The Journal of the Association for Computing Machinery at UIUC

Banks of the Boneyard

Volume 18, Issue 2

ACM@UIUC's fifth annual Reflections | Projections

by Mike Khalili

Well it is October, and around ACM that means it time for Reflections | Projections, our annual Midwest Computing Conference. This is the fifth year for Reflections | Projections. Last year's conference is a tough act to follow for this year's conference committee. However, we believe that we have come up with the best conference yet.

The conference will begin on Friday, October 8th with the annual Reflections | Projections ACM Job Fair. This year's job fair is one of the largest of all time, with more than 35 companies. In fact, it has grown to such an extent that we have outgrown the job fair's old location of the DCL atrium and have moved into the Illini Union, rooms A, B, and C. The job fair gives students an opportunity to meet up with a many interesting companies who are interested in employing bright students in computer-related fields in a less crowded, less hurried environment than some other job fairs. We find that it is a generally useful and pleasant experience for student and employer alike.

After the job fair, we invite you to come over to the Bread Company on Goodwin Street for dinner. At the dinner, attendees will be able to sit down and enjoy a meal and a bit of conversation with each other, the corporate recruiters, and a number of conference speakers who should be arriving in time for dinner.

Saturday is the primary day for conference talks. Talks will address such diverse topics as artificial intelligence, networking, bringing computer programming to the masses, open source software, and optimizing code for such goals as security and speed. The morning will feature talks by Bill Nash, Noshir Contractor, Theo de Raadt, Ralph Johnson, and Astro Teller. There will also be a panel on the topic of Computer Science Education.

After lunch, in the early afternoon, will be the keynote address. This year's keynote is being given

continued on page 13

FROM THE CHAIR

Reflections & Projections

by Erik Gilling

Another conference is upon us. The months of hard work will finally pay off. I would like to thank everyone who has helped make this year's conference possible. I would also like to welcome and thank everyone who has come this year's conference.

As I have looked back on ones gone past, I am filled with excitement about this year's conference. Over the years our seminars have become more informational and exciting. We have expanded into a second building to accommodate this year's lectures. Unlike previous years, a main lecture room and a smaller secondary one to hold lectures is no longer sufficient. For the first many years our keynote was held in the library across from the Computer Science building. This year the keynote has expanded into the Illini Student Union.

I've also watched MechMania, our programming competition, grow in size, complexity, and graphics quality. This year we have 16 teams and a complete 3D graphics engine. We have also put much effort into making the game more fair with respect to the game rules and the tournament structure.

When I reflect on how much conference has been growing in recent years, I see many changes ahead. I expect we will need more rooms for seminars. I see our career fair growing well past 50 companies. I also expect MechMania will soon double in size and need more space, computers, and food for its contestants.

I hope everyone enjoys this year's conference and comes back next year for the sixth installment in it's great history.

In this isssue

The Banks of the Boneyard is a student publication of the UIUC Chapter of the ACM, the First Society in Computing, and does not necessarily represent, in whole or in part, the views of the University of Illinois, the Department of Computer Science, or the International ACM. Partially paid for by SORF and the Department of Computer Science.

Editing, Layout, and Design: Anne Nowinski

The Association for Computing Machinery at UIUC
1225 Digital Computer Lab, MC-258
1304 W. Springfield Avenue
Urbana, IL 61801
(217) 333-5828

Corporate liaison update

by Nathan Lau

Interested in getting a job? ACM has been blessed with several large corporations as well as many smaller firms who want to recruit students here at the University of Illinois in sponsorship with ACM. Several include the likes of Microsoft, Enterprise Information Systems, NCS, Crowe Chizek, GHS Software, and ASC Software. In these information sessions, valuable information is usually given, along with the promotions of their respective firms. The recruiting has taken off and won't cease until students fill up the positions. As many might have experienced, Engineering Expo was a catalyst for the redemption of the "corporation" in UIUC this year.

Given that many corporations want you to slave away, doing tedious amounts of work at a full time position or an internship, salaries and nice compensations have generally given precedence and have quieted the spirits of those who intend to rebel the "big brothers" of the University. Yes, the University is essentially a business model, forming their products, "students" and selling them off to corporations. Have you ever thought that classes you pick and choose are no different than training sessions held by your business? In any case, may I remind you that soon you are to be rewarded for your hard work.

Attend info sessions and attend the ACM conference! With so many opportunities, there is no better choice! Here are some info sessions coming up:
Microsoft 2240 DCL PM Oct. 7th
Crowe Chizek 1310 DCL PM Oct. 13th
GHS Software (TBA) ?? Oct. 20th, 21st ??

WEEKLY MEETINGS FOR FALL 1999

Monday
5:00PM, Exec Board
7:00PM, SigVR
7:00PM, Women In Computing
7:30PM, SigNet

Tuesday
6:30PM, SigBio
7:00PM, SigOps
7:00PM, SigDave
8:00PM, Linux Users Group
8:00PM, SigBot

Wednesday
6:00PM, SigSoft
7:00PM, Be Users Group
7:00PM, SigMusic
7:00PM, SigArt
7:00PM, SigWeb
8:00PM, WinDevils

Thursday
7:00PM, SigGraph
8:00PM, SigArch
8:00PM, SigUNIX

Friday
8:00PM, SigMobile

Saturday
3:00PM, MacWarriors

The Edge or Fundaments of Unix

by David Moore

I have spread my knowledge of computer among many friends and strangers sometimes for recreation and other times for classes and there is really only one question I hear consistently, "Dave, How do you know so much about computers?" My easiest answer is that I learned a lot of it through my father, school, work and play. I have discovered that to survive in computer-oriented classes, it is important to accrue as much knowledge as possible about using the computer before entering the class.

Students that enter any class are supposed to want to learn something about the topic, but each individual comes in with their own mix prerequisites. I came to University of Illinois with a decent familiarity with Unix and Windows. In addition, I knew C and Basic. This gave me an edge over other students who had used a computer for little more than word-processing prior or not at all. What was interesting was that the skills most useful to me were not my familiarity with C and Basic, but my ease with Unix and Windows.

A basic understanding of Unix can set the mood of a course for the whole semester. A student whom has never seen Unix before will struggle just to access the files and compile whereas even a beginner skills in Unix make an assignment much easier. Many times I have seen students struggle through a course at the beginning more because they lacked the ability to tell the computer what they wanted it do, and not because they do not understand programming.

I strongly urge anyone considering a computer course to buy a book that teaches the basics of navigating within the computer. If there is any sure way to assist a student in a computer class, it is to give them a strong foundation in methods to use the computer.

To learn more about Unix, look on the world wide web, read books and watch for ACM workshops. LUG InstallFest will touch on basic shell usage as one of its topics. It is scheduled to be on October 16th-17th.

"Just what do you think you're doing, Dave?"

-HAL, 2001: A Space Odyssey

SIGDAVE

by Stephan Behling

SigDave was founded to be an eclectic forum for discussing computer issues, hardware, and software that might not get talked about otherwise. It is meant to empower its members to explore cool things and socialize.

So far this year we've seen the innards of an old Sega Genesis, talked about everything from supercomputers to Game Boy, and even seen the makings of a Fortran compiler one of our members was tinkering with. In the future we'll probably look at stereoscopic display concepts and anything else our members want to explore!

SigDave is informal, relaxed, and you don't even need to be named "Dave" to join! We meet in 1225 DCL, the ACM office, on Tuesdays at 7:00PM, and sometimes branch off to other rooms as needed. Everyone is welcome, regardless of experience or knowledge!

by Jason Gallicchio

The special interest group for computer architecture and digital hardware got off to a great start this year. There were many new and dedicated members along with many returning members with new ideas.

As a first order of business, we had an introductory hardware tutorial since personal hardware experience is much less common than software experience. We discussed the electrical considerations of TTL logic and everyone was eventually convinced that "open switch" was something TOTALLY different from "Logic Zero" despite years of flowing water analogies.

We found a responsible treasurer who will make it his one goal in life to please the hundreds of adoring fans of the Power Glove Serial Interface the group developed years ago. Properly managed, these funds can be used to afford the expensive hardware evaluation and test equipment required to do cool projects.

The gigantic, graphical, web-based, real time, LED display is in the process of being reworked to connect to the bus of our AMD network evaluation board rather than the parallel port of a PC. 16-bit network drivers and IP/UDP stacks were written for it last year as a separate project, and now that these two endeavors are meeting head on, with the assignment of an IP to the board, the LED display suddenly become a true web appliance, accessible to millions with the simple click of the mouse.

ACM@UIUC would like to thank Judy Tolliver and Shirley Finke for all their support and help

Light-years Beyond

by Stephan Behling

One of my all-time favorite video games was released in 1993 amid a tremendous marketing extravaganza. The marketing wasn't focused solely on the game's great action, wonderful sound, and engrossing play, however. "StarFox," home to the first generation "Super FX" microprocessor, drew in audiences with a new generation of 3D game graphics for the Super NES. "Why go to the next level...when you can go light-years beyond!", exclaimed the voice on the TV commercials.

