Les langages de programmation sur Mac

1) Introduction

L'hégémonie du langage Objective-C sur Mac est dû au succès foudroyant de l'iPhone (même et surtout hors de la sphère Mac), ce langage étant incontournable ou presque sur celui-ci. Java n'est d'ailleurs plus supporté par l'environnement de développement XCode pour construire des applications Mac.

Les systèmes d'exploitations vont-ils devenir mono-langages de programmation ? Windows vire au C#, Mac OS X carbure à l'Objective-C, Linux reste fidèle au C/C++. Pour les deux premiers, il s'agit bien d'une stratégie d'entreprise de Microsoft et Apple, comme Oracle qui s'assure l'exclusivité de Java avec le rachat de Sun. L'enjeu se situe au niveau de l'interface utilisateur graphique qui est l'emblème du système d'exploitation. Pourtant, la relation langage / programmeur est plus forte voire plus intime qu'avec le système d'exploitation.

Est-on alors contraint à cette monoculture ? Non ! Outre Objective-C et AppleScript, il existe bien d'autres langages de programmation disponibles sur Mac. Heureusement aussi, la vitalité de langages comme Pascal avec le compilateur FPC fait mentir les usines à mono-langages. FPC est proposé sur Linux, FreeBSD, Haiku (BeOS), Mac OS X, DOS, Windows, OS/2 et même pour l'iPhone.

XCode, tour de contrôle de la programmation sur Mac, ne permet qu'un nombre limité de langages de programmation. XCode n'apparaît alors plus comme l'ouverture souhaitable sur les nombreuses possibilités de langages. Il est restreint essentiellement à Objective-C pour renforcer la sécurité des applications dédiées à l'iPhone. Qu'à cela ne tiennent, nous allons voir que nous pourrons utiliser de nombreux autres langages sur Mac.

Comme souvent, la richesse vient de la diversité. Il s'agit ici moins ici de choisir un langage de programmation que d'offrir un panorama des langages disponibles sur Mac. La compétence d'un programmeur se jugeant moins sur son expertise d'un seul langage que sa connaissance de multiples langages.

À noter que les outils mettant en oeuvre ces langages sont pour la plupart part inclut dans les outils du développeur Apple ou gratuits disponibles sous forme de logiciels libres.

Mais tournons-nous à présent vers une présentation rapide des langages les plus populaires, mes langages pratiqués puis ceux présents sur Mac avec leur utilisation.

2) Les langages de programmation les plus populaires

(Suivant le classement de "The Transparent Language Popularity Index" d'août 2010 : http://lang-index.sourceforge.net, ils sont présentés avec la date de création, le créateur, le type de langage et le dernier standard en date)

La publication du classement de popularité des langages de programmation sur Internet par le site "The Transparent Language Popularity Index" montre l'entrée récente du langage de programmation Objective-C dans le top 20 à la 12ème place. Cela démontrent un peu plus la conjonction du succès établit de l'iPhone et celui plus récent de l'iPad avec l'exclusivité imposée par Apple pour le langage de programmation Objective-C.

1 *C*
(1972, Dennis Ritchie - américain, langage procédural compilé utilisé en premier pour le système UNIX, ISO/IEC 9899:1999 Cor. 3:2007(E))

2 Java
(1995, James Gosling - canadien, langage orienté objet dérivé du C compilé pour la machine virtuelle Java multi-plateforme, standard Sun 6.0 2006)

3 PHP
(1994, Rasmus Lerdorf - danois, langage orienté objet interprété dérivé du C, standard PHP Group 5.3.2 2010)

4 *C++*
(1983, Bjarne Stroustrup - danois, langage orienté objet dérivé du C compilé, ISO/CEI 14882:2003)

5 (Visual) Basic
(1963, John George Kemeny - hongrois, langage non structuré interprété interactif puis structuré compilé, dérivé du Fortran, orienté objet dans sa version Visual, multi-dialectes minimal ISO/CEI 6373:1984 complet ISO/CEI 10279:1991/Amd 1:1994)

6 Python
(1990, Guido Van Rossum - néerlandais, langage orienté objet interprété dérivé du C mais aussi de Modula-3, standard Python Software Foundation 3.1.2 2010)

7 *C#*
(2001, Anders Hejlsberg - danois, langage orienté objet dérivé du C compilé pour la machine virtuelle .NET, ISO/IEC 23270:2006)

