Course : Perl language, advanced programming

Practical course - 2d - 14h00 - Ref. LAP
Price : 1470 CHF E.T.

This course will teach you how to improve the performance of your programs and maintain them efficiently. You'll master the main kernel primitives, the creation of graphical interfaces and input masks. Finally, you'll discover the more modern approach of Object-oriented programming.

Teaching objectives

At the end of the training, the participant will be able to:

- ✓ Handling system functions with Perl
- ✓ Implementing Object concepts
- ✓ Using Perl/Tk
- ✓ Implementing GTK

Intended audience
IT specialists involved in operating procedures in Unix, Linux or Windows environments.

Prerequisites
Good knowledge of the Perl language or knowledge equivalent to that acquired in the course "Perl language" ref. PRL.

Course schedule

1 System programming

- Reminder of process principles: fork and exec.
- Reminder of the principles of signal management.
- Send a signal to processes: the kill function.
- Signal management: sigtrap pragma, installing a signal manager.
- Programming sockets and services from IO::Socket to Net::Server.
- Introduction to thread programming.

Hands-on work
Write Perl programs to manipulate system functions.

(2) Perl and the Object

- Some definitions : Object-oriented programming, class, instance, constructor, destructor, inheritance...
- Object implementation: package for class, hash for attributes, functions for methods.
- Instance creation: the blessing (bless) of the object reference.
- Program a builder.
- Act when the object is destroyed, the DESTROY method.
- Inheritance: the @ISA array and the UNIVERSAL class.
- Method overload, use of SUPER.
- Abstraction: the future of the object in Perl, from Class::Std and Class::Accessor to Moose.

Hands-on work
Class programming. Implement object concepts (inheritance, overloading, etc.).

(3) Graphical user interface: from Perl/Tk to GTK

- Understand and maintain code written in Perl/Tk.
- Presentation and installation of Perl/Tk.
- Geometry manager: Pack, Grid, Place.
- Widgets by family.
- Buttons: Button, Checkbutton, Radiobutton.
- Label and entry field: Label, Entry.
- Scrollbar, Listbox, Menubutton, Optionmenu, Text.
- Containers: Canvas, Frame, Toplevel.

Hands-on work
Creation of an application with a graphical interface in Perl/Tk and event-driven management.

(4) Writing new code using GTK

- Link between Perl and GTK: install GTK and its libraries.
- Widgets: hierarchy, creation, keyboard gas pedals, basic containers...
- Widget: signals and events.
- Button types: toggle, radio...

Hands-on work
Handling various components: buttons, button box, frame widget, etc.

(5) Developments

- Perl 5.10, 5.12, 5.14 features.
- Perl 6: a new language.
- New features: sigils, static and dynamic typing, formal function parameters.
- A true object language.
- Junctions and lazy evaluation.