Games with 3D polygon graphics really weren't a new thing, of course, as games (e.g. Race Drivin' and Hard Drivin') had already been marketed. In fact, StarFox itself is said to have been at least partially inspired by another game (programmed by the same company) called "Starglider." The difference was until that time it wasn't practical to get very math-intensive graphics on a traditional character-based display. While the Super NES did sport very nifty special effects for its time, such as its renowned "Mode 7," the 65C816 CPU at its heart didn't even have a built-in multiplication instruction, let alone the sheer prowess to handle the kind of workload a game like StarFox needed.

Earlier games like "Pilotwings" and "Super Mario Kart" resorted to coprocessors mounted on their cartridges, and U.K.-based Argonaut Software (http://www.argonaut.com/) decided it needed to take a similar route, but this time with a processor built for its needs. With Nintendo's blessing the project started in the early '90s.

StarFox and its little-known sister game, "Super StarFox Weekend" (a special 4-minute competition version) were the first games released with the Super FX. A double-speed version powered lesser known games like "Stunt Race FX," "Vortex," "Yoshi's Island," "DOOM," and "Dirt Trax FX. Many other games were planned, but unreleased, such as "StarFox 2," "Commanche," "Powerslide," and "FX Fighter."

Dylan Cuthbert, one of the StarFox programmers, gave an interview earlier in the year at http://www.emulatorium.com/newsup/2.htm He since moved to work for Sony Computer Entertainment while Giles

to be continued 14

SigMobile's Mobile Computing Device Spotlight: TI-89 Graphing Calculators

by Pavan Tumati,

Lately, educational institutions are placing more emphasis on the use of technology in math classrooms to help students gain a greater understanding of the mathematical concepts taught. Texas Instruments graphing calculators are one of the many tools used. Many students on the UIUC campus own and use these devices everyday. Texas Instruments offers various models of calculators targeted at students in different stages of their education. At the high end of TI's calculator line is the TI-89. The TI-89 is designed with the science and engineering university student in mind. It offers many features such as 3-dimensional graphing capabilities, a computer algebra system, and an interactive numeric solver. It can be programmed in a language resembling basic. Furthermore, one of the most enjoyable and entertaining features (especially to those who like to develop software at a low level) of the TI-89 is its ability to be programmed in MC68000 assembly language.

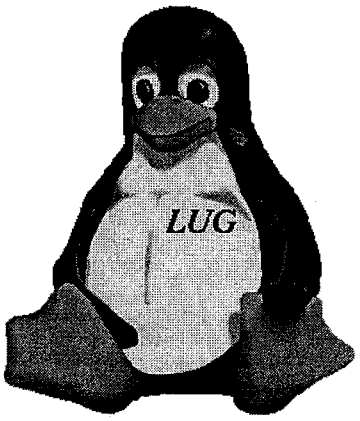
Programming the TI-89 in assembly language allows the knowledgeable code-hacker to take and extend the functionality of this remarkable machine from a mathematical tool to an entertaining, general purpose computing device. The TI-89 sports a 10 MHz MC68000 processor, 188k of RAM, 384k of Flash ROM (usable by TI-BASIC programs), a link port, and a crisp 160x100 pixel display screen — all in a package that fits nicely in an engineering student's pocket. TI-89 users can extend the functionality of their calculators by downloading ROM upgrades offered by TI, and by constructing one of many devices that interface to the link port and perform various functions. Some examples of unofficial upgrades to the TI-89 are:

-An Infrared Link. You can construct a device that interfaces to the link port and control infrared devices from your calculator or communicate with other calculators over a wireless connection. (Information available at **http://sami.ticalc.org/irlink/e_intro.htm**.)

-Speakers. You can build speakers that plug into your calculator's link port and play music. (Information available at: http://www.calc.org/hardware/speakers.html.)

In addition to a nice hardware configuration, TI-89 users receive a much support from the TI calculator

to be continued 13



Linux Myth of the day

* Linux is hard to install! - Linux is extremely easy to install as a machine's primary operating system, and only slightly more difficult to install as a second OS. If you have ever tried installing Win9x as a machine's second OS, you'll find that Linux can be much easier to install- possibly because Linux distributors count on many of their users dual booting between various OS's, something other OS designers certainly don't count one

To discern the Linux truth come to LUG on Tuesdays at 8PM.

LUG is also having an Installfest on Saturday and Sunday, October 16 and 17. Contact lug@uiuc.edu for further details.

SIGBIO

In our last meeting (9/15/99), there was a mini-lecture about biometrics. A good website about biometrics is http://www.ctst.com/biometrics.html. The title of this page is "Everything You Need to Know About Biometrics." We also decided on adding either the Cardio-pulmonary system, the Muscular system, or the Skeletal system to the Virtual Anatomy Textbook(VAT). The next SigBio meeting will be on Wednesday, October 6th at 6:30PM in room 488/494 Burill Hall. Directions: The rooms are right next to each other. The southwest entrance will be open. Take the southwest elevator up to the fourth floor and join us in room 488/494. This is Romesh's lab. He will give us a tour of it, and we might even get to see some NMR technology in action! We will also decide on a layout for the Virtual Anatomy Textbook and decide on what parts each of us will work on.

MacWarriors

MacWarriors is a Special Interest Group for people who use and develop for Macintosh and related technologies, offering tutorial workshops, engaging projects, and a community for Apple enthusiasts to have fun in. It meets every Saturday during the academic year at 3:00PM in the ACM office (1225 Digital Computer Lab). The co-chairs are Rick Roe and Ilya Bakshee. Join the mailing list by mailing MacWarriors-request@acm.uiuc.edu

Don't Forget! AppleScript Tutorial Workshops
Saturday, October 16, 1999
Saturday, October 23, 1999

Women in Computing

Our newest SIG, focusing on women in the computing industry has begun meeting regularly. At the moment this group has both enthusiastic leadership and members but is lacking a solid direction. We are looking for people (both men and women) who are interested in working on trying to address this issue, and get involved with introducing students in the rural areas surrounding Champaign-Urbana to computing.

Currently we are attempting to establish a high school program to interest young women and men in programming, engineering and other areas of computer science

If your are interested in getting involved with this group, email Monica Shaw at mlshaw@uiuc.edu, or Rene Hendrix rhendrix@uiuc.edu or attend one of our meetings on Mondays at 7PM.

SigUNIX, the Special Interest Group for general UNIX interests, has started off the year well; currently, our main project is to develop a scheme on the Linux and FreeBSD platforms that allows one to create files underneath certain directories so that by default, they are created with a default set of permissions.

This is very useful in several contexts. For example, in group projects, several times I've been working with a group maintaining some files in a directory, and my being in the directory's group allows me to create files in the directory. Since the directory has the setgid bit, all files will be created so that their group owner is the group I'm a part of. However, once in a while, people working in the group will have a umask of 077, so the files they create will not allow the group access to the file, thereby possibly stalling a development team. Having a default set of permissions on the files in this directory will eliminate this problem. Another use for this is on cgi-bin directories, where one wants to create all files with executable permissions, or in public_html directories, where files should be created 644 and directories 755.

So far, after much debate and consideration, we believe we have come up with a general overlay of the scheme, and are now proceeding to work on the system level aspects. While our ideas will probably not be integrated into the mainstream, it will give each SigUNIX member each practice in low-level, useful system programming.

SigUNIX meets on at 8:00 in front of the ACM office, after which we generally move to another room, often 1235 DCL.

Why Telnet is evil

by Charles Dowell

If you are like most people, you probably value your privacy greatly. Most people routinely lock their doors and close the blinds on their windows when they want privacy. But for some reason, people do not take similar precautions when they go online. A significant number of the student population use telnet to log into their accounts to check e-mail, most of them not even realizing that their privacy is in jeopardy each time they use it. What these students don't realize is that all the information sent back and forth through telnet is sent as clear text This clear text information can be read by any computer that is on the direct network path between the computer you are using and the computer you are connecting to. All of the e-mail that you are writing and reading through telnet could be monitored or read by someone else. But, what makes this even worse is that everything you type can be read, including your username and password. Every character you type is sent across the network individually and then echoed back to your computer. For someone to grab your password off the network, all they have to do is set up a program called a sniffer that listens to all network traffic. Once they have recorded the network traffic, they can run filters on it to select only the telnet traffic sent back and forth between two computers. Now they just have to combine the individual characters and they have your login and password. After doing this, they not only have access to your personal data, but they can also impersonate you as well.

Now that you know some of the security concerns that arise with using telnet, you are probably wondering what you can do to protect yourself. There are other ways to log into accounts that secure your user name, password, and even data by encrypting it before it is sent across the network. One of the most popular secure alternates for telnet is ssh, or secure shell. Almost all of the University's major servers and workstations support ssh, and some are even being upgraded to support ssh2 (which also has a secure version of ftp titled sftp). Both Linux and Windows have free implementations of ssh, so go download a version for your computer and start protecting your privacy online.

If you found this interesting and want to learn more about computer networking and security, come to a SIGNet meeting Mondays at 7:30. SIGNet, the Special Interest Group for Networking and computer security, has set up a switched network in the netlab (www.netlab.uiuc.edu) and is now working with Cisco router configurations. Anyone who wants to work on a project, or just wants to learn is always welcome to attend.