8 Perl
(1987, Larry Wall - américain, langage procédural interprété dérivé du C, standard The Perl Foundation 5.12 2010)

9 Delphi (Object Pascal)
(1995, Anders Hejlsberg - danois, langage orienté objet compilé dérivé du Pascal, standard Embarcadero -ex Borland- 2010)

...
11 Ruby
(1995, Yukihiro Matsumoto - japonais, langage orienté objet interprété dérivé de Perl et Smalltalk, standard ruby-lang.org 1.9.1 2009)
12 Objective-C
(1986, Brad Cox - américain, langage orienté objet dérivé du C mais aussi de Smalltalk, standard Apple 2.0 2007)
...
14 Pascal
(1970, Niklaus Wirth - suisse, langage procédural compilé, multi-dialectes classique ISO 7185:1990 étendu à ISO/IEC 10206:1991)
...
18 Ada
(1983, Jean Ichbiah - français, langage orienté objet compilé dérivé du Pascal, ISO/IEC 8652:1995/Amd 1:2007)

3) Mes langages de programmation

Un langage de programmation est analogue aux briques Lego, avec quelques instructions, nous créons des infinités de possibilités de programmes au bout de nos doigts. Les langages de programmations sont très nombreux (plus de 100 ayant une notoriété publique), en fait, chacun de nous peut s'en inventer un à soit. Leur complexité est aussi très différentes. Aux premières heures de l'informatique, qui n'avait pas encore ce nom, les langages étaient rudimentaires proches de la machine (assembleur sur cartes perforées), puis ils ont évolués de génération en génération, réservés d'abord aux experts (Fortran, Cobol, C sur bandes magnétiques), ils se démocratisent soudainement avec l'apparition de la micro-informatique (Basic, Logo sur cassette audio) puis redeviennent plus complexe (Ada, C++, Java sur disques durs) avec l'apparition des interfaces utilisateurs graphiques multi-fenêtres.

Je regrette le temps où tout un chacun créait sur son ordinateur un petit programme basic très simplement...

Ma liste par ordre chronologique d'apprentissage (avec le classement du site Lang-Index) :

5 Basic
NC Assembleur 6502, 6800, Z80, 680x0, x86
14 Pascal
9 Delphi (Object Pascal)
25 Fortran
45 Bourne again shell

1 C
18 Ada
2 Java
75 AppleScript
12 Objective-C

Comme quoi la popularité d'un langage n'a jamais été pour moi un critère de choix. J'utilise aussi ponctuellement plusieurs autres langage mais sans en avoir fait l'apprentissage complet.

4) Les langages de programmation sur Mac

Voici les langages présents nativement sur Mac (leur version dépendra donc de celle de Mac OS X) ou pour quelques autres nécessitant une installation spécifique. Cependant, ils ont tous besoin de l'installation des outils de développement XCode (voir les détails dans AVM expert 5). Néanmoins, je ne développerai pas l'utilisation de XCode pour plutôt démontrer la possibilité d'utilisation de ces langages avec le Terminal plus universel.

De plus, les logiciels utilisés sont libres d'utilisation, ces essais ne vont donc pas nous coûter un sous. Allons-y !
(De façon usuelle le signe $ représente l'invite de commande du Terminal, ^D représente la combinaison de touches <ctrl>-D)

a) Utilisation des langages C, C++, Objective-C

Ces trois là ont besoin d'utilitaires de compilation pour créer un fichier exécutable contenant les instructions correspondantes au microprocesseur du Mac.
La version installée :
$ gcc -v
Using built-in specs.
Target: powerpc-apple-darwin8
...
Thread model: posix
gcc version 4.0.1 (Apple Computer, Inc. build 5370)
Pour aller plus loin, le site officiel : http://gcc.gnu.org

a.1) Utilisation du langage C

C est le langage historique de la programmation du système UNIX.

Un exemple simple :

```
$ cat > hello.c
#include <stdio.h>
int main(void)
{
 printf("Bonjour avec C %d.%d.%d !\n", __GNUC__, __GNUC_MINOR__,
__GNUC_PATCHLEVEL__);
 return 0;
}
^D
$ gcc hello.c
$ ./a.out
Bonjour avec C 4.0.1 !
```

a.2) Utilisation du langage C++

C++ doit surtout son succès avec l'utilisation des interfaces utilisateurs graphiques.

