Lenguajes de programación Mini-COMO.

Risto S. Varanka *mailto:risto.varanka@helsinki.fi*
Traducción: Antonio Álvarez Platero *uji01380@uji.infomail.es* 6 enero 2000. Traducción: 5 marzo 2000.

Una breve comparación de los principales lenguajes de programación para Linux y de las principales librerías para la creación de interfaces gráficas de usuario (GUIs) en Linux.

Contents

1 Introducción

Linux es un sistema operativo fascinante porque permite a cualquier usuario participar en su desarrollo. La variedad de lenguajes disponibles, sin embargo, puede llevar a la confusión a los desarrolladores principiantes. Este documento hace un listado de las opciones más comunes existentes para el desarrollo diario. (Bueno, el más común y el principal, según yo lo veo). Mi propósito no es ni el de reseñar los lenguajes ni el de determinar cuál es el mejor. Cada lenguaje es una herramienta que sirve para determinados trabajos y gustos. Puede obtener más información (a menudo conflictiva) con facilidad, si pregunta por ahí o si mantine los oídos alerta. La sección de enlaces en este documento le dará indicaciones para que pueda investigar por su cuenta.

Hay una pléyade de lenguajes y librerías de programación para Linux, así que este documento solamente cubre los lenguajes más comunes y los conjuntos de herramientas para la construcción de interfaces gráficas de usuario del momento. Nótese también que, tanto los lenguajes como las herramientas para construcción de interfaces de usuario, se verán desde el punto de vista de Linux: no se tratarán sus características para otras plataformas.

Este documento se ha sumado recientemente al LDP (Linux Documentation Project: Proyecto de Documentación de Linux), de modo que no ha habido muchas oportunidades de recibir comentarios por parte de la comunidad. Sin embargo, se saca a la luz con la esperanza de que pueda ser útil a los que estén interesados en programar en Linux. Un signo de interrogación en las tablas indica que se carece de información. Si puede completarla, por favor, contacte con el autor.

1.1 Copyright

1.2 Licencia

Los siguientes términos de licencia son de aplicación a todos los documentos de LDP, a menos que así se especifique en el documento. Los documentos de LDP pueden ser reproducidos y redistribuidos por completo o parcialmente, en cualquier medio físico o electrónico, siempre que se reproduzca este aviso de licencia en dicha reproducción. Se permite y se anima a su redistribución comercial. En caso de redistribución, se agradecerá la comunicación de la misma, vía correo electrónico, a los autores con treinta días de antelación, para darles tiempo a poner al día los documentos.

1.2.1 Requisitos para los documentos modificados.

Todos los documentos que sean modificados, incluyendo traducciones, antologías o documentos parciales, deben cumplir los siguientes requisitos:

1. La versión modificada debe estar etiquetada como tal.
2. La persona que haga las modificaciones debe identificarse.
3. Se debe conservar el reconocimiento al autor original.
4. Se debe identificar la localización del documento original no modificado.
5. El nombre (o nombres) del autor (o autores) originales, no puede ser utilizado para afirmar o implicar la aprobación del documento resultante sin el permiso del (o de los) autor (o autores) originales.

Ademas se pide que:

1. Las modificaciones (incluidas las supresiones) sean comunicadas.
2. El autor sea notificado por correo electrónico de las modificaciones antes de su redistribución, si se da dirección electrónica en el documento.

Como excepción especial, las antologías de documentos de LDP pueden incluir una única copia de estos términos de licencia en un lugar visible dentro de la antología y reemplazar otras copias de esta licencia haciendo refrencia a esta única copia de la licencia sin que, por ello, el documento sea considerado modificado para los propósitos de esta sección.

El hecho de agregar documentos de LDP a otros documentos o programas en el mismo medio no conlleva a que esta licencia se aplique a esos otros trabajos

Todas las traducciones, documentos derivados, o documentos modificados que incorporen cualquier documento LDP no pueden tener términos de licencia más restrictivos que éstos, excepto en el caso de requerir a los distribuidores que hagan disponible el código fuente del documento resultante.

1.3 Exención de Responsabilidad

ESTE DOCUMENTO CUBRE UN CAMPO AMPLIO Y EN CONSTANTE CAMBIO. ASÍ PUES, LA INFORMACIÓN QUE CONTIENE PUEDE SER INCORRECTA O ESTAR DESFASADA. EL USO QUE SE HAGA DE ESTE DOCUMENTO Y TODA INFORMACIÓN EN ÉL CONTENIDA, QUEDA BAJO SU RESPONSABILIDAD. EL AUTOR (Y EL TRADUCTOR) NO DA NINGÚN TIPO DE GARANTÍA EXPLÍCITA O IMPLÍCITA.

