

Linux Lunacy V & Perl Whirl '05

Running Concurrently Southwestern Caribbean

OCTOBER 2ND TO 9TH, 2005

Linux Lunacy V and Perl Whirl '05 run concurrently. Attendees can mix and match, choosing courses from both conferences.



SPEAKERS

Nicholas Clark

Scott Collins

Mark Jason Dominus

Andrew Dunstan

brian d foy

Jon "maddog" Hall

Andrew Morton

Ken Pugh

Allison Randal

Randal Schwartz

Doc Searls

Ted Ts'o

Larry Wall

Michael Warfield

DAY	PORT	ARRIVE	DEPART	CONFERENCE SESSIONS
Sunday, Oct 2	Tampa, Florida	—	4:00pm	7:15pm, Bon Voyage Party
Monday, Oct 3	Cruising The Caribbean	—	—	8:30am – 5:00pm
Tuesday, Oct 4	Grand Cayman	7:00am	4:00pm	4:00pm – 7:30pm
Wednesday, Oct 5	Costa Maya, Mexico	10:00am	6:00pm	6:00pm – 7:30pm
Thursday, Oct 6	Cozumel, Mexico	7:00am	6:00pm	6:00pm – 7:30pm
Friday, Oct 7	Belize City, Belize	7:30am	4:30pm	4:30pm – 8:00pm
Saturday, Oct 8	Cruising The Caribbean	—	—	8:30am – 5:00pm
Sunday, Oct 9	Tampa, Florida	8:00am	—	

Perl Whirl '05

Seminars at a Glance

Perl Whirl '05 and Linux Lunacy V are running concurrently. Attendees can mix and match, choosing courses from both conferences.

You may choose any combination of full-, half-, or quarter-day seminars for a total of two-and-one-half (2.5) days' worth of sessions. The conference fee is \$995 and includes all courses, course materials, and the Bon Voyage Cocktail Party.

Introduction to Perl 6 (half day)

Speaker: Allison Randal

This talk is an introduction to Perl 6, the next major version of Perl. The primary focus is on the syntax of Perl 6, but beyond the syntax, we'll briefly look at some of the reasons behind the changes. What makes one feature more desirable than another? What makes it Perl-ish? What makes it DWIM? How on earth does Larry decide? You'll leave with a better understanding of the changes in Perl 6, and the fundamental nature of Perl itself.

Parrot and the Perl 6 Compiler (half day)

Speaker: Allison Randal

Parrot is the virtual machine that will run Perl 6, as well as Python, Ruby, PHP, and a host of other dynamic languages. But more than that, it's a suite of tools for compiler writers. It boasts its own object-oriented assembly language, which is a breath of fresh air to anyone who's worked with XS. It even plays Tetris! This talk is an introduction to Parrot: what it is, how it works, and what you can do with it today. We'll also dig into the implementation of the Perl 6 compiler as a standard compiler module in Parrot.

Regular Expression Mastery (half day)

Speaker: Mark Jason Dominus

Almost everyone has written a regex that failed to match something they wanted it to, or that matched something they thought it shouldn't, and often it can be hard to predict what a regex will do. This class will fix that.

The first section will explore the algorithm that perl uses internally to do regex matching. Understanding this algorithm will allow us to predict whether a regex will match, which of several matches Perl will find, and which regexes will be faster than others. During this discussion we'll pause to discuss practical applications that illustrate features of the algorithm. We'll examine the essential but frequently misunderstood concept of 'greed', and we'll learn why commonly-used regex symbols like '.', '\$', and '^|' might not mean what you thought they did.

In the second section, we'll apply our knowledge of the internals, examining at several common disasters, a few practical parsing applications, and some new features such that would have been hard to understand before. We'll see an example of every regex metacharacter and modifier. We'll finish with a discussion of some of the new optimizations that were added in Perl 5.6, and why you should avoid the '/i' modifier.