Have you heard of a little thing called ssh?

by Frank Tobin

First and foremost, ssh allows one to login to easily remote hosts securely, using encryption. When you use telnet, and are prompted to enter your username and password, this information (and everything else you type!) is sent in the 'clear' (unencrypted) to the host. This is an undesirable situation; however, when you use ssh, it sets up a secure channel using strong encryption techniques before you type in your username and password, allowing you to communicate securely. Hence, ssh allows you to send your password and all other information across in a secure manner.

However, in some circumstances, even sending your password across encrypted is undesirable. Sometimes you don't even trust the system with your password. "What do you mean?", you ask. Let's a group of machines that all 'work together'; that is, your password works on each of the systems. The CS department's SUN systems is a good example of this. None of those systems have your password; they ask you for your password when you connect via telnet or ssh, and then query a master server to check your authentication. Therefore, if the remote host you are logging into is compromised, the attacker can modify the ssh daemon to spit out the password you type to it, and then it's "game over"; your password has been given away. Using RSA-key based logins prevents this. I'm not going to go into details, but general overview is that the system, when you connect via ssh, presents you with a challenge, and using a public-key based system, authenticates you for that session. You never send over any secret, as you did when you used your password; your password was the 'secret'. This prevents the person who compromised the machine from getting any useful information then he already had. For more details of this, consult the manpages for ssh and ssh-keygen.

ssh also acts as a replacement for non-anonymous ftp. Using scp, one can securely transfer files to

to be continued on page 14

Reconfigurable Computing

by Jeffrey Cook

Reconfigurable computing is an upcoming technology that uses reprogrammable logic technology for computation. Instead of utilizing full software solutions or full hardware solutions in computational problems, reconfigurable computing (RC) allows for a mutable hardware solution. In essence, reconfigurable computing technology is the ability to modify a computer system's hardware architecture in real time. Configuration is done post-fabrication, and provides a form of General-Purpose Custom Hardware which gives a speedup over software only solutions and allows changes to the hardware after fabrication.

The main impetus for the use of RCs is the large speedup realized versus software only computational algorithms. In addition, RCs allow for changes in the processing to occur after the initial design phase by simply reconfiguring the hardware. This allows for lower design costs, faster design time, and the modification of a product if requirements change, without the need to replace the product. One example of reconfigurable hardware in a current commercial product is the inclusion of a Field Programmable Gate Array (FPGA) in high end Cisco routers. This approach allows internal hardware to be changed once a product is shipped and still has the speed associated with hardware only implementations. The use of reprogrammable logic as a computational medium in RCs versus simple glue logic, as it has been used in The past, has been aided greatly by the advent of larger, faster, more complex reprogrammable logic devices.

ARCHITECTURE

In examining reconfigurable systems, it is necessary to examine the hardware architectures that will be used. The main forms of reconfigurable programmable logic that are currently available are SPLDs, CPLDs and FPGAs.

Simple Programmable Logic Devices (SPLDs) are also known as PLAs (Programmable Logic Arrays), PALs (Programmable Array Logic), GALs (Generic Array Logic), and PLDs (Programmable Logic Devices). They are the smallest and consequently least expensive form of programmable logic. A typical SPLD is comprised of a small number of macrocells (4-16), which can typically be used to replace several 7400-series TTL devices. A macrocell contains a sum-of-products combinatorial logic function and an optional flip-flop. The combinatorial logic function typically supports four to sixteen product terms with wide fan-in. A macrocell may have many inputs, but the complexity of the logic function is limited. Contrast this to most FPGA logic blocks where the complexity is unlimited, but the logic function has just four inputs. Each of the macrocells is usually fully connected to the others in the SPLD. For reconfigurable computing, SPLDs contain far too few macrocells to be useful.

Complex Programmable Logic Devices (CPLDs) are similar to SPLDs except that they have a significantly higher capacity. A CPLD typically contains 18 to 256 macrocells. A group of 8-16 macrocells is typically grouped together into a larger block called a logic block. The macrocells within a logic block are typically fully

to be continued on page 16

SigBiz

Technology Invents Business VS. Business Invents Technology.

by Nathan Lau

Everyday, we see the inventions of people. The light bulb by Thomas Edison, the telephone by Alexander Graham Bell, the computer by Von Neumann, or the printing press by Gutenburg. However, we are beginning to see the evolution of 'invention'. Creativity and invention have become more of a collaborative process instead of breakthroughs of single men and women. Replacing single men and women are the businesses that you see today. With more and more demanding market areas of the world, computing and digital areas of industry have nearly taken over the processes of invention through the use of computers and technology. Without them, the very essences of the twentieth century would not exist in the same fashion as they do today. With these essences, large name businesses tend to show up as creators and inventors. There can be nothing more exciting than forming a new business and watching it grow and build according to the constraints that one has given it. SigBiz intends to help form an alliance for people who are interested in keeping their visions alive, seeding the new inventions of tomorrow. Forming new business units under the alias "entrepreneurship" can be a very exciting thing, and in itself, an invention of process, an invention of doing new things, a different way at looking into concepts, ideas, and thought. There is not a better place to share ideas and key concepts than at a SigBiz meeting. Aiming to perform as an intellectual search engine, SigBiz wishes to create a better way of forming innovative ways of doing business. In this, we see the invention of the invention, so that maybe someday, we all will be inventors, of business.

Surviving the Gen-Eds

by Ashley Wise

You're an engineer from birth, a scientist at heart. You create your own operating system, invent hardware interfaces, and tackle any MP in hours. Your have the mind of a genius. And then it hits you. "In 8-10 pages examine the role of women in first century Roman culture." Where is there time to write when you have MPs? Where is the time to read 300 pages a week when you have labs to do? You have been struck down by a general education class.

There are two reasons why the college elite get weeded out by gen-eds. The first is the strict requirements. We all understand the importance of a well-rounded education, but we should at least be given the opportunity to pick the various gen-ed topics that actually interest us. Instead we are given a list of strict requirements by both the college and the campus -- both of which have to be met -- which includes a short list of courses which fulfill those requirements. And furthermore, the courses that satisfy the college requirements do not always satisfy the campus requirements. Some people might have more interest in a certain culture or social science, but the strict requirements say we must have a western and a non-western class, and two humanities and two social science courses. We are not allowed to pick classes that interest us, or take a sequence of classes that interests us, because only at most two classes in any one subject will qualify. Due to the number of required core classes we have for our major, we do not have enough free credit hours to take any extra gen-ed classes that do not fulfill the requirements. Moreover, a lot of interesting classes satisfy no requirements, or they all satisfy the same requirement so you can only take one of them.

So how do we pick our gen-ed classes when none of the ones that interest us count towards graduation? Well, hopefully someday the colleges will see the light and loosen the strict gen-ed requirements. But until then, your only hope is to get a minor. For instance, I love Japanese culture and wanted to learn Japanese. Unfortunately, the College of Engineering does not count language courses toward graduation, and I have no credit hours to spare. So I got an international minor in Japanese, and with this the college allowed me to count some Japanese classes towards gen-eds. They still did not count enough that I could take as many Japanese classes as I wanted, but at least I was able to take two of them.

No matter how lucky you are, you will probably have to take at least two gen-ed classes that are the least worst of your alternatives. If you pick 100-level courses they will generally be easier to the non-LAS major than high level courses. You should ask your friends who have taken the courses or the department to find out which has the least strenuous workload. You will be assured to have at least one big paper in each class, and/or several small papers.

So how do you survive these classes? Well the first way is going to lecture. Unlike engineering and science courses where lecture notes (sometimes better than the actual lecture) are on-line, gen-ed courses generally have no lecture notes. The companies around campus that attend lectures and sell

continued on page 15

S I G B o t

by Doug Armstrong

Brace yourselves for what SigBot is cooking up this year. We will have an unadulterated display of technology as our two-legged robot walks itself out autonomously. At an ambitious scale of over three feet to the waist this extreme machine will turn more heads than just it's own. A project of this size will cost an interesting amount as well, which we hope will be made possible with the support of corporate sponsors. Now is your chance to make history in the field of robotics. We already have part of the legs built and have started working with the control electronics and feedback systems. Don't have experience with robotics? Well, it's such a cross-disciplinary arena, that it doesn't matter. Just awe at the strength of this beast. Ankles that can hold 6 tons and lift 960 pounds. Come, and be a part of it all. We have official meetings every Tuesday at 8PM where we usually meet at the ACM office (1225 DCL) and usually mosey on down to our lab/workshop in L510 DCL. Plus we make lots of cool field trips to hardware stores and the Everitt machine shop.

WinDevils

- Who We Are

We are the Association for Computing Machinery's Special Interest Group for Windows Development and Programming. We work on anything that has to do with the Evil Empire.

- Our Purpose

No college classes teach the special programming skills that employers want. This includes Win32 programming, the Component Object Model, DirextX, and other important APIs and methods that are widely used in industry today. WinDevils fills this gap by offering hands on workshops to teach today's computer oriented students the skill to help them obtain more rewarding, higher paying competitive jobs.

