$2.50

MUUGlines

May 1998

Volume 10, Number 9

This Month at the MUUG Meeting

This month’s meeting is entitled New Math (and UNIX) on the Web, presented by Michael Doob of the University of Manitoba’s Mathematics Department

We'll look at some applications of web technology to mathematical concepts. In some cases they either give or help to give new mathematical results. In other cases they allow better visualization of mathematical concepts. We'll look at some examples that push the boundaries of mathematical communication: papers that compute results while you watch, that allow you to grab objects and manipulate them, and that are a result of cooperative efforts by many computers giving up spare CPU cycles. Almost all of these examples arise from research carried out in (and often reflect) a UNIX environment.

As usual, we will have a round-table discussion, in which anyone can raise questions regarding their experiences (or lack thereof) with all things Unix. We realize that it can be a little intimidating, but please be assured that no question is too easy (or “dumb”)! Don’t forget that we have door prizes to give away, too.

In June, we plan to meet at the Applied Multimedia Training Centre, 495 Portage. This should be a whiz-bang presentation, so be sure to attend this, our last meeting of the MUUG year. Meetings resume in September.

Our meeting this month is Tuesday, the 12th of May. We meet at IBM Canada’s offices in the TD Centre, at the corner of Portage and Main. We’ll gather in the lobby on the main floor – please try to be there by about 7:15 PM. Steve Moffat will then take us up to the meeting room just before the meeting starts at 7:30. Don’t be late, or you may not get in.

Parking is available either in the parkade behind the TD building, off Albert Street, or in the ground level lot just north of the TD building. Entrance to the lot is from Albert Street, behind the parkade. Either way, parking is a $1.25 flat rate for the evening.

Our PGP Presentation

At our February meeting, Arne Grimstrup presented “Protecting Data Transfer in an Insecure Environment.”

He began with a reminder that networks are passing much information as non-encrypted text. People who send out this information sometimes don’t realize the security risks that are involved.

People can use encryption to secure the information that they are transmitting over a network. A popular encryption program is PGP, written by Phil Zimmerman in the late 1980s. This is a public key system. There are two keys generated; one key is a private key and the other is a public key that the distributed to other users. Text encrypted with the public key can only be decrypted with the private key, and vice-versa. To send someone a secure message, for example, the sender would encrypt the message with the receiver’s public key. That means that only the receiver's private key will be able to decrypt the message.

You can also use the PGP system to sign a message. You can create a signature string for a message with your private key. The recipients of the message can use the sender’s public key to verify the signature. In this way, anyone can verify the message, but only the sender could have created the signature.

There are four programs that the PGP system uses: 1) pgpe, for encrypting; 2) pgpv, for decrypting and verifying; 3) pgps, for signing, and 4) pgpk, for key creation and management. The programs are available from the official MIT distribution site at http://web.mit.edu.

For example, to create a new key you would use the following command:

```
pgpk -g RSA
```

You can view a key:

```
pgpk -c userid@muug.mb.ca
```

To encrypt a command, you can give the following command:

```
pgpe -r <recipient email address> -u <user email address>
<filename>
```

The -a option will create an ASCII file that can be sent with a text email program. The -s option will encrypt the message and add a signature.

To sign a message, you can give the following command:

```
pgps -ba -u <user's email address> <filename>
```

The PGP program allows for secure transmission of files, but the interactive shell still sends all its commands in clear text. For example, all login userids and passwords are sent in clear text. Someone could examine the packets transmitted on the network and collect this information.

One solution to this is the Secure Shell. This system has a server component (sshd) and provides replacements for the rsh, rcp, rlogin, and telnet programs. It uses a program similar to PGP to authenticate and encrypt traffic between machines on a network.

Secure shell was written by Tatu Ylonen. A web page that discusses ssh is available through http://www.cs.hut.fi/.

KDE Beta 4 - A User Review

by Mike Haas (gnosis@infinet.com)

I suppose the following review will receive a fair amount of negative feed back for several reasons; KDE’s use of Qt, its similarities to that other OS, etc. I won’t address that here.

What I will address, is the pleasure of running a WindowManager that is not only “hands-off” with regards to editing .rc’s, (save for setting the LD Library path in /$User/.profile, and adding “startkde” to /$User/.xinitrc), but is surprisingly complete and STABLE for a “beta” WindowManager. But one only has to use KDE for a few minutes to understand that it is far more than just a “WindowManager”, and has features you simply won’t find in other popular WindowManager distributions.

With the plethora of K-applications which come with the distribution, KDE has all or most of the features you would expect to find in other full-blown OS’s and pound for pound, has far more flexibility than CDE, which I’ve used, and can honestly say is FAR from what its cracked up to be.