- Inside the Regex Engine
 - Regular Expressions are Programs
 - Backtracking
 - Quantifiers
 - Greed
 - Anti-greed
 - Anchors and assertions
 - Backreferences
- Disasters and Optimizations
 - Where machines come from
 - Disaster examples
 - Regex modifiers
 - Tokenizing
 - New optimizations
 - Matching strings with balanced parentheses

Programming with Iterators and Generators

Speaker: Mark Jason Dominus (half day)

Sometimes you'll write a function that takes too long to run because it produces too much useful information. A search function might locate ten thousand matches. A database query might return ten million records. What can you do?

Perl provides a simple and familiar model for dealing with such problems: The filehandle! Instead of reading every file in one giant gulp, we can use filehandles to staunch the flow of information, trickling it into the program in manageable gulps.

In this class, we'll see several important modules, such as File::Find and DBI, which use this approach. We'll go inside these modules and see how to implement filehandle-like data structures and objects ourselves. We'll see how to write functions that suspend themselves and then pick up later where they left off. We'll learn how to take long-running slow functions and convert them into speedy filehandle-like data generator objects.

This fundamental technique is a mainstay of programmers in other languages, but isn't as well-known as it should be. You'll be amazed at how many difficult problems become simple when you unleash the power of filehandles.

- Introduction
 - Filehandles are Iterators
 - Iterators are Objects
 - Common Examples of Iterators: readdir, each, and DBI
- Building iterators and Examples
 - File tree scanning
 - Permutations
 - Genomic Sequence Generator
 - Filehandle Iterators
 - A Flat-File Database
 - Searching Databases Backwards
 - Random Number Generation
- Alternatives to Iterators
- Filters and Transforms: 'map' and 'grep' for iterators
- Recursively-Constructed Iterators
- The Semipredicate Problem
- Alternative Interfaces to Iterators: Tied scalars and handles
- An Extended Example: Web Spiders

Testing Perl (half day)

Speaker: brian foy

Learn how to test your scripts and modules using Perl's test harness, `Test::More`, `Test::Simple`, and the popular `Test` modules available from CPAN. You don't have to be a Perl expert to start testing your creations. Spend a little time with testing and spend a lot less time debugging. Attendees should know how to create and run Perl scripts, and experience using Perl modules is helpful but not necessary. If you want to make your Perl more robust and have more confidence in your work, this course is for you.

Creating Perl Distributions (2 hours)

Speaker: brian foy

Learn how to put together a Perl distribution, including an installer, a test suite, documentation, and all of the other things that go into sharing your creations with the world. Learn how to use `Module::Release` to distribute your Perl creations. Attendees should know how to create and run Perl scripts, and experience creating Perl modules is helpful but not necessary. If you want to share your Perl code with the world, this course is for you.

Introduction to Mod_perl 2.0 (half day)

Speaker: Randal Schwartz

Learn about the heavy-lifting `mod_perl` interface for Perl embedded in the popular Apache web server in this introduction. We'll go over the API, including sample code, and installation instructions. A light introduction to the emerging `mod_perl 2.0` for Apache2 will also be included.

Developing Web Applications Using

CGI::Prototype (half day)

Speaker: Randal Schwartz

Randal presents an introduction to what Captain Neil calls "the most significant web application module since `CGI.pm`", the recently released `CGI::Prototype` module.

Learn how every CGI application can be reduced to "figure out what state we're in ... what got sent ... what to respond", and how to represent that in this flexible framework. The framework maximizes code re-use through easy creation of data accessors and methods in a hierarchical class structure, including trivial one-off lightweight object classes for unusual situations or exceptions. The framework also leverages the industrial-strength `Template Toolkit` for generating output.

Full and robust "MVC" applications can be created using `CGI::Prototype` for the Controller, `Template Toolkit` for the View, and `Class::DBI` for the Model data. And although the name begins with CGI, this module is also fully compatible with `mod_perl` as well, either as a direct handler or as an `Apache::Registry` "script". Even in CGI mode, many things are lazy-loaded and can be cached, for rapid response!