Un exemple simple :

```
$ cat > hello.cpp
#include <iostream>
int main()
{
 std::cout << "Bonjour avec C++ !\n";
}
^D
$ g++ hello.cpp
$ ./a.out
Bonjour avec C++ !
```

a.3) Utilisation du langage Objective-C

Objective-C est le langage naturel de la programmation Cocoa sur Mac.

Un exemple simple :

```
$ cat > hello.m
#import <Foundation/Foundation.h>
int main (int argc, const char * argv[])
{
  NSLog(@"Bonjour avec Objective-C !");
  return 0;
}
^D
$ gcc -framework foundation hello.m
$ ./a.out
2010-07-28 16:19:04.305 a.out[3145] Bonjour avec Objective-C !
```

b) Utilisation du langage Java

Java doit son succès à l'extraordinaire importance de sa bibliothèque couvrant à peu près tout les domaines de l'informatique surtout Internet et à sa machine virtuelle permettant la création d'applets, petites applications s'exécutant dans un navigateur Web.

En américain Java a pris la signification de café depuis l'importation du café depuis l'ile de Java à partir du milieu du 19ième siècle

Il s'agit donc d'un langage compilé dans un format spécifique (ByteCode) qui est exécuté par la Java Virtual Machine (JVM) dédiée à chaque platforme procurant ainsi une universalité requise par Internet.

La version installée :

```
$ javac -version
javac 1.5.0_19
```

...

Un exemple simple :

```
$ cat > hello.java
public class hello
{
    public static void main(String[ ] args)
    {
        System.out.println( "Bonjour avec Java !" );
    }
}
^D
$ java hello
Bonjour avec Java !
```

Pour aller plus loin, le site officiel : http://www.java.com/fr

c) Utilisation des langages interprétés PHP, Perl, Python, Shell et Ruby

Ils ont besoin d'un interpréteur adapté au Mac mais celui-ci présente un résultat identique quelque soit la plateforme Windows, Mac ou Linux. En effet, le programme ou script est décodé et exécuté à la volée par l'interpréteur.

L'avantage est donc l'universalité mais l'inconvénient est leur relative lenteur par rapport aux langages compilés.

c.1) Utilisation du langage PHP

Conçu dès le départ pour le Web, son succès ne se dément pas utilisant ses liens forts avec le serveur Web Apache et le moteur de base de données MySQL.

La trompe et la tête forme le premier P, les jambes le H
et l'imagination fait le reste pour le deuxième P

La version installée :

```
$ php -v
PHP 4.4.9 (cli) (built: Sep 17 2008 16:31:15)
Copyright (c) 1997-2008 The PHP Group
Zend Engine v1.3.0, Copyright (c) 1998-2004 Zend Technologies
```

Un exemple simple :

```
$ cat > hello.php
#!/bin/php
<?php
echo "Bonjour avec PHP ", phpversion(), " !\n";
?>
^D
$ php hello.php
Bonjour avec PHP 4.4.9 !
```

L'utilisation pour le Web est très simple puisque le code PHP s'insère dans le code HTML pourvu que le serveur de votre site supporte PHP.
Vous pouvez également utiliser PHP en local sur votre Mac en activant le partage Web personnel dans les préférences Systèmes->Partage->Services et en modifiant le fichier ~/Sites/index.html.
Pour aller plus loin, le site officiel : http://www.php.net

c.2) Utilisation du langage Perl

Successeur en titre des utilitaires historiques d'Unix comme sed, awk, grep..., son succès provient de l'immense bibliothèque de programmes CPAN (http://search.cpan.org).

Perl le langage oignon où chacun y trouve une pelure à son goût

La version installée :

```
$ perl -v
This is perl, v5.8.6 built for darwin-thread-multi-2level
(with 5 registered patches, see perl -V for more detail)
Copyright 1987-2004, Larry Wall
```

Un exemple simple :

```
$ cat > hello.pl
#!/usr/bin/perl
printf "Bonjour avec Perl %vd !\n", $^V;
^D
$ perl hello.pl
Bonjour avec Perl 5.8.6 !
```

Pour aller plus loin, le site officiel : http://www.perl.org

c.3) Utilisation du langage Python

Langage multi-emploi, outre son utilisation basique pour automatiser les tâches d'un logiciel, il est utilisé pour l'enseignement de la programmation et les applications scientifiques.

