$2.50

MUUGlines

September 1998

Volume 11, Number 1

Don’t miss this meeting: September 8, 1998 at Applied Multimedia Training Centre!

Learn the power of Silicon Graphics (SGI), IRIX-based, 3D Animation Applications. Applied Multimedia Training Centres' (AMTC) Animation Instructor Larry Mersereau will discuss and demonstrate the latest in Computer Animation Technology as used in blockbuster special effects movies like Titanic and Jurrasic Park. These features are computationally intensive and rely heavily on the high-end, real-time graphics capabilities of the new SGI O2 Unix systems. Larry will go in-depth on the advanced software functions of Softimage -- Dynamic Simulation, Displacement Mapping, Ray-Tracing and the use of Particles to create those incredible 3D effects.

Please note our meeting location, for this month only: The Applied Multimedia Training Centre is located at 495 Portage Avenue (the same building as the Bus Depot). Parking is available on Portage Avenue and in the Bus Depot Parkade (entrance off Colony Street). When entering the building after hours via the parkade, take the parkade elevator to the basement and follow the signs to AMTC's labs; when entering off of Portage Avenue after hours, enter through the Bus Depot entrance, turn left and follow the signs.

As usual, we will also 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. This month a lucky attendee will go home with a copy of O’Reilly’s “Linux In A Nutshell”. We also have the two most recent copies of Linux Journal to give away.

Next Month - October 13, 1998: Low-cost Supercomputing with Beowulf and PVM. Dr. Peter Graham, of the U of M's Computer Science department, will be describing (and, we hope, demonstrating) how a cluster of inexpensive Linux boxes, using the PVM software, can be made to operate as a Beowulf-class parallel machine.

Linux Install Fest 1998

Friday, September 25, 14:00 to 17:00
Saturday, September 26, 12:00 to 17:00
Room 130 Machray Hall
Fort Garry Campus
University of Manitoba
Winnipeg, Manitoba, Canada
Free admission, free parking on Saturday!

The Manitoba UNIX User Group and the University of Manitoba Department of Computer Science are pleased to announce the Winnipeg Linux Install Fest 1998, to be held in Winnipeg, Manitoba, as part of the National Install Fest 1998, organized by the Canadian Linux Users' Exchange.

T he event will be held over two days: Friday, September 25, from 2pm to 5pm, primarily for the benefit of U of M staff and students, and Saturday, September 26, from 12 noon to 5pm, to coincide with the National Install Fest. The general public is invited to attend on either or both of those days. There is no cost to attend, and all are welcome.

Parking is available for free on Saturday in the K-lot and A-lot, except where otherwise indicated. Some meter parking (K- and A-lots) and ticket dispenser parking (K-lot) will be available on Friday, for those wishing to attend on that day.

What is a Linux Install Fest?

Since Linux rarely comes pre-installed on computers, and many people are uncomfortable installing an operating system, many Linux user groups have organized install fests as an opportunity for any interested computer user to have a copy of Linux properly installed on their own computer. Experienced Linux users will be on-hand to demonstrate the use of Linux, and to answer your questions. This will allow you to get a better understanding of what Linux is, see demos of the system in action, and get free assistance from technically competent people with issues involved in the installation and setup of Linux.

Who should attend?

If you're simply curious about what Linux is, and what all the fuss is about, come on down and have a look. And don't be afraid to ask questions. If you've already decided you want to have Linux, bring your system down, and we'll help you get it installed on your machine, for free!

Don't want to lug your machine down, or can't stick around for long enough? Come on down anyway, and have a look at the installation process on someone else's machine. You'll also be able to pick up a Linux installation CD, for only a few bucks, and you can do the installation for yourself later.

Already know how to do a Linux installation? Come on down anyway! You'll maybe pick up a few handy tips from other users, or you may be able to help others out with their installation problems.

Contacts

For more information, contact Gilbert Detillieux at 474-8161 (day). You can also send your questions by e-mail (to info@muug.mb.ca) or by fax to 474-7609.

Further details, including links to campus maps, can be found on the MUUG web server:
http://www.muug.mb.ca/ifest/

Corel WordPerfect for Linux Prerelease

As you may know, Corel WordPerfect 8 for Linux will be shipping in September. But as a courtesy to the Linux community, Corel Corporation is making a pre-release version available for free download via FTP. This pre-release version is feature-complete, and includes all the great features you've heard about, such as:

Compatibility - file format is consistent with previous versions, ensuring compatibility with existing documents
Shadow Cursor - point, click and begin typing anywhere on the page
Grammar-As-You-Go™ - identifies phrases in question and provides suggestions
Corel® Versions® - keeps track of document revisions for workgroup collaboration
Drawing Layer - draw objects directly on top of text, group graphics, and add color gradient fills or patterns
Table Enhancements - size table rows, and join and split cells with a single click
Internet Publishing Capabilities - edit or create your document in Corel WordPerfect 8 and publish it to HTML
Guidelines - changes margins and columns directly on your page
Hyperlinks - automatically creates Internet links from text beginning with www, ftp, mailto and more
Spell-As-You-Go™ - identifies words that might be misspelled so you can correct them as you type

See the state-of-the-art in native Linux word processing. To download your copy, go to ftp://ftp.sdcorp.com/pub/demos/linux/wp8prerelease.