The Beta 4 distribution installed itself well over my Beta 3 distribution, leaving my desktop settings intact. I would like to point out a few tips to beginners, to make their usage of KDE a more pleasurable.

1. Wallpapers KDE can change wallpaper on the fly. However, KDE’s desktop manager will NOT overwrite a root window you create if you chose to stick XV or xpmroot in your.xinitrc! You can add all the images you want to /opt/kde/share/wallpapers in .gif, .jpg, or .xpm format. Start KDE raw, with no external root window image. KDE will start with its default background colour. You can then insert Wallpaper on the fly with the desktop manager. KDE comes with its own assortment of images, and you can add to these to your hearts content.

2. KSoundManager By default, KDE starts with its CD player and mixer running. If you hit the “X” widget to close them, they don’t - they are swallowed by kpanel and show up on the panel in a manner similar to Win95’s Systray.

Shut ‘em off. You will notice a *remarkable* improvement in speed if you disable this default setting. By all means, use KDE’s CD player and Mixer if you wish, they work fine. I believe that its the Kpanel “swallow” feature that is the culprit here, and not necessarily any “bloat” from these two K-applications. Once I turned this feature off, KDE became quite snappy on my Pentium 166 MMX w/ all of 24M of ram.

3. Installing applications and Icons You’re gonna love this. The installation of applications and icons is a breeze and is performed in a manner similar to Win95. You can even drag an application right out of its directory on to the desktop as a link, but this is the least flexible method.

KDE stores its menu information in /opt/kde/share/applnk, in associated sub-directories such as “Applications”, “Graphics”, “Internet”, etc., much in the same manner as Win95 uses c:windowsstart menuprograms. Applications installed in these directories show up in the “K” menu, whereas applications installed on the desktop have their respective location in /$User/Desktop. Either location takes the same form - *.kdelnk. Right clicking on a distribution-installed application will give you an idea of what information you have to enter, basically the name, location (if the app isn’t in your path), and what you want in the pop-up hints. The .kdelnk dialogue is much more flexible than that, including commands for iconifying and running the app swallowed in Kpanel.

Icons? This is the best part. Under the “Execute” tab of the .kdelnk dialogue, is the icon button. This defaults to /opt/kde/share/icons. You’ll get a pop-up window which shows thumbs of all the .xpm formatted icons you have to choose from, and again, you can add icons here to your hearts content (as of 4/24/98, the Incoming directory of ftp.kde.org has a file called kdeicons.tgz, which has a nice set of additional icons copied after NeXT, Win95, and BeOs, etc.).

The cool part is this: Say you simply dragged the Netscape icon from /usr/local/netscape and placed it on the desktop as a link to get things started, then later, decided to install Netscape with an appropriate icon in either /$User/Desktop, or /opt/kde/share/applnks/Internet. The linked icon would have showed up on the desktop as the “gear” icon. After installing Netscape and restarting Kpanel, voila - your original linked icon appears on the desktop with the new icon image! Oh, the joys of mime-types... In fact, mime-types are handled so well in KDE that I can see the day when KDE might even use a full-blown mime-type database. I just hope they call it something other than a “registry”...

4. KFM KFM has improved significantly from previous releases and behaves in a manner similar to Win-Explorer. In fact, whereas is previous releases, .html was handled through the “help” application, it is now handled through KFM. I’ve used both QTScape and KFM to browse the web, and KFM, as of the writing of this review, handles things at least as well, if not better, than QTScape. Both applications will certainly continue to improve, but its nice to have *TWO* other Web Browser choices than Netscape!

Right-clicking on a application in KFM will allow you to pull up properties - you can set the permissions of the application on fly here, which is quite handy. All of KDE is drap-n-drop enabled (NOT dndlib-based!) and “cut and paste” copying/moving of files works very well.

5. Desktops I only have one slightly negative comment here. By default, KDE allows you to scroll through desktops by moving the mouse above or below the desktop window - convenient for some, but with my particular PS/2 mouse, I find it annoying...

6. KDE applications They rock - I just can’t describe them any better than that. I run RH 5.0, and because 5.0 does NOT come with ppp-on or ppp-off scripts, you generally have to run the control-panel and turn on dial-up networking from there. Kppp works very smoothly, although I had to enter a script - I couldn’t get PAP login working, not that I really spent anytime playing with it. My login script is simple enough that I was able to enter it and get Kppp running in minutes.

KDE comes with a host of very decent Internet applications. Kmail works extremely well and is very configurable, as is its Usenet Newsreader, Krd. KDE even comes with a network utilities application (ping, traceroute, etc.).

