Programmeertalen mini-HOWTO

Risto S. Varanka,
Vertaald door: Ellen Bokhorst, bokkie@nl.linux.org 22 juli 2000

Een beknopte vergelijking tussen belangrijke programmeertalen voor Linux en belangrijke library's voor het aanmaken van grafische gebruikersinterfaces (GUI's) onder Linux.

Inhoudsopgave

1 Introductie

Linux is een fascinerend besturingssysteem omdat het iedere gebruiker laat participeren in de ontwikkeling ervan. De variëteit aan beschikbare programmeertalen, kan echter voor beginnende Linux ontwikkelaars verwarrend zijn. In dit document worden de meest gebruikelijke opties voor alledaagse ontwikkeling opgesomd en formuleert bovendien van deze talen de belangrijkste feiten.("meest gebruikelijk" en "belangrijk" zoals ik dat waarneem).

Mijn bedoeling is noch de programmeertalen onder de loep te nemen, noch vast te stellen welke de beste is. Iedere taal is een hulpmiddel die bij een aantal taken en smaken past. Je kunt gemakkelijk aan meer (vaak tegenstrijdige) informatie komen, als je eens om je heen vraagt of je oren open houdt. In de sectie Links in dit document vind je een aantal verwijzingen voor een eigen onderzoek.

Er is een overvloed aan programmeertalen en library's voor Linux beschikbaar, dus in dit document worden op het moment alleen de meest gebruikelijke programmeertalen en GUI (Graphical User Interface) toolkits behandeld. Het is de bedoeling dat dit document tamelijk neutraal is, maar ik heb niet alle beschikbare programmeertalen opgenomen. Aangezien ik in mijn oordeel ongetwijfeld op diverse wijzen ben bevooroordeeld, adviseer ik serieuze ontwikkelaars een kijkje te nemen op die sites die beter zijn in het catalogiseren van alle programmeertalen en library's. Alleen de Linux implementaties van de talen en GUI toolkits worden behandeld, er wordt over mogelijkheden op andere platformen van deze talen niet gediscusseerd of iets over geïmpliceerd.

Dit document is onlangs aan de LDP toegevoegd, dus er was geen kans voor veel feedback van de gemeenschap. Het is echter uitgegeven in de hoop dat het van nut zal zijn voor mensen die geïnteresseerd zijn in programmeren onder Linux en dan met name voor beginners. Een vraagteken in de tabellen geeft het ontbreken van informatie aan. Als je het in kunt vullen, neem dan alsjeblieft contact op met de auteur.

1.1 Laatste versie van het document

Je kunt de laatste aanpasssingen vinden op *http://www.helsinki.fi/˜rvaranka/Computer/Linux/HOWTO/* `<http://www.helsinki.fi/~rvaranka/Computer/Linux/HOWTO/>`

1.2 Copyright

1.3 Licentie

1.3.1 Vereisten voor aangepaste werken

Alle aangepaste documenten, inclusief vertalingen, bloemlezingen, en gedeeltelijke documenten, moeten voldoen aan de volgende vereisten:

1. De aangepaste versie moet als zodanig worden gelabeld.
2. De persoon die de aanpassingen maakte moet worden geïdentificeerd.
3. Erkenning van de oorspronkelijke auteur moet behouden blijven.
4. De lokatie van het oorspronkelijke ongewijzigde document moet zijn geïdentificeerd.
5. De naam van de oorspronkelijke auteur(s) mag niet worden gebruikt onderschrijving te beweren of impliceren van het resulterende document zonder permissie van de oorspronkelijke auteur(s).

Bovendien is het vereist dat:

1. Van de aanpassingen (inclusief verwijderingen) een aantekening wordt gemaakt.
2. De auteur voor de herdistributie via email van de aanpassing op de hoogte wordt gebracht, als een e-mailadres in het document is opgenomen.

Als een speciale uitzondering, mogen bloemlezingen van LDP documenten een enkele kopie van deze licentievoorwaarden op een opvallende lokatie opnemen binnen de bloemlezing en andere kopiëen van deze licentie vervangen door een verwijzing naar een enkele kopie van de licentie zonder het document als "aangepast" aan te merken als doel van deze sectie.

Louter een verzameling van LDP documenten met andere documenten of programma's op dezelfde media zal niet tot gevolg hebben dat deze licentie op die andere werken van toepassing is.

Alle vertalingen, afgeleide documenten, of aangepaste documenten welke enig LDP document verenigd mogen geen beperktere licentievoorwaarden dan deze hebben, behalve dat je van de distributeurs mag vereisen dat ze het resulterende document beschikbaar stellen in het bronformaat.

1.4 Disclaimer

IN DIT DOCUMENT WORDT EEN GROOT EN CONSTANT VERANDEREND DOMEIN BEHANDELD. DAAROM KAN DE INFORMATIE IN DIT DOCUMENT ONJUIST OF VEROUDERD ZIJN. AL HET GEBRUIK EN ALLE INFORMATIE HIERIN IS OP EIGEN RISICO. DE AUTEUR GEEFT GEEN WAARBORG OF GARANTIE, NOCH EXPLICIET NOCH IMPLICIET.