Jikes!

by Frank Carlos, alphaWorks - Production Manager
http://www.alphaworks.ibm.com/

Jikes for the Linux Platform is available for downloading. The Linux community has been anxious for the Jikes compiler to be released to them and now the wait is over!

http://www.alphaworks.ibm.com/formula/Jikes

Jikes is a Java compiler that translates Java source files as defined in The Java Language Specification (Addison-Wesley, 1996) into the bytecoded instruction set and binary format defined in The Java Virtual Machine Specification (Addison-Wesley, 1996). Unlike other compilers, Jikes accepts the Java language only as specified: not as a subset, variant, or superset.

In addition to strictly adhering to specifications, Jikes is faster than most compilers and can compute the complete dependency relations in program files. This allows for the generation of dependency makefiles suitable for use with make. Jikes can also be run in incremental mode: after the initial compilation, Jikes waits until prompted, determines which files have been changed, and then does the minimal amount of work needed to bring the class files into a complete and consistent state. This cycle can be repeated until the q command is entered to end the compilation.

The Linux Compatibility Standards Project

Debian and Red Hat are collaborating on a written specification of the "Linux Compatibility Standard" (LCS). The LCS will clearly outline the specifications necessary to be LCS compliant. This LCS document will be used as a guideline on how to build a "proper" system by those building Linux distributions, and it will be used by developers needing information on the basic system configuration to develop application programs. We expect this LCS effort to complement the example LSB implementation on which Bruce Perens is working.

Erik Troan (Red Hat) and Dale Scheetz (Debian) will jointly manage the LCS working group, which will be working on the written standard using the lcs-eng@lists.debian.org mailing list for the discussion. lcs-eng@lists.debian.org is an open subscription list. Everyone who is interested is encouraged to subscribe. Most of all, we encourage all current members of the LSB committee to subscribe to this list. Even if you intend to work on the code base, subscribing will be helpful.

Subscribing

To subscribe to the lcs-eng mailing list send an e-mail to lcs-eng-REQUEST@lists.debian.org with the single word subscribe in the subject line. You will shortly receive a confirmation request. Follow the instructions and you will become subscribed to the list.

For further information, please send email to press@debian.org or visit the Debian homepage at http://www.debian.org/. For more information about Red Hat, please visit http://www.redhat.com/.

Linux 2.2 and the Frame-Buffer Console

Please send all comments, complaints, etc on this document to its author, Joseph Pranevich, at knight@baltimore.wwaves.com. Feedback is appreciated and always welcome.

Since my initial release of my 'Wonderful World of Linux 2.2' document (a simple list of feature additions for Linux 2.2 for the non-techy user), I have received many, many questions regarding this and that. However, the questions I have been asked most frequently all revolve around Linux 2.2's newest (and most shocking) kernel addition: the frame-buffer console driver. 'What's the big deal?' you say? This document will attempt to explain exactly why this is such a 'big deal' and introduce the non-technical Linux user to the basics of the Linux video subsystem and give a number of pros and cons to the (more than just proposed) new system. Enjoy!

Before one can understand the implications of the new, kernel based, fbcon system, one must have a working understanding of how the video drivers in Linux operate. If you are already familiar with the video driver subsystem of Windows then this may come as a bit of a shock to you...

1) A Peek at What We Have Today

The current video subsystem is divided into, at best, three distinct parts: the text-mode driver, the console video driver, and the X server. Now, only the text-mode portion (in the kernel) is carved in stone, the other two bits are optional but represent a common configuration for most distributions. For the purposes of this introduction, let us assume that your console driver is the SVGAlib.

In any scenario, the only portion of the video subsystem that is truly carved in stone is the text-mode driver. (Even if you are using a serial console, the console driver manipulates the serial port as if it were any other text device.) This is a driver, in the kernel, that controls cursor positioning and video hardware features and provides a set of vt100 interfaces for the manipulation of the text data. Additionally, the kernel driver knows how to work with any special other features of your card, including and especially using video memory for scrollback, etc. But, through all this support, this kernel driver contains NO graphic primitives of any kind.

The console graphic driver (actually, a library) is the next component of the Linux video subsystem. This library starts where the kernel leaves off, by providing direct access to the graphical capabilities of many video cards. The reality of the situation here is that far too few video devices are supported in full by this library, causing it to fall into disuse.

The third and final component of this model is the X server, a driver (and more) that, like the console driver, provide a uniform interface to the graphical capabilities of your hardware. Unlike the console driver, however, the X server is also in charge of a number of network and graphic-device sharing functions that make it somewhat bloated for low-memory use and difficult to use with stand-alone applications. In addition, many X servers are not free.