The last half of the class will be spent building a complete application using `CGI::Prototype`, including deployed `Class::DBI` classes using `DBD::SQLite` for the database.

Debugging Web Applications (quarter day)

Speaker: Randal Schwartz

Debugging web applications can be annoying at best, and virtually impossible if you don't follow some basic guidelines. Randal talks about his methodologies of debugging web applications, starting with the basics of avoiding those "500 errors" through a simple placement of `CGI::Carp`. He then shows how to use a death handler that incorporates `Data::Dumper` as a means of placing a breakpoint in your web code to reveal the state of the application at any selected point. Randal also talks about using `Devel::Cover` and good web tests with `WWW::Mechanize` (and its cousin, `CGI::Prototype::Mecha`) to ensure that the testing covers your code, so you won't have as many surprises in the field once your code is deployed.

When Perl is Not Quite Fast Enough (half day)

Speaker: Nicholas Clark

This is a tutorial on how to make Perl run faster, without having to use C or XS. In it you will learn ways I've found to make your troublesome slow scripts go faster, and how you can try to avoid some problems in the first place.

The tutorial starts with ways to cheat and duck the whole optimising problem, as that's usually a much better solution. I present a systematic approach to optimising in general, and show the Perl tools available to help you carry it out. By explaining why the cause of slowness in Perl programs often differs from your experience of compiled languages, you will gain a better feel for the right types of trade-offs to make to gain speed.

The second half presents a smorgasbord of optimising ideas to try out, and speed traps to locate and avoid. There's bound to be at least one approach you've never seen before.

The tutorial will demonstrate the techniques using specific real code as examples, showing code before and after. Running this interactively will give a feel for what sort of results can be achieved.

Creating Applications with PostgreSQL and Perl (half day)

Speaker: Andrew Dunstan

In this seminar learn to use the features of PostgreSQL in an application — with the emphasis on Perl-based applications. This session is about putting the knowledge from the previous sessions to practical use. Especially suitable for data architects, application architects, and application programmers.

- Using layered application design
- ServerSide vs. ClientSide programming
- Loading serverside language(s)
- Trusted and Untrusted languages
- Simple functions
- Triggers
- Returning a record
- Returning a set of things
- Calling back to the database
- PLPerl limitations, and future plans
- Getting and installing the client libraries
- `libpq` – what it is, who uses it and who doesn't
- Using DBI with PostgreSQL
- PostgreSQL programming for web apps (including `mod_perl`)
- Prepared Statements, and why you should use them whenever possible
- Other client interfaces (perl and non-perl).
- Case Studies from the speaker's experience
- Best Practice tips

Linux Lunacy V

Seminars at a Glance

Linux Lunacy V and Perl Whirl '05 are running concurrently. Attendees can mix and match, choosing courses from both conferences.

You may choose any combination of full-, half-, or quarter-day seminars for a total of two-and-one-half (2.5) days' worth of sessions. The conference fee is \$995 and includes all courses, course materials, and the Bon Voyage Cocktail Party.

Risk Management (quarter day)

Speaker: Ken Pugh

Risk management revolves around minimizing your exposure to risk and your losses when a risky event occurs. In this session, we'll examine the basic methods for managing risks. We'll explore risk assessment techniques with some interactive exercises and develop a risk management plan for a Linux system.

Firewall Basics (quarter day)

Speaker: Ken Pugh

This session encompasses network concepts and packet filtering. It starts with an examination of the common Internet protocols IP, TCP, UDP, and ICMP. It pays particular attention to the packet headers. It then covers how packet filtering works in both a stateless and stateful manner. In an interactive mode, the attendees will create a security policy to implement in the "Setting up iptables" session.

