GNU/Linux and C++

About Me

- Software Analyst – Citi Alternative Investments
- Chief Operations Officer – Zinkk, Inc.
- Education
  - *Associates in Computer Science (OCC)*
  - *Bachelors in Computer Science (NJIT)*

Brief C++ History

- 1972 – C was developed by Dennis Ritchie at Bell Labs (AT&T)
- 1979 – Bjarne Stroustrup begins work on C++ at Bell Labs
- 1992 – Alexander Stepanov begins research into generic programming
- 1994 – First version of Standard Template Library appears
- 1998 – ISO committee officially publishes C++98 (with Standard Library)
- 201X – The next ISO standard, *C++0X*, due out very soon

What C++ Is (and Isn't)

- C++ is compiled
- C++ is free form
- C++ is platform independent
- C++ is a multi-paradigm programming language

- C++ *is not* C
- C++ *is not* interpreted
- C++ *does not run* on a virtual machine
- C++ features *may be* different depending on the compiler

C++ vs. The *Other* Brands

- **C#, Java, Python** are compiled into an intermediate byte code which is executed on a virtual machine
- **C#, Java, Python**'s virtual machine must be compiled and tested for individual platform configurations
- *C++ is compiled into native machine code and executed for the specific CPU and operating system*
- *C++ executables gain a speed advantage because they are immediately compiled into native CPU code*

C++ Syntax

- Java syntax is very similar to C++
- Atomic data types are the same keywords (e.g. *int, float, double, char, etc*) as Java and C
- C Preprocessor directives are the same: *#include, #if, #define*
- C++ uses the `curly braces` and `semi-colon` as for blocking out code and terminating expressions

C++ Specifics

- Generics
- Object-oriented
- Procedural
- Multiple Inheritance
- Modular

Brief GNU/Linux History

- 1983 – Richard Stallman starts the GNU open source project
- 1985 – Stallman creates the Free Software Foundation
- 1991 – Linus Torvalds releases first version of `Linux` kernel
- 1991 – GNU General Public License version 2.0 released
- 1994 – First version of Red Hat Linux
- 2004 – Ubuntu first release

What GNU/Linux Is

- Linux *is a Unix-like* operating system kernel; GNU *is* the userland tools and compiler libraries

- GNU/Linux *is* completely open source under GPLv2, Apache, LGPL and MIT

- GNU/Linux software is primarily written in C (about 71%) and various languages (C++, Shell scripting, Perl, and assembly languages)

Fun Facts about Linux

- Estimated that around 60 percent of the world's servers run a flavor of Linux
- Linux is used inside appliances, set-top boxes, phones and medical equipment
- In Debian GNU/Linux 4.0 the distribution contains over 283 million lines of source code which would cost an estimated $7.37 billion dollars to develop conventionally
- The TiVo video device uses a custom version of Linux

Shell Scripting

- A script written specifically for a command line interpreter (/bin/sh or /bin/csh)
- Generally used to automate a series of commands the same way across multiple systems
- Can be in many programming languages: *Python, Perl, Ruby, Sh, etc*
- GNU Build System applications generate scripts for compiliation, configuration and installation of packages

GNU Build System

- Autoconf
- Automake
- Libtool
- GCC

GNU Compiler Collection

- Front-end compilers for *C, C++, Java, Ada, Objective-C, Fortran, Modula-2, Modula-3, Pascal, PL/I, D, Mercury, VHDL, Fortran 95, Fortran 2003* as of GCC 4.3

- Processor architectures include *ARM, IA-32 (x86), IA-64, x86-64, PowerPC, SPARC, Motorola 68000*

GNU C/C++ Compiler and Linker

- Plethora of compiler/linker options
- Basic ones include: *-g, -Wall, -p, -D, -L, -I, -O0*
- *Examples:*
- `g++ -o executable_name -g -Wall -I./include main.cpp`
- `gcc -o executable_name -g -Wall -I./include main.c`
- `g++ -DHAVE_CONFIG_H -I. -I. -I./build-aux -fPIC -Wall -Wno-write-strings -I/usr/local/include -I/usr/include -I./include -I./src/shared -I/usr/include/gtk-2.0 -I/usr/lib/gtk-2.0/include -I/usr/include/atk-1.0 -I/usr/include/cairo -I/usr/include/pango-1.0 -I/usr/include/glib-2.0 -I/usr/lib/glib-2.0/include -I/usr/include/freetype2 -I/usr/include/libpng12 -I/home/johnb/include -I/home/johnb/include/gtkextra-2.0 -I/usr/local/include/gtkextra-2.0 -g -O0 -MT src/realtime/lib_realtime_la-Network.lo -MD -MP -MF src/realtime/.deps/lib_realtime_la-Network.Tpo -c src/realtime/Network.cpp -fPIC -DPIC -o src/realtime/.libs/lib_realtime_la-Network.o`

URLs and References

- References (http://wikipedia.org)
- My Startup (http://zinkkinc.com)
- My Twitter (http://twitter.com/johnbellone)
- My E-mail (jvb4@njit.edu)
- Code (http://github.com/johnbellone/lecture1)