In conclusion, KDE Beta 4, with all of its onboard applications and utilities is not only a fine choice for any Linux user running X., but is an excellent candidate for inclusion into a Linux Distro as the default WM. For the first time, a freely-available WindowManager which requires NO external editing of script files and is more powerful than, IMHO, any other WindowManager on the market (including the commercial CDE) is available for Linux. I've literally run 'em all, with WindowMaker being my favourite - until now. Any first-time user who can run Win95 with any facility can make great use of KDE.

Perhaps my opinion will change when Gnome hits full-blown (but running in E???), or when the GNUstep project produces its first Beta WindowManager, but its like the old saying goes - "don't knock it until you've tried it".

One LINUXer's Opinion

Linux is as much a community as it is an operating system. The community is not extremely different than the Macintosh community in that it has diehard followers, and some of which tend to be almost ready to go to radical extremes, and zealous in their practices and attitudes. There is a distinct difference between the two, they are at opposite ends of the "user-friendly" spectrum.

Another distinct difference is the attitude of some (small but vocal section) of the Linux community in regards to the "newbie." About the only way I know to describe it is much like the attitudes of surfers in Hawaii and California in the 80's, a "locals only" attitude. I am not the first to suspect that there is an underlying "hackers only" attitude in the Linux community, as well as how the world sees Linux today.

To be completely honest, Linux has many great features and dimensions that other operating systems (OS's) lack. Linux took to the Internet with the GNU license shortly after it was created by Linus Torvalds. With a birth like this, networking abilities are hard to match, because they were crucial for development itself. And there is no denying that GNU itself is unmatched in what it fundamentally brings to the world of computing. Software, constantly underdevelopment, worked on around the world by large teams of people who may never meet face to face, and available for free to the public. What more could anyone ask for? Well, honestly, a lot.

Rapid development is a blessing and a curse all rolled into one. When you consider that in 1991, the it all began, and by 1994 it reached v1.0, it is quite a remarkable achievement. Something that reached a working level in a short time. And when one looks at what has happened in the last 2 years, or even 6 months, it is truly remarkable, because it has become a full fledged, fully functional, competitive OS in a market of sharks (SUN, Microsoft, Apple, etc.). A true alternative, freely available, and capable of providing strengths that commercial offerings lack.

But there is a drawback, and IMHO, that is a complete lack of standards. Options are one thing, and standards are a totally different issue the way I see it. GNU development of different xwindow managers is great, and the on going battle between GNOME and KDE does lead to healthy competition. But there could be some competition in the Linux community that is unhealthy, and causing more problems than good. Software installation is one of these issues.

There is no doubt that Linux is a great development platform, and capable of turning a standard "PC" into a real "workstation" with even more of a personal feel than any other "PC" OS out there. I can run wm2, work with X applications, and keep all my system stripped down and command line based, and my binaries will run just fine on a system running KDE that looks totally different and much more "user friendly." But, getting the applications to run in the first place is a chore many times.

Look at what is out there, there are "source tarballs" (tar.gz) for some applications, then there is Red Hat package management (.rpm), and Debian packages to mention yet another. All have their up side, and their down side.

First consider the idea of compiling from source. Seems like a great idea in theory, you stand a better chance of having it run on more platforms. You also gain the advantage of doing an optimized compile to get the application best suited for your specific system. How many times do you have to ftp something for 45 minutes, "tar -zxvf" it, and struggle with "Error (1)" when the compile fails before you get sick of it? And how do you know that the libraries you have installed were the same libraries that the developers were using? Seems the libraries themselves can be updated more frequently than the applications themselves sometimes! This process is no match for the ease of "WinShield" no matter how you slice it, and problems like this will keep Linux from ever going main stream. My mom will never run Linux when she can buy a CD of Quicken, install it in 10 minutes without any worries of ever even seeing a command line, and then she can start to struggle with how to use the menus (face it, a very large percentage of the population using computers has a hard time using GUI apps that would seem idiot proof already).

The rpm system has the advantages that it comes as a binary that you download for your platform, and can be installed with a simple command such as "rpm -ivh program.rpm" without having to deal with compiling. Well, rpm's aren't perfect either. I haven't meet an "rpm" user yet who hasn't been subject to the "failed dependencies:" syndrome and ended up hunting sunsite and other sources for even more rpm's to get the first one to install. Then you frequently run into the "conflicts with" problem, and you can't get the thing to install anyway. And, rpm binaries don't allow you to do any optimization for your specific system.