Setting Up iptables (quarter day)

Speaker: Ken Pugh

This session starts with a review of the security policy that a firewall is to implement. We'll then set up iptables to implement the firewall. We'll create rules and chains to perform the necessary packet filtering. Bring your Linux laptop with a wireless card to this session. This session assumes that you either attended the Firewall Basics session or that you have the equivalent knowledge.

Intrusion Detection (quarter day)

Speaker: Ken Pugh

We'll explore the basic concepts of intrusion detection, concentrating on how suspicious traffic is detected, as well as common types of attacks. Then we'll use snort. Finally, we'll examine some of the built-in rules and write a few rules of our own. Bring your Linux laptop with a wireless card to this session.

Socket Programming for Linux (quarter day)

Speaker: Ken Pugh

This session covers programming TCP and UDP using sockets. We cover creating both a client and a server. Bring your Linux laptop to this session. This session assumes that you either attended the Firewall Basics session or that you have the knowledge of TCP and UDP and that you have programmed in C.

New Developments in ext3 Filesystem (quarter day)

Speaker: Ted Ts'o

This talk will discuss upcoming new features that are being developed for the ext3 filesystem, including extent-based allocation, enhanced scalability features, allocation reservations, and delayed reservations.

Introduction to the Linux Kernel (full day)

Speaker: Ted Ts'o

This talk will be an introduction to the Linux kernel, and will touch upon the basic structure of the kernel, what features it provides, and the most important algorithms it employs. It will not contain any detailed examination of the source code, but will, rather, offer an overview and roadmap of the kernel's design and functionality. Topics covered will include the Linux scheduler, virtual memory system, filesystem and I/O layers, and networking stacks.

The Linux Boot Process (quarter day)

Speaker: Ted Ts'o

Ever wondered what happens when your Linux box boots? This talk will take you through the boot process, starting from Lilo or Grub, and moving through the kernel initialization and initial ram disks, to the /etc/rc.d boot scripts.

Recovering From Hard Drive Disk Disasters (half day)

Speaker: Ted Ts'o

Ever had a hard drive fail? Ever kick yourself because you didn't keep backups of critical files, or you discovered that your regular nightly backup didn't? (Of course not, you keep regular backups and verify them frequently to make sure they are successful.) For those of you who don't, this tutorial will discuss ways of recovering from hardware or software disasters. Topics covered will include backup strategies as well as low-level techniques to recover data from a corrupted ext2/ext3 filesystem when backups are not available: recovering from a corrupted partition table, using e2image to back up critical ext2/3 filesystem metadata, and using e2fsck and debugfs to sift through a corrupted filesystem.

Hands-On GUI Programming with Qt 4 (half day)

Speaker: Scott Collins

Qt is the dual-licensed (GPL and commercial), cross-platform, GUI library at the heart of KDE.

This session will start with an overview of Qt and Qt programming, briefly examining the changes from Qt3 to Qt4, and then dive into a hands-on tutorial implementation. Attendees will build a working, non-trivial application using Qt. Programmers new to Qt or to GUI programming in general will gain a working knowledge of the Qt toolbox, how to apply it to their specific problems, and how easy it becomes to expand the domain of their Linux app to other platforms. Current Qt3 users will learn the changes in the landscape for Qt4, and the tools available to migrate their existing code. This session is aimed at C/C++ programmers from just above the novice level and up, who want to learn about building GUIs and/or what's new in the latest generation of Qt.

An Introduction to Voice- and Video-Over-IP (half day)

Speaker: maddog

Tired of paying those phenomenal phone bills? Tired of paying for every little service that the telephone company deems you should have? Want to save your business heaps of money, yet still get the type of phone service you dream of? Then welcome to the wonderful world of OpenVoIP.

Through projects like Bayonne, GnomeMeeting and Asterisk people can finally do the things that were demonstrated at the 1969 World's Fair, at a price they can afford. By the end of this half-day session you *will* be able to:

- make a long distance phone call from PC to PC for free
- make a long distance phone call from PC to any phone for pennies
- make a video call PC to PC for free
- learn about various low-cost devices to enhance your VoIP experience and many more things. This talk will be aimed at a minimum of “technobabble” and a maximum of “here is how you do it, and here is where you get more information”. Remember, the author wrote the first edition of *Linux: for Dummies*.