1.4 Autor

Se agradece el envío de comentarios al autor en: *risto.varanka@helsinki.fi* `<mailto:risto.varanka@helsinki.fi>`.

La página del autor se encuentra en:*http://www.helsinki.fi/˜rvaranka/* `<http://www.helsinki.fi/˜rvaranka/>`.

1.5 Agradecimientos

Quiero expresar mi agradecimiento a todas las personas que han remitido sus comentarios sobre temas de lenguajes de programación. Esta información me ha proporcionado una mejor visión de los diferentes lenguajes, y espero que en el futuro ayuden a este mini-COMO a madurar con el tiempo. Quisiera, sobre todo, agradecer a las personas de IRCnet channel #linux: Morphy, Bluesmurf, Vadim, Zonkˆ, Rikkus y otros cuyos nombres haya olvidado. Agradecimientos también pra Stig Erik Sandoe por sus útiles consejos.

1.6 Enlaces

Listas exhaustivas de librerías y herramientas de desarrollo en Linux:

- *Freshmeat* `<http://www.freshmeat.net/appindex/development/>`
- *Linux Development Tools* `<http://www.hotfeet.ch/˜gemi/LDT/>`
- *linuxprogramming.com* `<http://www.linuxprogramming.com/>`

El *Hacker FAQ* `<http://www.tuxedo.org/˜esr/faqs/hacker-howto.html>` de Eric S. Raymond es otro texto interesante para los desarrolladores principiantes en Linux. Se centra an aspectos culturales y sicológicos del desarrollo del código libre.

Otros *documentos LDP* `<http://www.linuxdoc.org/>` que cubren temas generales de programación incluyen el Reading List HOWTO and the Linux Programmer's Guide - se han escrito algunos más para temas específicos.

2 Lenguajes de programación

2.1 Conceptos en la Tabla

Lenguaje

Indica el nombre común del lenguaje.

Principiante

Indica lo apropiado que es el lenguaje para personas con poca experiencia en programación. Un lenguaje que se encuentre marcado por un "sí" debería ser viable como primer lenguaje de programación para un principiante.

Rendimiento

La probable velocidad de ejecución de sus aplicaciones cuando se usen comercialmente. Las prestaciones dependen más de sus habilidades algorítmicas de programación que del propio lenguaje. Empíricamente, C, C++ y Fortran son necesarios a veces porque pueden dar mejores prestaciones que otros lenguajes -otras veces pueden ser pesados para el proposito original. (Una idea para las pruebas de rendimiento de los lenguajes sería la implementación de un sencillo algoritmo de ordenación en todos ellos y la comparación posterior de los tiempos de ejecución. ¿Alguien querría ayudarme con ésto?

POO, Progamación Orientada a Objetos vs. otros paradigmas

La Programación Orientada a Objetos es un paradigma de programación importante que está ganando popularidad. En la programación orientada aobjetos, las estructuras de datos y los algoritmos se integran en unidades, a menudo llamadas clases. La POO contrasta con la programación procedimental (que usa algoritmos y estructuras de datos separados). No depende estrictamente del lenguaje: se puede hacer POO con lenguajes no clasificados como tales (por ejemplo C), y se puede programar en estilo procedimental con lenguajes clasificados como Orientados a Objetos. He clasificado como de POO a lenguajes que tienen características especiales o añadidos que facilitan la POO. Los lenguajes funcionales (LISP, por ejemplo) son de una clase un poco diferente -entre otras cosas, los lenguajes funcionales son un superconjunto de POO. La programación lógica (Prolog), también llamada programación declarativa, por otro lado, no tiene relación con otros tipos de programación en un sentido similar.

RAD, Rapid Application Development (Desarrollo rápido de aplicaciones)

Más dependientes de las herramientas que se usan que del lenguaje propiamente dicho. Hay un COMO de herramientas de desarrollo para GUI (Interfaz gráfico de susuario), aunque está desfasado. Con una buena herramienta gráfica se pueden realizar RAD. A veces los RAD se basan en la reutilización de código también, por lo que el software libre nos puede proporcionar un buen punto de partida.

Ejemplos

Describe los campos de programación en los que normalmente se usa el lenguaje. Se dan otros tipos de usos, buenos y malos, aunque no son tan frecuentes.