- What we do

Why would a college student want to go to more lectures? You're right, they wouldn't, and thanks to WinDevils they won't be. We've thrown away the fall asleep while we lecture you format. This year, all of our workshops are hands on labs. You sit at the computer the entire time -- programming, experimenting, and learning.

For More information mail Ash Wise (awise@uiuc.edu) or attend a WinDevils meetings on Wednesdays 8:00PM.

Mechmania V

Over one hundred years ago the planet earth was embroiled in its most dismal war. No one yet living remembers its exact cause, but no one was too young to be spared its merciless horror as rival terran and colonial clans waged bitter conflict throughout the solar system.

Most humans learned to live new lives in hidden bunkers--timid rabbits fearful of slaughter by the most ruthless of the elite warrior class. The only humans who lived in security and comfort were the employees of weapons manufacturers who readily supplied all clans with advanced armaments. Most notable was the line of "MECH" systems, whose sophisticated artificial intelligences aided their pilots and allowed for very complex warfare tactics.

The rest of the galaxy, which once regarded our solar system as relatively peaceful by its own standards, now referred to it in new terms. Loosely translated they meant "Place of the Bloody Metal" or "Place of the Soulless Conflict."

To the weapons contractors' programmers, however, the war had a different name. Since the companies were too valuable, the programmers, freed from fear of destruction, could watch the battles with an evil joy. They didn't care about the human aspect of war. They instead competed with themselves to develop the best MECH artificial intelligences.

To them, the war was called "MechMania."

With the end of the clans' war the battle-scarred MECH systems were locked away in vaults with the naive hope they'd never see action again. Out of sight, out of mind. Humanity prospered in peace. One day a salvage team from the financially desperate Nimbus Terraform Corporation stumbled across a forgotten vault and, despite interplanetary treaties outlawing intrusion, sent an experienced survey crew to investigate. As the survey crew entered the musty, forlorn bunker it marvelled at its size. Damaged spacefighters, plasma-scarred pulse laser cannons, and giant quantum generators lined corridor after corridor. It was quite a find, yet although the team knew no one had set foot in the vault for decades it couldn't help but feel the presence of something more menacing eyeing its movements. Accidentally triggering a proximity sensor, the team powered up backup vault power systems.

The original MECH programmers accomplished many miracles in their designs. Genetic algorithms, trillions of synthetic neurons, and hyper-advanced heuristics harmonized in wonderful elegance. MECHs could function with near complete autonomy back in the war. What no one bargained for was the MECHs' ability to learn--even without human intervention! Even under reserve power the MECH systems' sythetic brains grew in capacity; their only goals: continued war and revenge.

The survey crew was never heard from again. Instead the salvage team radioed frightening video footage back to its parent company. Battle-hardened, century-old MECH systems tore apart their titanium-reinforced concrete graves. Intelligent or no, the look of determination and hatred in their optics was unmistakable. Locating other vaults across the solar system, the MECHs revived allies and fought rival MECH systems with the weapons their original human pilots had unknowingly taught them to use. Only the strongest survived for the final conflict.

Thus MechMania began once again!

MechMania 5: Vengeance of the Slain

Programming contest teams will write C++ code to control their mechs in the scenario which will be given out on Friday, Oct 8. After about 17 hours of lab time, the teams' mechs will compete against each other.

Sponsored by Trilogy

MP3 Digital Audio & the Internet

by Albert Ko

As one of the fastest growing forms of media today, the Internet is giving the general public easy access to an incredibly powerful resource. One of the fastest growing uses of the Internet is to access digital multimedia and audio. In fact, "mp3" is one of the most often searched entry in search engines today. There are a variety of standards and methods that the computing industry is using to take advantage of this growing interest in audio and multimedia. So lets get started on looking at what all the options are right now.

Well, to start off, what is an MP3? Essentially, it is music in a compressed digital format. MP3 is actually MPEG Layer 3 audio, a sort of cousin to the digital format that is used on DVDs. The MP3 encoding scheme was developed mainly by Fraunhofer-Gesellschaft - IIS (http://www.iis.fhg.de/amm/).

So how does it sound? Pretty good depending on a variety of factors. Many people use encoding software on their computers to make MP3s from regular audio CDs. Those factors include the level of compression: 128 kilobits per second is about CD quality for most MP3s but you can even do 256 kilobits per second encoding to get near studio quality sound, pretty much an exact duplicate of your CD... masked sounds and all. Xing Technology (http://www.xingtech.com) currently has an encoder that does variable bitrate compression. Essentially, it changes

to be continued on page 15

Open Source Licenses

by Mike Perry

What is Open Source?

Open source is a relatively new term created by an organization spearheaded by Eric S. Raymond called the Open Source Initiative (OSI) to clear some of the ambiguity of the informal term "free software".

The concept of free software, however, is not new. Since the earliest days of UNIX, most software development was done by Universities and research labs that released their source code for other programmers and institutions to use, modify, and improve.

All right, if free software has existed for years, then why the sudden name change? Well, with commercial software companies distributing shareware and free binaries (and even limited distribution source), it has become important to distinguish between software for which there is no monetary exchange, and software that is truly free: open source.

The Open Source Definition

To this end, the OSI has published a standard, called the Open Source Definition (http://www.opensource.org/osd.html), that licenses must meet if they are to be OSI Certified and programs licensed under them are to be considered open source. The main criteria include free redistribution, complete access to source code, and the right to create derived works (or patches to the original work).

In this article, we will discuss only those licenses that have been OSI Certified.

Licenses Available

The popularity of open source has caused scores of different licenses to spring up, along with very heated debates regarding which license is more favorable. With the myriad of licenses available to the open source programmer, it is important for him to be knowledgeable about the different nuances of each one.

Basically, licenses can be divided into two classes: 1. Those that ensure that derivative work remains open source, and 2. Those that allow derivative works to become proprietary. The differences within the classes arise in how each license chooses to define derivative work.

Protecting Derivative Works (aka: Copylefting)

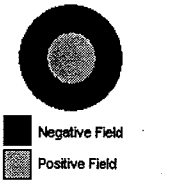
The first group of licenses consists of those that

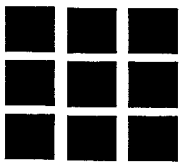
to be continued on page 16

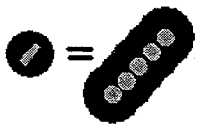
A little rant about eyes

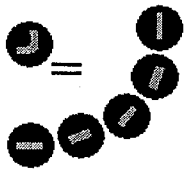
by Borislav Dzodzo

Until I had to program software that would recognize objects, I never really gave my visual cortex much thought. Simply thinking about the vastness of the problem has driven me to madness and tears. A broken man, I turned to recent advances in neurobiology for a guiding light. As it turns out our retinas already modify the image that you receive. Ganglion cells enhance the signal from an area if there is a bit of a difference within that area. In other words, something like a mask is applied. The mask is going to provide a highest result if the center pixel is high and surrounding are low, medium if it's all one color, and very negative if the center is low and surroundings high. This effectively enhances the pixels that are at the edge. So by now you probably think that I am lying. Take a look at it yourself, look at the picture with black squares. You should note that the centers of white crossings seem darker. If not, don't worry there is just something wrong with your eyes. This is because the white lines are close to the edge everywhere but at the center. Now in your visual cortex there are neurons that detect active masks aligned in a linear pattern. These neurons represent local lines. Then there are neurons that become active which local lines are in particular arrangement. These neurons mostly detect curvature of lines. Beyond this point, I found the whole field of neuroscience to be of no use. Eagerly I turn to ongoing research only to find the same ideas reiterating themselves. Most image recognition software scans for locally consistent in all locations and at all angles. Local patterns include colors, corners, blobs, lines, edges and curvature. Then this information is fed to a trainable agent (such as neural nets) and is trained to its maximum. Our group SigArt intends to extract 3D information about the environment and use this as another set of features that would enhance recognition. We meet on Wednesdays at 7:00PM in the ACM office, you can contact us at sigart@acm.uiuc.edu.









From the Secretary's Desk

by Nick Michels

In the bustle of the largest computing conferences, we forget that many company representatives, and visitors are not aware of what ACM is and does.

UIUC's student chapter of the Association for Computer Machinery (ACM) is a professional organization dedicated to the study of computing technology and its uses. With over 300 members the ACM is one of the largest computing organizations at the University of Illinois. The majority of ACM members are students who seek greater knowledge through group projects and challenges unlike those offered in academic work.

ACM about 20 Special Interest Groups, or more commonly known as SIGs each devoted to more specific areas of the computing. These range on topics from architecture to web development, and cover almost anything between. Many of these groups meet weekly to discuss current member interests and work on various projects.

ACM in general is focused on providing a very productive environment with the resources needed to prepare members for success in the computing industry. Whether its problems with academic work, projects, or general curiosity; ACM provides a large knowledge base of class work, various texts, computer hardware, and resume services.

Like any student based organization, ACM is not 'all work, and no play'. The Association for Computer Machinery devotes almost as much time to social activity as it does to working. Events include picnics, movie nights, Frisbee, game tournaments, and activities.

Overall, The ACM@UIUC is a large organization focused on promoting the use of technology, and the advancement of industrial standards. Each of its members are urged to work together and challenge their knowledge.

