuesday, February 15, at 7:30 PM. The 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 at 7:15 PM to let everyone in. Hope to see you at the meeting.

# Agenda

### for
### Tuesday, March 8, 1993, 7:30 PM
### Samuel N. Cohen Auditorium
### St-Boniface Hospital Research Centre
### Main Floor, 351 Taché

1. President's Welcome 7:30

3. Business Meeting 7:35
   a) Old Business
   b) New Business

5. Presented Topic 7:45
   The PowerPC
   Presented by Bruce Danforth

4. Coffee Break and Informal Discussion 9:00

**Note**: Please try to arrive at the meeting between 7:15 and 7:30, to avoid disrupting the meeting in progress.

## Coming Up

**Meeting:**

Next month's meeting is scheduled for Tuesday, April 12, at 7:30 PM. Meeting location will be the St-Boniface Research Centre, as usual. The March meeting topic is software packages. Stay tuned for details.

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

**Newsletter:**

If you are interested in a particular topic, let me know. I'm sure I could coerce you into writing an article! I could use a few articles — especially shorter ones — 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>. He may be old, but he's not ready for retirement yet!