Comentarios

Información adicional sobre el lenguaje, como son sus capacidades y dialectos.

2.2 Lenguajes Principales

```
PERL
Principiantes: Sí - POO: Sí
```

```
Ejemplos: Scripts, administración de sistemas, www
Comentarios: Potente para la manipulación de textos y cadenas

Python
Principiantes: Sí - POO: Sí
Ejemplos: Scripts, scripts de aplicaciones, www
Comentarios:

TCL
Principiantes: Sí - POO: No
Ejemplos: Scripts, administración de sistemas, aplicaciones
Comentarios:

PHP
Principiantes: Sí - POO: Sí
Ejemplos: Www
Comentarios: Popular para las bases de batos basadas en web

Java
Principiantes: Sí - POO: Sí
Ejemplos: Aplicaciones para plataformas cruzadas, www
Comentarios:

Lisp
Principiantes: Sí - POO: Funcional
Ejemplos: Modos de Emacs (para elisp), AI
Comentarios: Variantes Elisp, Clisp y Scheme

Fortran
Principiantes: No  - POO: No
Ejemplos: Aplicaciones matemáticas
Comentarios: Variantes f77 y f90/95

C
Principiantes: No  - POO: No
Ejemplos: Programación de sistemas, aplicaciones
Comentarios: Muy popular

C++
Principiantes: No  - POO: Sí
Examples: Aplicaciones
Comentarios:
```

2.3 Programación del shell

Los shell son también unos entornos de programación importantes. No los cubro aquí porque no domino el tema con la suficiente profundidad aún. El conocimiento de los shell es importante para quien trabaje con Linux regularmente, y más aun para los administradores de sistemas. Hay similitudes entre la programación del shell y los scripts -a menudo consiguen los mismos propósitos y tenemos la oportunidad de elegir entre los shell nativos o un lenguaje de scripts. Entre los más populares están los shell bash, tcsh, csh, ksh y zsh.

Puede obtener información acerca de su shell con *man* comando, *man bash* por ejemplo.

2.4 Otros Lenguajes

Otros lenguajes dignos de mención: AWK, SED, Smalltalk, Eiffel, ADA, Prolog, assembler, Objective C, Logo, Pascal (p2c converter)

2.5 Enlaces

- *Un sitio de información general* `<http://www.tunes.org/Review/Languages.html>` sobre lenguajes de programación, mucha información y opiniones
- *TCL* `<http://www.scriptics.com/>`
- *PERL* `<http://www.perl.org/>`
- *Python* `<http://www.python.org/>`
- *PHP* `<http://www.php.net>`
- *Java* `<http://www.javasoft.com/>`
- *clisp* `<http://clisp.cons.org/~haible/packages-clisp.html>`

3 Conjuntos de herramientas para la construcción de interfaces gráficos (GUI Toolkits)

3.1 Conceptos en la Tabla

Librería/Biblioteca

Nombre común o abreviatura del conjunto de herramientas

Principiantes

Si el conjunto de herramientas es adecuado para un programador principiante.

Licencia

Los diferentes tipos de licencias para los distintos conjuntos de herramientas gráficas tienen una importancia práctica. Las licencias de GTK+ y de TK le permiten desarrollar tanto aplicaciones de código libre como de código propietario sin tener que pagar licencia. La licencia de Motif requiere pagar, mientras que la licencia de QT requiere pagar solamente si se escribe código propietario.

Lenguaje

El lenguaje que con más frecuencia se usa con el conjunto de herramientas.

Vinculados

Otros lenguajes que pueden hacer uso del conjunto de herramientas.

Ejemplos

Aplicaciones que usan el conjunto de herramientas.

Comentarios

Información adicional sobre el conjunto de herramientas.

Librería	Principiante	Licencia	Lenguaje	Vinculados	Ejemplos	Comentarios
TK	Sí	Libre	TCL	PERL, Python, otros	make xconfig, TKDesk	
GTK+	No	Libre (LGPL)	C	PERL, C++, Python, muchos otros	GNOME, Gimp	
QT	No	Libre para código abierto	C++	Python, PERL, C, ¿otros?	KDE	
Motif	No	Propietaria	C/C++	Python, ¿otros?	Netscape, Wordperfect	

3.2 Principales conjuntos de herramientas para la creación de interfaces gráficas de usuario.

3.3 Enlaces

- *TK* `<http://www.scriptics.com/>`
- *GTK+* `<http://www.gtk.org/>`
- *QT* `<http://www.troll.no/>`
- *Motif* `<http://www.metrolink.com/>`