Book Review

by Vik Kulkarni

Title: Programming the Be Operating System
Author: Dan Parks Sydow
Publisher: O'Reilly

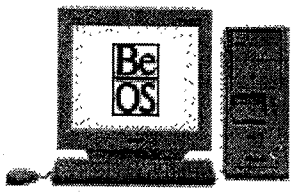
Okay... You've been hearing about how cool the BeOS is for a while now... You've heard all sorts of things about it's API; about how easy it is to write programs for. And now you want to sit down and hack out some code. But where do you begin? Reading the header files is no fun, and the sample code is a bit confusing... The BeBook is a bit dry, and a bit light on the explanations... So where do you turn for help?

Your local Be Users Group, that's where (http://www.be.com/world/usergroups/ and http://www.acm.uiuc.edu/bug/). But users groups don't meet every day, so where do you turn when your fellow BUG'ers aren't around? You might want to turn towards "Programming the Be Operating System" written by Dan Parks Sydow and published by O'Reilly.

The book provides a good introduction to most aspects of the BeOS API. But its focus is on the InterfaceKit and a few other Be classes that a beginner is likely to deal with.

The book starts out with a quick overview of how the Be API is organized (into 13 Kits). It then gives you a detailed tour of the BeIDE (Integrated Development Environment) and all the development tool you'll need to write all your BeOS apps. And then the fun begins!

Sydow starts out with the HelloWorld example code (provided with the BeOS in /boot/optional/sample-code/intro/HelloWorld) and guides you through modifying it until you've written a simple text editor. Along the way, you learn about how the basics of the Be API, and you'll be ready to start that wonderful project you've had brewing in your head.



by Matt Wronkiewicz

This month in the Be Users Group we will be continuing to hold workshops to introduce people to programming the BeOS. During the first month of meetings the BUG covered the basics of BeOS programming, along with some advanced concepts such as messaging and multithreading. Attendance at the workshops has been good, and we hope to see more people discovering the simplicity and fun in writing software on the Media OS.

This week in the BUG we are writing a game called Othello for programming practice. Othello is an ancient game that we can easily implement in the BeOS using message passing and multiple GUI views. The week after that we will hold a screen-saver writing competition just for the heck of it. Later, Jason Luther will hold workshops on the Media Kit in preparation for work onnext year's Engineering Open House project.

The Be Users Group meets Wednesdays at 7:00 at the ACM office. For more information about BUG events and the BeOS, contact Ryan Ott at ryanott@uiuc.edu.

ACM@UIUC would like to congratulate Professor Reingold on the birth of his two grandchildren

SigGraph

by Josh Michaels

SigGraph is back! Yes, SigGraph - near defunkt - is now back and powerful as ever with some great projects on the horizon. You may say to yourself, "Hah.. yeah, that's what they say every year." We do honestly have a lot of work going on some very interesting projects.

First and foremost - get ready for the return of Sounds and Visions. Not just one, but two! We will be holding a winter S&V this year with primarily rendered pieces. SigGraph and SigMusic have already hooked up and started working together on this. Unlike past years in which the "collaboration" of the sigs was simply "we make music, you make graphics," this year SigGraph members have been partnering up with SigMusic members to work together on both music and graphics. S&V is something everyone should be looking forward to this winter.

Second - our amazing project. The goal - make an application that renders worlds not in the traditional sense, but as artwork. Starting off with models provided by past research in this area, we are attempting to make a real time rendering engine that will "paint" worlds instead of render them. SigGraph members are right now working on the structure and method for the program as well as doing research into the styles of various artists. In the end you will be able to select the artist of your choice and walk through a variety of pre-created environments or render your own. This will be quite a nifty project once complete.

Finally we will be holding several workshops over the next few weeks. We at publication time are having a workshop on 3D rendering, and will be having more rendering and graphics programming (OpenGL) workshops in the weeks to come.

Well that's it for this month. I will have a nice synopsis of our project and design structure for next months Banks, so keep your eyes pealed.

SIGMusic FAQ

What is SigMusic?

Special Interest Group (SIG) for Electronic and computer Music. SigMusic is a student group devoted to incresasing the creation, performance, and discussion of electronic music.

How do you become a member of SigMusic?

You just show up to our weekly meetings, Wednesday 7pm. We use the ACM office (1225 Digital Computer Laboratory, corner of Springfield & Mathews) as a meeting place, and sometimes move to a more convenient location.

Where and when does SigMusic meet?

Wednesdays at 7pm.

How do I get on the SigMusic mailing list?

E-mail sigmusic-request@acm.uiuc.edu with the word subscribe in your subject line.

Kernel Architectures

by Sid Cammeresi

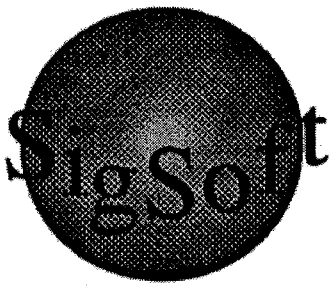
The Special Interest Group for Operating Systems is the ACM's SIG that deals with anything and everything related to operating systems. We talk about existing operating systems, discuss operating systems concepts, but perhaps most importantly, we help our members write their own operating systems. If you want to learn about operating systems, there is no better way to learn than to write your own. If you are interested in doing this, SigOps meets on Tuesday evenings at 7pm CDT at the ACM office.

When you are writing your own operating system, however, one decision you must make very early on in your effort relates to the architecture of your operating system's kernel, its central piece of code that handle talking directly to the hardware in the computer, scheduling tasks to run on the processor, managing memory, and the like. There are two main branches down which you can descend: the path of the monolithic kernel or the path of the microkernel.

Most traditional operating systems such as UNIX have a monolithic structure. In such an operating systems, large chunks of functionality are present in the kernel including things like device drivers and network stacks. All of this code adds up, and the size of such a kernel usually ranges from one megabyte up to seven or eight on some systems. While this might not seem like that big of a deal, the memory used by the kernel most often cannot be paged out to the disk when the amount of available physical memory runs low. A system which only has 32MB of memory may notice a significant slowdown when it runs such kernel several megabytes in size. There is an advantage to this approach though, and that is speed. Monolithic kernels are usually easier to make run fast, but also at the cost of keeping all of the pieces of them from interfering with each other, resulting in a crash.

Some systems have used and are beginning to use a new architecture called the microkernel. As you might guess, microkernels are designed to be much smaller than their monolithic counterparts by moving code that takes up the most space like device drivers and protocol stacks outside of the kernel to run as user space programs.The degree to which this occurs varies greatly depending on the operating system. Systems like Mach

to be ontinued dunno yet



SIGSoft is a Special Interest Group of ACM@UIUC that focuses on issues concerning all aspects of software design and development.

SIGSoft's objective is to improve our competency in software development by stimulating interaction among individuals with an interest in software engineering. This broad scope allows engagement in projects and discussions which delve into divergent disciplines of computer science.

SIGSoft@UIUC seeks to create an environment that is beneficial to members of all ability levels and skill sets.

SIGSoft's current project is a networked, real-time strategy game.

Meetings are held Wednesday at 6:00PM. in 1225 Digital Computing

---

SigVR stands for the Special Interest Group in Virtual Reality which is a group of students interested in graphics, art, hardware, and software as it relates to virtual reality at the University of Illinois at Urbana-Champaign chapter of the Association for Computing Machinery. Right now, we have decided to devote all our time to VRML and related projects. Our current project is a multiplayer maze game.

We meet every Monday at the ACM office at 7:00PM, and then usually move to 1330DCL. If you are interested in SigVR find out more information by mailing E-mail sigvr@uiuc.edu to be put on the mailing list

ACM@UIUC's fifth annual Reflections|Projections cont'd from page 1

by Larry Tesler, of Stagecast Software. In previous jobs at Apple Computer and Xerox's Palo Alto Research Center, he helped to lay the foundation of the modern Graphical User Interface. He also contributed to a number of very cool projects ranging from Quicktime to the OpenDoc object model. At Stagecast, he has been working an Stagecast Creator, which allows for programming and education for young children. He will be talking on the subject of novice programming.

After Tesler's keynote, there shall be even more talks, given by Brian Totty, Mike Abrash, Rick Mercier, Eric Allman, Kevin Powell, and Guido van Rossum.

Sunday will also be a day that will be full of interesting events. In the morning Guido van Rossum will be giving a 2-hour workshop on his Python programming language. This workshop will be aimed at students with a basic concept of programming in at least one programming language, but who are not familiar with Python. Following Mr. van Rossum's workshop, there will be a panel on the subject of privacy. In the afternoon, there will a workshop on Win32 Programming.

Another element of Sunday's itinerary, is the Mechmania programming contest. Mechmania will be running all day. An Open Season will take place in the morning, and the main event in the afternoon.

Before concluding this article, I'd like to take a moment to thank all the people who helped make this year's conference a reality. First I'd like to tank all our corporate sponsors, and all the speakers who volunteered to give talks at the conference. I'd also like to thank the Department of Computer Science which has always been a friend of ACM, and has helped a great deal with this year's conference. I'd also like to thank the conference staff, without whom none of this would be possible. Finally, I'd like to thank all of you for attending this year's conference.