2) A Preview of What We'll Have Tomorrow

The new kernel system is quite a bit simpler than the above example. In the new system, the kernel controls all direct video manipulation via the 'frame-buffer' device. This device, located at /dev/fb?, presents a device-independent API to the user-layer for manipulating any video device that you have a driver for. Under this new system, you would still need to run an X server to provide the network-connectivity of the previous system, but everyone would run the same server (XF86_FBDev) and that would merely talk to the generic interface. (A new SVGAlib is expected to be developed that will behave likewise for legacy console applications.)

3) Issues

Unfortunately, this merely scratches the surface of the issue. By all accounts, neither system looks terribly bad. And what's more, the current system is well-tested and well-accepted by the community. So, why change? The following is an assembled list of some of the defects of the systems, starting with the established trinity. I have attempted to give you the facts rather than opinions on these subjects and to allow you to best make up your own mind on the issue. I'm positive that I've left out some perfectly reasonable complaints either way, please feel free to email me with your own justifications so that I personally can make a better decision on the matter. (And, perhaps, include those reasons in a future version of this document.)

3.1) Security

The first thing that comes to my mind when discussing the problems with not having graphic code in the kernel is the issue of security. (or, lack thereof.) Linux has been designed from the start as a very security-conscious system. On a Linux system, two things really constitute a breach of security (although the first can imply the second):

> Access to 'root' (administration) things by ordinary users Causing a freeze or other denial-of-service

To these ends, it was decided (quite rightly) that allowing direct access to hardware could result in big problems. Thus, Linux allows only certain privileged programs access to hardware manipulation.

To allow a program to write graphics to the screen using the console driver, the program must be owned and run by the site administrator, root. Since it would be inconvenient to have to log in as root every-time someone wanted to access one of these programs, a flag was invented ('setuid') to allow any user to run a specific root-owned program as if he or she were root. Thus, even regular users can run graphical programs.

Unfortunately, there is a big problem here. There is no way to distinguish between a program that only wants to manipulate video hardware and one that wants to be root in other ways. Being root implies complete control over the system and all of its files, programs, and network connections. One must only hope that these graphical programs don't have any errors or problems that would cause them to go awry and trash your system...

Fortunately, nearly all programs designed to run as root do so without any adverse side-effects. However, it is possible in some cases for a black hat (cracker) to gain complete access to a system by exploiting small holes in setuid root programs. Any and all binaries that access the hardware, including X servers, can be vulnerable to this attack if the programmers are not careful. And, once an attacker gets in...

The fbcon system resolves most of this security problem by exporting an interface that requires no special privileges to manipulate. It is even possible to run the X server for fbcon without root access. In fact, the only real drawback here is that if a driver *is* buggy and in the kernel, it could take down more of the machine than just a buggy user-space X server.

3.2) Driver Availability
The current system of drivers, at least as far as the X server is concerned, is well supported. Nearly all common cards are supported to some extent in XFree86 or by an alternate X server provider. In some cases, these servers are for-pay but that is the exception and not the rule. In terms of the console driver library however, less cards are supported fully. This lack of support for the console library makes it very difficult to reasonably support it as an option for graphical development.

fbcon solves one of these problems: no longer will there be a discrepancy between what cards are supported in X and what cards are supported at the console. On the downside however, fbcon currently properly supports only a small subset of the cards that either the X server or the console library supports. We shall have to wait and see whether or not we will have a truly usable driver-base for 2.2.

3.3) Portability
The current driver trinity is a well established fact in the UNIX world. svgalib has been ported to several other UNIXes for source compatibility. The concept of having an X server to control windowing applications is older than I am. However, the current driver system is severely lacking, at the console, with portability to non-SVGA systems. Not all platforms use SVGA compatible cards as their primary video source and those that don't will not be able to build the source for our console applications.

fbcon does not solve these problems either. The new system of video drivers would allow applications designed for Linux to run more easily on other Linux platforms. All Linux platforms, even those whose current console libraries are incompatible, will be able to share graphical programs. That's quite an accomplishment.

At the same time, compatibility with other UNIXes will be severely impaired. Linux is, so far as I know, the only UNIX to use this new variety of frame-buffer system. Those platforms that still rely on SVGAlib and other legacy methods of video control may have significant problems

3.4) Configuration
My biggest complaint with the present system of drivers is the lack of any kind of global configuration system. A lot of information, including monitor timings and etc., must be present in multiple locations to keep everything running smoothly. It could be argued that a decent system administrator should have no problem maintaining this system, but the fact of the matter is that any duplication of work opens the potential for subtle errors that may take time for a busy network administrator to locate and fix.

Once again, the new drivers take a stab at solving this problem but may come up short. Under the new system, this information would only need to be repeated only once: the module-load time. As the current implementation of the fbcon system is rather incomplete, it remains to be seen exactly how such a large number of parameters would be passed into the module. (Especially considering that it is likely that the module will be loading *before* access to the filesystems can be established and a config file can be read.)

MUUG 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.