1.5 Auteur

Het insturen van feedback aan de auteur via: *risto.varanka@helsinki.fi* `<mailto:risto.varanka@helsinki.fi>` wordt verwelkomd.

De website van de auteur is te vinden op *http://www.helsinki.fi/˜rvaranka/* `<http://www.helsinki.fi/~rvaranka/>`.

1.6 Krediet

Ik ben verscheidene mensen dankbaar die opmerkingen gaven over programmeeronderwerpen. Deze conversaties zorgden ervoor dat ik een beter inzicht kreeg in de verschillende talen, en ik hoop dat toekomstige conversaties het mogelijk zullen maken deze mini-HOWTO in de loop der tijd verder te ontwikkelen. Ik zou vooral graag de mensen op het IRCNet kanaal #linux: Morphy, Bluesmurf, Vadim, Zonkˆ, Rikkus en anderen wiens naam ik ben vergeten, willen bedanken. Dank gaat ook uit naar Stig Erik Sandoe voor zijn behulpzame opmerkingen.

1.7 Links

Grondige lijsten met Linux development library's en tools:

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

De *Hacker FAQ* <http://www.tuxedo.org/~esr/faqs/hacker-howto.html> van Eric S. Raymond is een andere interessante tekst voor beginnende Linux ontwikkelaars. Het concentreert zich op een aantal culturele en psychologische aspecten van open source development.

Andere *LDP documenten* <http://www.linuxdoc.org/> behandelen algemene programmeeronderwerpen waaronder de Reading List HOWTO en de Linux Programmer's Guide - er zijn er nog meer over specifieke onderwerpen geschreven.

2 Programmeertalen

C, Lisp en Perl zijn traditionele hacking programmeertalen in de GNU/Linux cultuur; Python, PHP, Java en C++ hebben recent nieuwe grond verworven.

2.1 Concepten in de tabel

Taal

Een gebruikelijke naam van de taal.

Beginner

Geeft aan hoe geschikt de taal is voor mensen met weinig programmeerervaring. Een taal gemarkeerd met "ja" zou uitvoerbaar moeten zijn als eerste programmeertaal voor een beginner.

Performance

Hoe snel je applicaties naar alle waarschijnlijkheid zullen draaien wanneer je ze in productie neemt. De performance hangt meer af van je algorithmische programmeervaardigheden dan de feitelijke taal. Als vuistregel zijn C, C++ en Fortran soms noodzakelijk omdat ze een betere performance bieden dan andere talen - andere keren zijn ze wellicht ongeschikt voor het gewenste doel. (Een idee voor het niet wetenschappelijk "benchmarken" van de programmeertalen zou zijn een eenvoudig sorteeralgoritme te implementeren in alle programmeertalen en de uitvoertijden ervan te vergelijken. Hiermee wordt natuurlijk niet de performance van de feitelijke taal gemeten, aangezien dat concept geen zin heeft, maar slechts de implementatie. Natuurlijk geen erg betrouwbare of grondige methode, maar het zou een voorbeeld kunnen geven van het verschil in uitvoertijden van de verschilende talen. Kan iemand me hiermee helpen?)

OOP, Object-Oriented Programming vs. andere paradigma's

Object geöriënteerd programmeren is een belangrijk paradigma dat meer aan populariteit wint. In object geöriënteerd programmeren zijn gegevensstructuren en algoritmen in eenheden geïntegreerd, vaak classes genoemd. OOP staat vaak in contrast met procedureel programmeren (waarin aparte algoritmen en gegevenstructuren worden gebruikt). Het is niet strict afhankelijk van de taal: je kunt OOP doen in talen die niet als zodanig worden opgesomd (zoals bijvoorbeeld C), en in de procedurele stijl programmeren in talen die gecatalogiseerd zijn als OOP. Ik heb talen als OOP gecatalogiseerd die speciale mogelijkheden hebben of add-ons om OOP te vergemakkelijken. Functionele talen (zoals bijvoorbeeld Lisp) zijn een wat andere soort. - functioneel programmeren is onder andere een superset van OOP. Logisch programmeren (Prolog), ook declaratief programmeren aan de andere kant is in vergelijkbare zin niet gerelateerd aan de andere typen programmering.

RAD, Rapid Application Development

Eerder afhankelijk van de gebruikte tools dan de feitelijke taal. Er is ook een HOWTO over GUI development voor Linux, alhoewel het verouderd is. Met een goed grafisch hulpmiddel kun je RAD doen. RAD kan krachtig zijn wanneer bovendien gebaseerd op hergebruik van de code, dus vrije software zou een goed begin kunnen vormen.

Voorbeelden

Het benoemen van de programmeergebieden waarvoor de taal het vaakst wordt gebruikt. Andere goede (en slechte) gebruiken bestaan ook, maar ze zijn minder kenmerkend.

Opmerkingen

Extra informatie over de programmeertaal, zoals mogelijkheden en dialecten.