Enjoy this year's Reflections | Projections conference.

TI-89 Graphing Calculators cont'd from page 4

community on the Internet. Many sites, such as www.ticalc.org and www.calc.org, offer a wide variety of software programs ranging from mathematical and engineering tools, to personal organizers, and even to popular video games. Some very impressive clones of popular video games, such as the hit "The Legend of Zelda" and "Street Fighter" have been developed for the TI-89. (Screenshots of these programs are available at http://www.calc.org/programs/index.cgi.)

In summary, the TI-89 is an excellent tool for helping a university student understand math and learn about assembly language programming with the MC68000 processor. It is a small, flexible, and powerful device that is just another example of how mobile computing can help educate and entertain us.

Kernel Architechtures cont'd from page 12

(used in Mac OS X and GNU/HURD) and BeOS have many of these subsystems in their kernels even though they call themselves microkernels. On the other hand, the operating system on which I'm working, OpenBLT, has none of these things in the kernel. In OpenBLT, things like the filesystem drivers and the IDE hard disk driver are implemented entirely in user space programs and get run from a script at boot time as would any other program. The main advantage of this approach is that when you are writing the various components, if the system is designed correctly, they cannot interfere with each other. If a crucial subsystem crashes, the computer might not be in a usable state, but at least the problem can be debugged from the kernel.

There are variations on these approaches such as MIT's exokernel, but they are beyond the scope of this article. If you are interested in learning about such concepts, SigOps encourages you to attend its meetings. Come, and learn about other such fun decisions that go into writing your own operating system.

Light-Years Beyond

continued from page 4

Goddard, another StarFox programmer, works for Nintendo. (I don't know where the third programmer, Krister Wombell, ended up, but if he worked for Sega I'm sure there would be some irony there.)

However, I've been equally interested in finding trivia about the Super FX coprocessor itself, so I began my never-ending web search for it.

The search began with Argonaut itself, since it was involved with most of the dirty work. It turns out that Argonaut was so pleased with the chip's success that it spun off a subsidiary to produce more processors. Thus "Argonaut RISC Cores" (ARC) was founded. (http://www.arccores.com) Richard Clucas (who can be seen at http://www.arccores.com/corp/index.html) worked on the Super FX and is listed in the credits to StarFox. He still works for Argonaut as technical director, and according to an email I got from him "there is no actual similarity between the SuperFX and the ARC, except they are both RISC chips." He went on to say how the ARC was designed in VHDL (which, like Verilog, is a popular hardware design language) while the Super FX wasn't, and that the ARC allows for point-and-click customization.

Having just started ECE 312, which uses VHDL extensively in its MPs, that sounded very odd, but my quest continued and finally I was able to find the ACTUAL hardware guru behind the Super FX. By sheer accident I stumbled across the web site of a Mr. Ben Cheese, and realizing he was also listed in StarFox's credits I quickly emailed him. U.K.-based Ben Cheese Electronic Design (http://www.bencheese.demon.co.uk) was involved in quite a few projects. His site claims that there are more Super FX chips than any other RISC processor in the world—a claim I wouldn't doubt for the gaming industry at that time, though I'm sure there are more MIPS-based consoles (e.g. Playsation and Nintendo64) now. Mr. Cheese explained to me how Argonaut had the ideas for the chip's instruction set (i.e. what it could do, functionally), and having seen some of his earlier work they invited him to design the Super FX (or actually the "Mario Chip" as it was called at the time).

Regarding hardware design languages, he confirmed that the Super FX didn't employ any: "In fact it was designed using schematic capture (this was a real retrograde step for us, but it at the time VHDL tools were just too expensive)."

The same split of responsibility between Argonaut and Ben Cheese also gave birth to the ARC itself, and the "point-and-click" RISC design Mr. Clucas mentioned really referred to customizable gaps in the ARC's instruction set.

The Ben Cheese web site confirms the rumors I had heard before about Sharp manufacturing the Super FX, and evidently FPGAs from Actel (http://www.actel.com) were used in its prototyping.

My quest for video game knowledge continues, of course, but there's nothing quite like getting in contact with the hardware and software champions behind my favorite games!

Have you heard of a little thing called SSH?

cont'd from page 6

another machine, as ftp has the same problems of sending cleartext information across as telnet. scp can also use the RSA-key-based login system. The only problem with scp is that is is more like rcp than ftp; it doesn't have interactive sessions. However, some, such as myself, really like this fashion of transferring files. Versions of ssh2 come with a tool sftp, which simulates a normal ftp session, so ftp users feel at home.

ssh does even more! Most UNIX users are familiar with the ability of X to remotely transmit the screen output of graphical programs. This is the scheme that many people use to do their MP's (CS programming problems) from home instead of trudging all the way to DCL. X also has the same problems as telnet; it sends all the information across in the clear.

Once again, this can be avoided. When one logs into a remote machine using ssh, any X application started the remote machine automatically gets transmitted back to the local host, securely! No work needs to be done! No more "setenv DISPLAY ...". It's automatic and secure.

Not only does it allow the system you are connecting to to authenticate you securely, it also allows you to authenticate the system as well if this isn't the first time you have logged into it. A good analogy of this is you telephoning someone, but the call gets redirected so that someone at the other end intentionally 'fakes' being the true recipient, and tricks you into sending him your password, because you think you are talking with someone else. ssh, using its public-key system, automatically prevents this, and warns you if it thinks the remote host has changed.

ssh has even more features than I've explained here; I encourage you to read the manpage for ssh thoroughly; you might be surprised at howpowerful, flexible, and easy to use ssh is. I hope that you can now agree with me, that ssh rocks!

Windows users can find an installation guide for TeraTerm, a popular telnet program, with the ssh plugin http://helpdesk.clarkson.edu/teraterm.htm. This ssh implementation for Windows is very nice, as it has the nicer features of ssh, such as X11-forwarding, and the ability to use RSA keys. Macintosh users can get get an ssh implementation at http://www.lysator.liu.se/~jonasw/freeware/niftyssh/. UNIX users can get ssh from ftp://ftp.replay.com/pub/replay/crypto/SSH/.

Surviving the Gen-Eds

continued from page eight

the notes usually take very poor notes compared to what an actual student would take, so I do not recommend living off of these. Therefore the only way to get the material is at lecture. Also, the readings for the class may be immense, often amounting to over a hundred pages a week. If it took two months to read, how are you expected to study it all again for the midterm? The answer is once again the lecture. What the professor talks about in lecture is what he deems important, and is what will appear on the exams. One last push for attending the lecture is that this is often the setting for Greek initiation rituals, which may bring amusement from time to time.

This brings us to the second reason why we get weeded out by Gen-Eds: we are uninterested in the material. It is hard to keep your eyes focused on a history book when you could be reading a programming book. It is hard to write a paper about something that you have no feelings for. Examining the effectiveness of the IA-64 architecture would probably be more exciting than examining how the economy is effected by trade. The trick to putting sufficient effort into a paper you do not care about is to turn it into a logical process, and to merge it with man's best friend: the computer. Do research on the internet, or take notes with your laptop. When your organize your ideas, do not make an outline, make a pre-order tree structure. Finally, do not bother with that paper medium. You may not have realized this yet, but most of us can type far faster than we can write (or think which occasionally gets us into trouble). So start right away on the computer when you do the actual writing. It is a lot easy to copy and paste when you do not need scissors and glue.

General education requirements are strict and mean, but they are a fact of college life. So choose your courses wisely and tackle the material systematically, and you should be able to avoid getting weeded out.

MP3 Digital Audio and the Internet

cont'd from page 9

the bitrate (the amount of data for a chunk of the music you're compressing) according how much data it needs to output a decent approximation of the original source. Audiocatalyst is another popular software encoding program and uses Fraunhofer-IIS technology to encode up to 20kHz frequency data. There are also some freeware encoders out there like BladeEnc (http://home.swipnet.se/~w-82625/) that will do the job if you're broke like me.

So what's the point of encoding MP3s to your hard drive you might ask? Well, with MP3s, the compression is good enough to get approximately a minute's worth of CD quality audio for about a megabyte of data. So not only could you conceivably put a big chunk of your CD collection on your hard drive. So what does this have to do with the Internet? Good question! Well, since the speed of access for many people on the Internet is increasing, MP3s have started to appear on the Internet!

People began to share the music that they enjoyed over the Internet through the MP3 format. However, as more and more people did this, the recording industry became concerned over the copyright violations happening with the distribution of MP3s. So they basically tried to shut down MP3 as a music format, but to no avail. MP3s had become so popular that even recording artists began to support the burgeoning format. Some artists, such as Public Enemy, TLC, Peter Gabriel, Tom Petty, have actually started officially releasing singles from their upcoming albums in MP3 format on the Internet. Whether MP3s are beneficial or detrimental to the recording industry is open to debate, but I'd rather not get stuck in that mess.