I have used SuSE, Red Hat, Open Linux, Linux Pro, and others, and frankly, they are all so different that it's hard to even get help on some problems in Linux newsgroups and IRC channels. Directory structures differ, packages of the same programs differ in name, and attitudes grow intense. Fractional divisions occur in the Linux community it self where the Slackware folks don't want to even help someone if they know how to solve the problem but then find out that it's on a Red Hat system. As a community that is the underdog in the OS world to start with, we are fighting amongst ourselves.

Some of the free UNIX people I talk to on Effnet frequently are FreeBSD pilots, and I have been doing a little testing remotely through shells. There is obviously a distinct advantage to having a standard directory structure. The ports system is fairly useful also, allowing the downloading and compiling of applications without the "error (1)" or "failed dependencies" headaches. To some people it doesn't matter, it's just a challenge, but to a very large section of the population, a computer is to be used to accomplish a goal, not to challenge their "make" error debugging skills.

Then there is the matter of the commercial support. There are commercial Linux distributions, but not many commercial applications. I have no problem with someone charging $50 or $100 for their distribution of Linux, because they are not selling Linux, they are selling the packaged installation, media, and technical support. The product you pay for isn't the OS itself, it's the support and package they offer. But what about the applications? There are clearly two distinct problems that keep companies away from marketing Linux applications. First, it's not yet a large market share, and their profits may not be as great as they would for Macintosh or Windows. Secondly, if they were to sell their application, what do they package it like? Not tar.gz because they don't want to hand out the source, not .rpm because that is even a smaller fraction of the Linux market, so it's some type of script that hopefully finds it's way to install the application in the right directory, without conflicting with other applications, and successfully in all users paths. Good luck, there is no directory that is sure to be in all users paths, there is no standard file structure in Linux to put the application in (some have X11, some have X11R6, some have one with a link to the other, some have both).

Linux is soon approaching v2.2.x, and soon after that point arrives, we can all count on all the major distributions of Linux coming out with new versions of their distribution. To me, the timing seems perfect to solve some of these problems. Of course, it wouldn't be easy, it would involve a lot of work, but the benefits would be far reaching. Making Linux into an OS that the masses could use isn't really all that impossible to do. And, even if Linux were to reach the mass market, it wouldn't detract from it's remarkable ability to be a well functioning workstation and great development platform.

If we, as a community, could work together, and develop some standards, we could gain a market share, make our life's easier in the long run, and gain the support of the commercial application and hardware businesses. I see, again IMHO, some steps we could take to do this. I will not suggest the standards, that should be decided by the whole community. But I will suggest what areas I believe where there should be standards. And when v2.2.x hits the streets, wouldn't it be great if along with a greatly improved kernel, we actually had a great complete product, not just the kernel?

1) directory structure standardized. With each v2.2.x kernel there are guidelines to where things go. Are some applications going to go in /opt or in /usr? Which applications? Where? Why? Why should I have some window managers in /opt, some in /usr, some in /usr/X11/bin....

2) a system like FreeBSD's ports. Not their system exactly, but something similar. Untar and unzip a directory structure that links directly to an application listing. Categorized for ease in finding applications you want. Standardized so that you can count on your compiling (or whatever installation method) working the first time without getting missing library errors, failed dependencies, etc... We already have access to some of the greatest distribution servers in the world, like sunsites archives! That would be very useful.

3) An update notification system for applications. Just because we have standards doesn't mean development has to stop! If a newer version of an application you frequently use comes out, it would be great to know about it, know where to look for it, and be able to install it easily. Some sort of tool to accomplish this would be great. Like a application that you set up to check in the standard "ports" collection for newer versions of the applications you use. This would take planning, would it be a daily updated thing created by the ftp server that lists all the packages you can download, unzip, and compare to your systems installed applications? Would it an application that searched the known standard directory structure on one of the Linux ftp servers to seek out the applications you want to monitor? There are ways to solve this problem, but the community has to decide its important enough to do.

Frankly, I think we could do it. I would put some time in to help out where I could. I would love to see it happen. I would love to be able to get more reliable use out of my Linux system. I would be happy to see Linux gain a market share. I would like to make my workstation more suited for my work and take less of my time to constantly reconfigure. Frankly, at this point, I am thinking of switching to the FreeBSD camp to get it, but I would like to see it happen in Linux.

Robert Current (aka BadlandZ) current@plains.nodak.edu

Contact Information

To contact the MUUG board for membership information or anything else, send e-mail to board@muug.mb.ca. We have a Web presence as well, at http://www.muug.mb.ca/, where you can find all kinds of information, including details of upcoming and past meetings and presentations and references related to them. E-mail the editor at editor@muug.mb.ca.