Les informaticiens peuvent être aussi de grands fans des Monty Python

La version installée :

```
$ python -V
Python 2.3.5
```

Un exemple simple :

```
$ cat > hello.py
#!/usr/bin/python
import sys
print "Bonjour avec Python", sys.version
^D
$ python hello.py
$ /usr/bin/python hello.py
Bonjour avec Python 2.3.5 (#1, Jan 12 2009, 14:43:55)
[GCC 3.3 20030304 (Apple Computer, Inc. build 1819)]
```

Python peut aussi s'utiliser de façon interactive comme une calculatrice évoluée par exemple :

```
$ python
Python 2.3.5 (#1, Jan 12 2009, 14:43:55)
[GCC 3.3 20030304 (Apple Computer, Inc. build 1819)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+2
4
>>> x=9
>>> y=5
>>> x+y
14
>>> _*2
28
>>> ^D
```

Pour aller plus loin, le site officiel : http://www.python.org

c.4) Utilisation du langage Shell (logo image02)

(1977, Stephen Bourne - britanique, langage procédural interprété, IEEE POSIX P1003.2/ISO 9945.2)

Étant le langage naturel du Terminal, il ne se présente plus.

Bourne-again shell jeu de mots avec le shell langage de commandes historique sur Unix créé par Steve Bourne

La version installée :

```
$ sh --version
GNU bash, version 2.05b.0(1)-release (powerpc-apple-darwin8.0)
Copyright (C) 2002 Free Software Foundation, Inc.
```

Un exemple simple :
$ cat > hello.sh
```
#!/bin/sh
echo "Bonjour avec Shell " $BASH_VERSION
```
^D
$ sh hello.sh
Bonjour avec Shell 2.05b.0(1)-release
Pour aller plus loin, le site officiel : http://tiswww.case.edu/php/chet/bash/bashtop.html

c.5) Utilisation du langage Ruby

Selon la volonté de son créateur, il se définit comme plus fort que Perl et Python.

Jeu de mots avec le langage Perl,
les informaticiens sont de grands enfants

La version installée :
$ ruby -v
ruby 1.8.2 (2004-12-25) [powerpc-darwin8.0]
Un exemple simple :
$ cat > hello.rb
```
#!/usr/bin/ruby
print "Bonjour avec Ruby ", RUBY_VERSION, " !\n"
```
^D
$ ruby hello.rb
Bonjour avec Ruby 1.8.2 !
Ruby peut aussi s'utiliser de façon interactive comme une calculatrice évoluée par exemple :
$ irb
irb(main):001:0> 2+2
=> 4
irb(main):002:0> x=9
=> 9
irb(main):003:0> y=5
=> 5
irb(main):004:0> x+y
=> 14
irb(main):005:0> exit
Pour aller plus loin, le site officiel : http://www.ruby-lang.org/fr

d) Utilisation du langage AppleScript

(1993, Apple, langage procedural interprété, standard Apple 2007)

Est au Mac ce que le Shell est à Unix, AppleScrit est lui aussi un véritable langage de programmation.

Un exemple simple :

```
$ cat > hello.scpt
return "Bonjour avec AppleScript " & (version of AppleScript as string) &
" !"
^D
$ osascript hello.scpt
Bonjour avec AppleScript 1.10.7 !
```

Pour aller plus loin, le site officiel : http://developer.apple.com/applescript

À partir d'ici, les langages qui suivent ne sont pas présents nativement avec Mac OS X. Voir la ligne "pour aller plus loin" pour les télécharger et les installer.

e) Utilisation du langage Basic

Langage de mes premiers programmes, le Basic des origines a été multiplié jusqu'à perdre son âme (s'il pouvait en avoir une).

Un exemple simple :

```
$ cat > hello.bas
print "Bonjour avec Basic !"
^D
$ bwbasic hello.bas
Bywater BASIC Interpreter/Shell, version 2.50
Copyright (c) 1993, Ted A. Campbell
Copyright (c) 1995-1997, Jon B. Volkoff
 Bonjour avec Basic !
```

Basic peut aussi s'utiliser de façon interactive comme une calculatrice évoluée par exemple :

```
bwBASIC: print 3 * 4
 12
bwBASIC: y=4
bwBASIC: print y*8
 32
bwBASIC: quit
```

Pour aller plus loin, le site officiel : http://sourceforge.net/projects/bwbasic

f) Utilisation du langage Pascal Objet

Toujours très utilisés, lui aussi perd peu à peu son âme avec Delphi qui lui greffe des extensions en tout genre et Freepascal qui lui emboite le pas.