2.2 Belangrijke programmeertalen

```
Perl
Beginner: Ja - OOP: Ja
Voorbeelden: Scripting, sysadmin, www
Opmerkingen: Krachtig voor het afhandelen van tekst en strings

Python
Beginner: Ja - OOP: Ja
Voorbeelden: Scripting, applicatie scripting, www
Opmerkingen:

TCL
Beginner: Ja - OOP: Nee
Voorbeelden: Scripting, sysadmin, applicaties
Opmerkingen:

PHP
Beginner: Ja - OOP: Ja
Voorbeelden: Www
Opmerkingen: Populair voor webdatabases

Java
Beginner: Ja - OOP: Ja
Voorbeelden: Cross-platform applicaties, www
Opmerkingen: Uitgestrekt naar nieuwe gebieden, bv. e-commerce infrastructuur

Lisp
Beginner: Ja - OOP: Functioneel
Voorbeelden: Emacs modes (voor Elisp), AI
Opmerkingen: Varianten Elisp, Clisp en Scheme

Fortran
Beginner: Nee  - OOP: Nee
Voorbeelden: Wiskundige (wetenschappelijke) applicaties
Opmerkingen: Varianten f77 en f90/95

C
Beginner: Nee  - OOP: Nee
```

```
Voorbeelden: Systeemprogrammering, applicaties
Opmerkingen:

C++
Beginner: Nee  - OOP: Ja
Voorbeelden: Applicaties
Opmerkingen:
```

2.3 Shell Programmering

Ook shells bieden een belangrijke programmeeromgeving. Ik heb ze niet behandeld omdat ik dit terrein nog niet zo door en door begrijp. Kennis van shells is voor ieder belangrijk die regelmatig onder Linux werkt, nog meer zelfs voor systeembeheerders. Er zijn tussen shellprogrammering en andere soorten scripting overeenkomsten, vaak kun je er dezelfde doelen mee bereiken, en heb je de optie te kiezen tussen de shell en een aparte scripttaal. Tussen de meest populaire shells bevinden zich bash, tcsh, csh, ksh en zsh. Je kunt basisinformatie over je shell krijgen met de opdracht *man*, zoals bijvoorbeeld *man bash*.

2.4 Andere programmeertalen

Andere te vermelden talen: AWK, SED, Smalltalk, Eiffel, Ada, Prolog, assembler, Objective C, Logo, Pascal (p2c converter)

2.5 Links

- *Een algemene info site* <http://www.tunes.org/Review/Languages.html> over programmeertalen, heel veel info en meningen
- *TCL* <http://dev.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 GUI Toolkits

Het standaard grafische subsysteem voor UNIX en Linux, genaamd X, heeft zijn eigen library's voor GUI ontwikkeling. Ze leveren een low-level programmeerinterface voor X, maar neigen ernaar moeilijk te gebruiken te zijn. Oude eindgebruikers applicaties en andere toolkits maken er natuurlijk goed gebruik van. Tegenwoordig wordt de Linux GUI scene gedomineerd door GTK+ en Qt, aangezien twee populaire complete gebruikersomgevingen, GNOME en KDE, hierop zijn gebaseerd.

Library	Beginner	Licentie	Taal	Bindings	Voorbeelden	Opmerkingen
TK	Ja	Vrij	TCL	Perl, Python, anderen	make xconfig, TKDesk	
GTK+	Nee	Vrij (LGPL)	C	Perl, C++, Python, veel anderen	GNOME, Gimp	Zeer populair
QT	Nee	Vrij voor open source	C++	Python, Perl, C, anderen?	KDE	Zeer populair
Motif	Nee	Niet vrij	C/C++	Python, anderen?	Netscape, Wordperfect	*Lesstif* `<http://www.lesstif.org/>` iseen vrije vervanging
GNUstep	Nee	Vrij (LGPL)	Objective C	Guile, Java?	Niet zo bekend, maar zie de*applicatie lijst* `<http://www.gnustep.org/resources/apps.html>`	GNUstep is nog steeds onder ontwikkeling

3.1 Concepten in de tabel

Library

Algemene naam of afkorting van de toolkit.

Beginner

Of de toolkit geschikt is voor een newbie programmeur.

Licentie

De verschillende licenties voor de verschillende GUI toolkits hebben een praktische betekenis. GTK+, TK en GNUstep licenties maken het mogelijk zowel open source als closed source applicaties te ontwikkelen zonder te hoeven betalen voor een licentie. De Motif licentie moet worden betaald, terwijl voor de QT licentie alleen hoeft te worden betaald als je closed source programma's schrijft.

Taal

De programmeertaal die het meest met de toolkit wordt gebruikt.

Bindings

Andere talen die gebruik kunnen maken van de toolkit.

Voorbeelden

Applicaties die gebruik maken van de toolkit.

Opmerkingen

Extra informatie over de toolkit.

3.2 Belangrijke GUI Toolkits

3.3 Links

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

- *Motif* <http://www.metrolink.com/>
- *GNUstep* <http://www.gnustep.org/>