The Brave New World of IPv6 (quarter day)

Speaker: Michael H. Warfield

IPv6 is the “Next Generation” IP protocol but has been largely ignored in areas of the world rich in IPv4 addressing capacity.

But, IPv6 is available anywhere the Internet is available, whether end users realize it or not. IPv6 has been gaining ground in various regions of the world and segments of Internet Society and represents new challenges to the established infrastructure.

What will be learned:

- What IPv6 is, how is it deployed and how well is it supported
- What IPv6 addressing looks like and what various address fields mean
- Support in various operating systems and environments
- How to access IPv6 anywhere on the IPv4 Internet
- Some security features and issues with IPv6
- Various advanced techniques in IPv6 access and security

Though IPv6 has been in existence for many many years, people in North America are largely unaware of how easy it is to access IPv6. This will be a fresh look at accessing and using IPv6 over current Internet infrastructures and some of the consequences of ignoring it (by end users or ISPs).

PostgreSQL: Seminars of Interest to both Linux Lunacy and Perl Whirl Attendees

PostgreSQL: Advanced Topics (half day)

Speaker: Andrew Dunstan

The advanced facilities of PostgreSQL can make your applications work efficiently and securely. We explore what those facilities are, and how to make sure they are working well. This seminar will be very useful for database administrators, systems administrators, data architects, application architects and application programmers.

- Composite types, nested types, domains, and arrays
- Views
- Permissions and Security
- Schemata
- Rules
- Triggers
- Stored Procedures and Functions
- Transactions
- Tuning PostgreSQL
- Configuration parameters
- Statistics
- Using EXPLAIN ANALYSE
- Vacuuming
- Connection pooling
- Replication
- Backup and Restoration

Introduction to PostgreSQL (quarter day)

Speaker: Andrew Dunstan

How and why to get PostgreSQL up and running. Designed for those who don't know much about databases, and those who know about databases but don't know much about PostgreSQL. Bring laptops -- we might include an installfest if time permits.

- What is a database? Why use one? Why PostgreSQL?
- Installing PostgreSQL
- Introduction to PostgreSQL Server and Client programs
- Configuring PostgreSQL
- Using the psql client
- Administrative GUIs
- PostgreSQL directions (or why I helped on the Windows port even though I'm a Linux fan)

PostgreSQL and Database Basics (quarter day)

Speaker: Andrew Dunstan

Introductory or refresher session covering the major concepts that underlie most database systems. If you don't know or can't remember what a LEFT OUTER JOIN is, or what the difference is between a UNION and a JOIN, this session is for you.

- Relations (a.k.a. tables)
- PostgreSQL data types
- The four basic operations (SELECT INSERT UPDATE DELETE)
- Functions and expressions
- Table JOINS and their flavors
- Table UNIONS
- Indexes
- Foreign keys
- Constraints
- Sequences and ID fields

Pricing and Booking Information

Course Fees: \$995 for Perl Whirl and/or Linux Lunacy (mix and match without any additional cost). Only passengers booked through Geek Cruises will be admitted.

Deposit: \$400 per person, due at time of booking.

Cabin Type	Cruise Rate (per person)
Standard Inside	\$ 699 (GS* available)
Standard Outside	\$ 799 (GS,* available)
Outside w/ Verandah	\$ 899 (GS,* available)
Mini Suite	\$ 999
Full Suite	\$1,499 (available)

(Cruise rates subject to change and availability.)

3rd and/or 4th Person Rate: ages 19 and older, \$649; ages 2 – 18, \$549; under 2 years old, \$499.

Single Occupancy: 150% for some inside cabins; 200% for all other cabins.