Hommage à Blaise Pascal par Niklaus Wirth génial inventeur de ce langage pour servir à l'enseignement de la programmation

La version installée :

```
$ $ fpc -i
Free Pascal Compiler version 2.4.0
Compiler Date      : 2009/12/20
Compiler CPU Target: powerpc
...
```

Un exemple simple :

```
$ cat > hello.pas
program Hello;
begin
writeln('Bonjour avec Pascal !');
end.
^D
$ fpc hello.pas
$ ./hello
Bonjour avec Pascal !
```

Pour aller plus loin, le site officiel : http://www.freepascal.org

g) Utilisation du langage Ada

Bien qu'initié au début des années 1980 par l'armée américaine qui croulait sous les langages les plus exotiques et bien que présentant une rigueur rebutante au premier abord, Ada révèle très vite ses atouts multiples qui en font un langage de premier choix pour s'initier ou programmer professionnellement.

Hommage à Lady Ada Lovelace considérée comme ayant écrit le premier programme informatique dans les années 1840

La version installée :

```
$ gcc -v
...
gcc version 4.3.4 20090511 for GNAT GPL 2009 (20090511) (GCC)
```

Un exemple simple :

```
$ cat > hello.adb
with Text_IO; use Text_IO;
procedure Hello is
begin
Put_Line("Bonjour avec Ada !");
end;
^D
$ gnatmake hello
$ ./hello
Bonjour avec Ada !
```

Pour aller plus loin, le site officiel : http://libre.adacore.com/libre

h) Utilisation du langage Fortran

(1956, John Backus - américain, langage procédural compilé, ISO/IEC 1539-1:2004)

Bien que plus ancien langage procédural, il est encore très utilisé dans les applications scientifiques.

La version installée :

```
$ gfortran -v
Using built-in specs.
Target: powerpc-apple-darwin8
Configured with: /Users/drew/Developer/Compiler/gcc-head/configure --disable-checking --disable-nls --enable-static --prefix=/usr/local/ada-4.3 --host=powerpc-apple-darwin8 --target=powerpc-apple-darwin8 --build=powerpc-apple-darwin8 --enable-languages=c,ada,c++,fortran,objc,obj-c++
Thread model: posix
gcc version 4.3.0 20080203 (experimental) [trunk revision 132082] (GCC)
```

Un exemple simple :

```
$ cat > hello.for
      PROGRAM Hello
      PRINT *, 'Bonjour avec Fortran !'
      END
^D
$ gfortran hello.for
$ ./a.out
 Bonjour avec Fortran !
```

Pour aller plus loin, le site officiel : http://gcc.gnu.org/fortran

i) Utilisation du langage Logo

(1966, Wally Feurzeig et Seymour Papert - américain, langage fonctionnel interprété dérivé du Lisp, multi-dialectes)

Anciennement très populaire dans l'éducation, Logo et sa célèbre tortue n'est pas qu'un langage pour les enfants, Logo est aussi un vrai langage de programmation.

XLogo a l'avantage d'être multilingue

Un exemple simple :
ecris [Bonjour avec Logo !]
Bonjour avec Logo !
Pour aller plus loin, le site officiel : http://xlogo.tuxfamily.org/fr/index-fr.html

5) Conclusion

La programmation est une passion, avec le Mac, je suis vraiment comblé car, comme tout système basé sur Unix, il contient une multitude de possibilités.

Mon langage de programmation favori est Ada (voir sur blady.pagesperso-orange.fr). Ce n'est pas le plus populaire aujourd'hui mais il est rempli de qualités. Sa structure, par exemple, permet d'identifier très tôt 80% des erreurs, dès la compilation avant même la première exécution du programme. Comme l'écrivait Jean-Pierre Rosen en conclusion de son livre Méthodes de génie logiciel avec Ada (voir sur wikibooks.fr) :
"C++ ressemble à un énorme gâteau à la crème, ruisselant de sucreries; Ada ressemble plus à un poisson poché / pommes vapeur. Les questions intéressantes sont 1) quel est le meilleur pour votre santé et 2) qu'est-ce que les gens ont tendance à choisir spontanément ?".

Mes espérances seront néanmoins comblées si ce tour d'horizon vous aura donner envie d'essayer la programmation sur Mac quelque soit le langage utilisé.

Pascal Pignard, septembre 2010.