There are also many free and legal tracks that you can download at various sites on the Internet including MP3.com (http://www.mp3.com) who are sponsoring the upcoming Goo Goo Dolls and Tonic concert at Assembly Hall. Another site of interest is Emusic (http://www.emusic.com). They are one of the growing number of sites that are beginning to sell MP3s for download and they often feature special or new tracks not available in retail channels. You can even search for MP3s through search engines provided at Lycos or multimedia sites such as Scour.net (http://www.scour.net). They can even be found on FTP sites and there are newsgroups on the Internet such as the alt.binaries.sounds.mp3 hierarchy that also have MP3s for download. Personally, I use the multi-site search engine provided by MP3Place (http://www.mp3place.com) to find MP3s that I want.

So now that you've got an ample supply of MP3s on your computer, you want to take them with you on the road? Well, until recently you couldn't unless you brought a laptop with you. But now, Diamond Multimedia (http://www.diamondmm.com/) and Creative Labs (http://www.creativelabs.com) have released portable MP3 players (the Rio and Nomad respectively) that use onboard memory or memory cards to store MP3s that are transfer from your computer. Unshockable and pure digital, these players are battery efficient and very portable. Even Sony is jumping on the bandwagon with their own new Walkman which will use "memory sticks" to store your MP3s. However, Sony plans to use its own format to store the MP3s. So right now, the choices are growing. There are even some plans to produce MP3 players that will read off of data CDs (for those of you who have CD recorders) so you can have nearly 650 minutes of music on a single CD for those marathon road trips. Well, enough yakking... time to listen to some more music!

Reconfigurbale computing

continued from page 7

connected, but the logic blocks of a CPLD are not always necessarily fully connected. The connections between logic blocks is done via a programmable interconnect matrix. Although CPLDs can be run at very high clock speeds, only a limited amount of computation per datum is available before it is too expensive to route between logic blocks, and as such, CPLDs are only marginally useful in reconfigurable computing.

Field Programmable Gate Arrays (FPGAs) differ from CPLDs very distinctly. An FPGA consists of an array of complex logic cells, surrounded by programmable I/O blocks that are connected with a programmable interconnect. A typical FPGA contains from 64 to several thousand logic blocks. Most FPGAs do not provide 100% interconnect between logic blocks, but instead sophisticated software places and routes the logic on the device.

There are two primary classes of FPGA architectures: coarse grained, and fine grained. Coarse-grained architectures consist of fairly large logic blocks, often containing two or more look-up tables and two or more flip- flops. In these architectures, a four-input (typically) look-up table implements the actual logic. A larger logic block usually corresponds to improved performance. In fine-grained architectures, there are a large number of relatively simple logic blocks. The logic block usually contains either a two-input logic function or a 4-to-1 multiplexer and a flip-flop. These devices are good at systolic functions (functions in which data flows from memory in a rhythmic fashion, passing through many processing elements before it returns to memory) and have some benefits for designs created by logic synthesis. This configuration allows for simple processing elements with regular and local connections which takes external inputs and processes them in a predetermined manner in a pipelined fashion.

PARALLELISM

Parallelism is essential to the acceleration provided by reconfigurable computing. Due to the nature of the programmable logic devices used for reconfigurable computing, the available silicon can be configured to make multiple units that execute the same or different yet simple computational operation. The main forms of parallelism used in reconfigurable computing elements are: SIMD, MISD, MIMD, and pipelined.

Single Instruction Multiple Data (SIMD) parallelism is when each instruction that is executed causes the same operation to be performed simultaneously on several different pieces of data within a single computer. A FPGA can be configured as SIMD by configuring all the computational datapaths within a FPGA to use the same lookup table and same instruction input. In addition, each datapath would have a common input but its own output port.

Multiple Instruction Multiple Data (MIMD) parallelism is when several different instructions operate simultaneously on several different data

Open Source Licenses

continued from page 10

ensure all derivative works will remain as open and free as the programmer's original work. This idea of ensuring complete access to a program is considered to be the exact opposite of the traditional copyright (which restricts access), and thus has been dubbed "the copyleft".

The most famous license in this category is the GNU General PublicLicense, or GPL. Written by Richard Stallman, the President of the FreeSoftware Foundation, the GPL has the strictest concept of derivative work out of all the open source licenses. The GPL was written to provide programmers with a means of ensuring that code they write and distribute would remain free.

The meat of the GPL can be found in sections 0 through 3. Section 0 defines the notion of a derivative work: any program containing or based on parts of the original program, including translations to different languages and linking to proprietary object files and libraries. A common source for confusion is program output. Programs such as flex and bison (and gcc) produce human readable source code. If you read Section 0 closely, it is apparent that these outputs are considered derivative works (translations) of the inputs, should the inputs be GPL'ed, and not derivative works of the program that generates them.

Section 1 simply allows for redistribution of the program and derivative works, while Section 2 describes the proper form for modifying the program, and differentiates between a derived work and the mere aggregation of two independent works (the latter of which is not covered by the GPL). Section 3 states that you may provide executable forms of the program or derivative work only if you provide means for obtaining all the source code needed to build the program (excluding code to libraries and utilities that are shipped with the operating system).

The rationale behind this strict definition for derivative work is to ensure that software that programmers intended to be free remains free. People at the Free Software Foundation grew tired of watching others profit by making their free software proprietary, and no longer distributable. Their priority is that software remain free for all to use and modify. This is also the reason why linking to commercial libraries is forbidden. Programmers wanted to ensure that their software couldn't be crippled and rendered unusable to many by being modified to depend on an expensive commercial library.

The GNU Library General Public License, or LGPL, follows the same\ sentiment as the ordinary GPL, with one exception: it allows linking to proprietary programs. The definition of derivative work is altered, and programs that link to the library are explicitly determined distinct works (and thus exempt from LGPL coverage). An additional proviso not found in the GPL

Reconfigurable Computing cont'd from page 16

within a single computer. A FPGA can be configured as MIMD by configuring all the datapaths within a FPGA to use separate instruction inputs. In addition, each datapath would receive its own respective data input and output.

Pipelined parallelism is used when several operations need to be performed on a piece of data in which multiple computational cycles are needed. When this occurs, it is necessary for several different computational units to be used to complete the operation. The instruction propagates through all the computational units, moving to the next computational unit on each computation cycle. As one instruction leaves a computational unit, a new instruction can be entering. Therefore, multiple instructions can be in the process of executing in the device at the same time, each in a different stage of the pipeline. A FPGA can be configured as pipelined by configuring the device with several different computational units in a linear fashion, with results of one computational unit entering the next sequential computational unit during the beginning of each new clock cycle.

PROGRAMMING TECHNIQUES

There are four main ways to program a RC: Hardware Description Languages (HDLs), high level programming language conversion utilities, custom RC-specific computing languages, and custom logic design. All of these techniques try to accomplish the same goal of efficient use of some reprogrammable hardware for a computational goal. Much effort is put into trying to implement an algorithm in a hardware setting. This is a problem shared with both RCs and normal hardware logic design. Another challenge is the design of compilers that can efficiently convert the wanted algorithm into the final technology.One set of languages used to program RCs are HDLs. HDLs describe how a specific piece of hardware behaves. specific piece of hardware behaves. They can also model a logic gate levelimplementationof a circuit. This is in contrast to a high-level computer programming language that abstracts design completely to an algorithmic level, with no regard to the underlying hardware that will be used to implement it in the final product. Also most high-level computer programming languages are sequential versus parallel which causes difficulties in a hardware realization of the algorithm. HDLs provide some benefits over gate level implementations of algorithms which includes a more portable, device independent solution, and a shorter design time. Some examples of HDLs are VHISIC Hardware Description Language(VHDL), Verilog HDL, and Altera HDL. HDLs still require a compilation and mapping step to allow for reconfigurable hardware realization.

BENEFITS

RC has definite benefits over both software only and hardware only solutions to computation. The most valuable benefit over software only solutions is the speedup achieved because the computation is being carried out in hardware. This usually realizes an order of magnitude increase in performance for an average algorithm. A quick time-to-market for product design is an important reason that reconfigurable devices have been chosen for many applications. Instead of having a chip fabricated with all of the research and development cost and time associated with custom ASICs, the use of a reconfigurable device is cheaper and quicker for small productions. This is common to most reconfigurable logic, but there are specific benefits that RCs have over other

continued on page 18

Open Source Licenses cont'd from page 16

libraries. It is recommended in the preamble of the LGPL that libraries for which there are no commercial equivalent be GPL'ed, while libraries that are intending to replace a commercial equivalent be LGPL'ed. This is done to promote the idea of open source software. It is the feeling of the members of the FSF and many others that commercial companies always have the choice (and resources) to re-implement the GPL'ed library on their own, should they be that vehemently opposed to releasing an open source product.

The Q Public License, or QPL, was written by Troll Tech for distribution of the free version of their Qt library. In practice, it is almost identical to the GPL. It forbids linking with commercial programs (to encourage purchase of the otherwise identical Qt Professional edition), and modifications must be made available. In spirit, however, it it much different. Troll Tech is the only party allowed to distribute complete modified versions of the Qt source. All others must distribute modifications as patches against an "official" Troll Tech version. This is much different than the mentality of the GPL, where everyone has the freedom to compose a complete derived work, should they choose.