GNU/Linux and C++

About Me

- Software Analyst – Citi Alternative Investments
- Chief Operations Officer – Zinkk, Inc.
- Education
  - *Associates in Computer Science (OCC)*
  - *Bachelors in Computer Science (NJIT)*

Brief C++ History

- 1972 – C was developed by Dennis Ritchie at Bell Labs (AT&T)
- 1979 – Bjarne Stroustrup begins work on C++ at Bell Labs
- 1992 – Alexander Stepanov begins research into generic programming
- 1994 – First version of Standard Template Library appears
- 1998 – ISO committee officially publishes C++98 (with Standard Library)
- 201X – The next ISO standard, *C++0X*, due out very soon

What C++ Is (and Isn't)

- C++ is compiled
- C++ is free form
- C++ is platform independent
- C++ is a multi-paradigm programming language
- C++ *is not* C
- C++ *is not* interpreted
- C++ *does not run* on a virtual machine
- C++ features *may be* different depending on the compiler

C++ vs. The *Other* Brands

- **C#, Java, Python** are compiled into an intermediate byte code which is executed on a virtual machine
- **C#, Java, Python**'s virtual machine must be compiled and tested for individual platform configurations
- *C++ is compiled into native machine code and executed for the specific CPU and operating system*
- *C++ executables gain a speed advantage because they are immediately compiled into native CPU code*

C++ Syntax

- Java syntax is very similar to C++
- Atomic data types are the same keywords (e.g. *int, float, double, char, etc*) as Java and C
- C Preprocessor directives are the same: *#include, #if, #define*
- C++ uses the `curly braces` and `semi-colon` as for blocking out code and terminating expressions

C++ Specifics

- Generics
- Object-oriented
- Procedural
- Multiple Inheritance
- Modular

Brief GNU/Linux History

- 1983 – Richard Stallman starts the GNU open source project
- 1985 – Stallman creates the Free Software Foundation
- 1991 – Linus Torvalds releases first version of `Linux` kernel
- 1991 – GNU General Public License version 2.0 released
- 1994 – First version of Red Hat Linux
- 2004 – Ubuntu first release

What GNU/Linux Is

- Linux *is a Unix-like* operating system kernel; GNU *is* the userland tools and compiler libraries
- GNU/Linux *is* completely open source under GPLv2, Apache, LGPL and MIT
- GNU/Linux software is primarily written in C (about 71%) and various languages (C++, Shell scripting, Perl, and assembly languages)

Fun Facts about Linux

- Estimated that around 60 percent of the world's servers run a flavor of Linux
- Linux is used inside appliances, set-top boxes, phones and medical equipment
- In Debian GNU/Linux 4.0 the distribution contains over 283 million lines of source code which would cost an estimated $7.37 billion dollars to develop conventionally
- The TiVo video device uses a custom version of Linux

Shell Scripting

- A script written specifically for a command line interpreter (/bin/sh or /bin/csh)
- Generally used to automate a series of commands the same way across multiple systems
- Can be in many programming languages: *Python, Perl, Ruby, Sh, etc*
- GNU Build System applications generate scripts for compiliation, configuration and installation of packages

GNU Build System

- Autoconf
- Automake
- Libtool
- GCC

GNU Compiler Collection

- Front-end compilers for *C, C++, Java, Ada, Objective-C, Fortran, Modula-2, Modula-3, Pascal, PL/I, D, Mercury, VHDL, Fortran 95, Fortran 2003* as of GCC 4.3

- Processor architectures include *ARM, IA-32 (x86), IA-64, x86-64, PowerPC, SPARC, Motorola 68000*

GNU C/C++ Compiler and Linker

- Plethora of compiler/linker options
- Basic ones include: *-g, -Wall, -p, -D, -L, -I, -O0*
- *Examples:*
- `g++ -o executable_name -g -Wall -I./include main.cpp`
- `gcc -o executable_name -g -Wall -I./include main.c`
- `g++ -DHAVE_CONFIG_H -I. -I. -I./build-aux -fPIC -Wall -Wno-write-strings -I/usr/local/include -I/usr/include -I./include -I./src/shared -I/usr/include/gtk-2.0 -I/usr/lib/gtk-2.0/include -I/usr/include/atk-1.0 -I/usr/include/cairo -I/usr/include/pango-1.0 -I/usr/include/glib-2.0 -I/usr/lib/glib-2.0/include -I/usr/include/freetype2 -I/usr/include/libpng12 -I/home/johnb/include -I/home/johnb/include/gtkextra-2.0 -I/usr/local/include/gtkextra-2.0 -g -O0 -MT src/realtime/lib_realtime_la-Network.lo -MD -MP -MF src/realtime/.deps/lib_realtime_la-Network.Tpo -c src/realtime/Network.cpp -fPIC -DPIC -o src/realtime/.libs/lib_realtime_la-Network.o`

URLs and References

- References (http://wikipedia.org)
- My Startup (http://zinkkinc.com)
- My Twitter (http://twitter.com/johnbellone)
- My E-mail (jvb4@njit.edu)
- Code (http://github.com/johnbellone/lecture1)