Port Charges and Taxes: Port charges/taxes are \$192 per person. The gratuity fee is \$70 per person; this fee covers the Dining Room Waiter, Assistant Waiter, Head Waiter, Maitre D', and Cabin Steward.

Pre-cruise Hotel Stays: Sightsee Tampa from your home base at the luxurious Hilton Tampa Airport Westshore. (Pricing is per person.)

	1 Night	2 Nights	3 Nights
Shared Double	\$83	\$165	\$248
Single Occupancy	\$165	\$325	\$490
3rd Person	\$30	\$60	\$90

Full payment is due on July 1, 2005 (or, if you book after July 1, at the time of booking). Please note: space will be very limited by July 1.

Foreign Booking Fees and Additional Payment Information: There is a FedEx shipping charge (cruise tickets) of \$60 per foreign residence (\$35 per Canadian residence). There is a \$25 charge for returned checks.

Air Add-ons: Airfare from most major cities is available through the cruise line. You can call our office for this pricing. (These rates include transfers to/from the dock/airport plus transfers to/from your hotel if we've booked the hotel as well.) In most cases, however, you will find better airfares on your own.



Andrew Morton

Andrew Morton is the lead maintainer for the Linux public production kernel. Morton works with Linux creator Linus Torvalds, the kernel subsystem maintainers, Linux distribution companies, hardware vendors and other interested parties to ensure that the public production kernel meets their needs. He is the final arbitrator on determining what code is accepted into the Linux production kernel. Morton has worked in software development for more than 20 years. As principal engineer at Digeo, he was responsible for the base operating system in the company's broadband digital home entertainment products. Prior to Digeo, he was product development manager for Nortel Networks Australian R&D labs. Andrew previously served as managing director of an Australia-based personal computer firm and also worked as a hardware engineer for an Australian maker of digital gaming equipment.

Linux Kernel Development (quarter day)

Speaker: Andrew Morton

Curious as to how, why, and whose code finds its way into the kernel? This high-level, after-hours discussion (and Q&A) will cover the various roles and responsibilities of kernel developers, resourcing, motivational, and management issues and how these have changed over time. Also the kernel's requirements, development, review, and testing processes will be revealed.

The audience will come away with an understanding of why kernel developers do what they do — and how they do it.

☺ Physically challenged available

***Guaranteed Share (GS) Fares:** This plan is for passengers who are coming on a Geek Cruise by themselves and wish to share a cabin with another Geek Cruises passenger (not the same gender) in an inside or outside cabin only. The prices are the same as the per person double occupancy rates. Share Passengers who smoke are not to do so in the cabin, unless okayed by fellow roommates. We try to match passengers with someone close in age, whenever possible.

Note: Carnival Cruise Lines will not accept any booking unless a fully completed Reservation Form is accompanied with a per-person deposit:

http://www.geekcruises.com/booking/1105_booking.htm

Linux Kernel Disk I/O (half day)

Speaker: Andrew Morton

This seminar will provide a description of the design and implementation of the 2.6 kernel's disk bulk-I/O design. Topics covered will include:

- read-ahead
- write-back
- cache coherence
- the relationship between page-cache
- disk buffers
- disk block mappings

The new-in-2.6 disk "BIO" layer will also be described, as well as the new-in-2.6 direct-IO driver.

Linux Kernel Memory Reclaim (half day)

Speaker: Andrew Morton

In this seminar you will learn about the design and implementation of the 2.6 kernel's caching, swapping, and memory reclaim architecture.

Some of the topics we will cover include:

- page reclaim
- page aging
- the role of swap-cache
- swap-out and swap-in

The discussion will also cover the reclaim of VFS cache memory and the balancing between page reclaim and VFS cache reclaim.

Geek Cruises, Inc.
1430 Parkinson Avenue
Palo Alto, CA 94301
650-327-3692
928-396-2102 fax
215-519-0141 cell
neil@geekcruises.com
CST# 2065380-40