The Mozilla Public License, or MPL, was written by the Mozilla organization. At first glance, it is similar to the LGPL, but closer inspection shows that derivative works only consist of modifications to existing source files. Proprietary object files may be added to create a larger work, so long as those object files do not contain any of the original code. The MPL also grants programmers limited rights to patented material owned by the licensor for purposes of modifying the source code.

(Almost) Unconditional Redistribution

In contrast with the above licenses, which attempt to ensure that derivative works remain free for everyone, licenses in this class are more concerned with getting as many sheer numbers of people to use the product as possible. To this end, they allow almost unconditional redistribution, in source or binary forms, usually with the only proviso being the retention of a copyright notice.

The most famous license of this class is definitely the BSD License. The BSD license allows for redistribution in source or binary with or without modifications, as long as the license is included or displayed along with a copyright notice. The license also forbids using the name of the organization or contributors to endorse the software. Up until July of this year, the license also required that in all advertisements of derived works, a short blurb be included that states that a product uses code derived from the original program.

The MIT/X license is almost identical to the BSD license, except that it is in paragraph form instead of bulleted form.

IBM also created a public license for their Jikes compiler. The IBM Public License is essentially the same as the BSD and MIT licenses except, much like the MPL, it grants limited rights to IBM patents in order to legally modify the code.

The Artistic License is very similar to the BSD style licenses, except that if you intend on making proprietary modifications, you must rename your modified executables as well as provide

continued on page 18

Reconfigurable computing from page 17

processing techniques. One interesting benefit is the ability to match the word length of the datapath to the data word size being computed on. For instance if the data in a specific application needs 37 bit computations, a large amount of computing resources would be wasted in a conventional computer, but in a RC the data flow can be specialized for n bit operations.

Parallelism can be achieved easily with RCs because the implicit parallelism of hardware. This parallelism can lead to speedup in computational execution if the task is parallelizable. A reconfigurable device also has increase fault tolerance because it can be configured to use only the functional portions of the chip much in the same way that computer hard disks mark bad blocks on the disk. Lastly, RCs allow for more efficient silicon usage because more silicon is dedicated to actual computation than control logic, in comparison to a microprocessor.

DRAWBACKS

While there are these benefits to RC, there are some important drawbacks. One being that if the application is not inherently parallel, the speed up will not be as large. Sequential code does not realize higher performance on parallel architectures. Another architectural problem is the lack of space which precludes floating point designs. This will be less of an issue as fabrication processes shrink and chips become larger. Even more importantly, current RC elements require enormous amounts of time to be reconfigured. If a RC has to change its hardware on the fly, which can require milliseconds, it could loose a significant amount of time just reconfiguring itself. This limitation is being solved by dynamically reconfigurable logic which only needs to reprogram what needs to be changed (similar to a diff of source code), and also through the use of multiple hardware contexts. Multiple hardware contexts, currently being developed by NEC, allow for reprogramming a reconfigurable element in under 5ns, bu only by switching between several preprogrammed configurations.

END

In the near future you may see reconfigurable hardware pop up in your personal computer for speed enhancement. One way of harnessing reconfigurable computing technology inside personal computers is through the use of a co-processing board. This co-processing board would allow for speedier computation of certain functions. Current graphics accelerator technology applies the same idea of off loading processing to specifically designed hardware for graphic processing. The difference is that in a reconfigurable setting, a co-processing board can be used for different functions dependant on the currently executing tasks on the host machine.

One example of research that has been done in this area was a project by Satnam Singh and Robert Slous that used reconfigurable logic to accelerate Adobe Photoshop. A commercial board was used that had a Xilinx XC6200FPGA on it and fit into a PCI slot in a personal computer. This project used the FPGA to implement several different filters. Software was written to allow thehardware FPGA filters be a plug-in in Photoshop. While the theoretical speed up versus a software-only approach was thirty times faster, only an approximate speed up of two was realized because of the limited bandwidth of the PCI bus.

Some more aggressive reconfigurable hardware in personal computers can be seen in approaches that put reconfigurable hardware on the same silicon die as a systems microprocessor. One theoretical implementation of this concept is the Garp processor by Berkely Reconfigurable Architectures, Systems, and Software. This design has a MIPS-II architecture processor core along with a configurable logic array that can communicate with the main CPU, the data cache, or directly to main memory. They estimate that the Garp processor could achieve a speedup of 2 to 24 over a UltraSPARC. The programming interface for the Garp microprocessor is MIPS assembly that allows for a sort of hardware subroutine where the processor clocks the FPGA portion of the chip for a specified number of clocks at which point the computation is done, and results can be read.

Although very few companies are currently producing processors with reconfigurable logic on the same die, the future will indeed show that reconfigurable computing has reserved spot in all general-purpose computers of the future.

Open Source Licenses cont'd from page 17

the original executables. Libraries linked to an Artistic licensed program are not considered part of the program.

The rationale behind these relatively lose licenses is that free software should provide users with complete freedom over the software, including the right to resell the software in any form they chose. People who support this class of licenses tend to feel that licenses from the other class (most notably the GPL) actually limit the freedom of users to do what they wish with the software.

What about money? (or: Die, You Pinko Bastard!)

At this point, the average commercial software advocate is becoming very concerned about where the money is going to come from. The Open Source Definition allows people to charge a "reasonable" distribution fee, usually to cover the cost of media and packaging. However, companies involved in the production of open source software have a myriad of revenue possibilities. Documentation and technical support are among the most obvious. Sendmail, Inc. (the makers of sendmail) and Artifex Software (the makers of ghostscript) release old versions of their software as open source. Cygnus capitalizes on first mover advantage by providing embedded systems companies with an initial port and continual support of the GNU development tools for their architectures. Troll Tech and Kaffe both follow a strategy of releasing separate identical versions of their libraries under both a commercial and an open source license.

But perhaps bigger than all this is the fact that most software written in the world isn't even sold! In many companies, software is just a means to an end. In some cases, it's developed at corporations internally to help them do a particular job. This is how perl got its start. In other cases, companies that aren't in the software business per-se may benefit

Reflections | Projections

would like to thank the following sponsers for their generous donations

Abbott Laboratories

Argonne National Laboratory

Collegehire.com

Confinity, Inc.

Crowe, Chizek and Company LLP

FactSet

Lante

Lucent Technologies

Morgan Stanley Dean Witter, Information Technology

SAP Labs

Sendmail, Inc.

STR

Trilogy Software

We would also like to thank the Department of Computer Science for its continuing support of ACM and the Reflections | Projections conference.

This year's conference would not be where it is today without the help and support of these companies and organizations.

ACM@UIUC would also like to thank the hard work of the conference committee and conference chair Mike Khalili

Open Source Licenses

continued from page 18

indirectly from open source software running better. from source software running better. Companies like SGI, Compaq, and Intel are eager to pay programmers to improve open source operating systems on their respective architectures.

So you see, there are plenty of opportunities to pursue monetary gain while still preserving your karmic alignment.

Conclusions

Open source software can be a wonderful thing. It promotes collaboration, rapid development and bug fixing, and above all, knowledge. When writing a piece of free software, it is important to consider which license you release it under. Picking the wrong license could lead to disastrous consequences regarding the future use of your software. Choosing GPL-style licenses may limit adoption of your software by commercial software companies, but you ensure that your software will be publicly available for all generations of programmers to peruse. Choosing BSD-style licenses may put your work in danger of becoming someone else's property, but they will vallow for largest distribution of your program in any form.

Also remember that as long you are the only copyright holder to the code,vyou may change the licensing scheme at any time, but once other people beginvto contribute modifications under a certain license, you are stuck with that license until you obtain every contributor's permission to change it, or youvremove their contributions.

Hopefully I have provided you with enough material to make an informativevdecision about which license to choose. Now go write some open source software!

References

http://www.opensource.org/licenses/ - A list of OSI Certified licenses

http://www.gnu.org - The FSF homepage

acm

student chapter
university of
illinois urbana-
champaign

membership form

Return or mail this form to:
1225 Digital Computer Lab, MC-258
1304 W. Springfield Ave.
Urbana, IL 61801

name:

campus address:

campus phone:

home address:

home phone:

electronic mail:

curriculum:

University Status

- ☐ freshman
- ☐ sophomore
- ☐ junior
- ☐ senior
- ☐ m.a. / m.s.
- ☐ ph.D.
- ☐ faculty / staff
- ☐ postdoc
- ☐ alumni
- ☐ other

special interest groups

bsdeamons	bsd users group
bug	be users group
lug	linux users group
macwarriors	mac users group
sigarch	architecture
sigart	artificial intelligence
sigbio	biocomputing
sigbiz	enterpreneurship
sigbot	robotics
sigdave	short-term distractions
siggraph	graphics
sigmobile	mobile computing
sigmusic	music
signet	networking and security
sigops	operating systems
sigsoft	software development
sigunix	unix programming
sigvr	virtual reality
sigweb	web development
wic	women in computing
windevils	windows programming

Membership Types

return form with check or money order payable to the ACM at UIUC

- ☐ $40 for eight semesters
- ☐ $22 for four semesters
- ☐ $12 for two semesters

ACM National Member

- ☐ yes — #
- ☐ no
- ☐ currently applying

for internal use

Received Date: by:
Chk # $
Enter Date: by: