LinuxFocus article number 277
http://linuxfocus.org

# Programmer en BASIC sous Unix


par John Perr
<johnperr/at/Linuxfocus.org>

*L´auteur:*

Utilisateur Linux depuis 1994; il est un des éditeurs français de la revue LinuxFocus.

*Résumé*:

Développer en BASIC sous Linux ou un autre système Unix ? Pourquoi pas ? Il existe plusieurs solutions libres qui permettent d'utiliser le langage BASIC pour développer des applications interprétées ou même compilées.

## Introduction

Bien qu'il soit arrivé après d'autres langages sur la scène informatique, le BASIC s'est vite imposé sur de nombreux système non-Unix comme le remplaçant des langages de script que les Unix possèdent nativement. C'est d'ailleurs, sûrement une des principales raisons pour lesquelles ce langage reste peu utilisé par les unixiens. Comme les divers langages de script, le BASIC est majoritairement interprété et possède une syntaxe relativement simple, pas de données typées, si ce n'est une différence entre chaînes et nombres. Historiquement, le nom du langage vient justement du fait qu'il est simple et permet d'initier facilement les étudiants à la programmation.

Malheureusement, l'absence de standardisation a conduit à l'apparition de nombreuses versions toutes incompatibles les unes avec les autres. On peut même dire qu'il en existe autant que d'interpréteurs ce qui rend BASIC tout sauf portable.
Malgré ces inconvénients et de nombreux autres que les "vrais programmeurs" ne manquerons pas de rappeler, BASIC reste une option à considérer pour développer des petits programmes rapidement, surtout depuis l'apparition il y a déjà plusieurs années sous windows de versions munies d'environnement de développement (IDE en anglais) qui permettent la conception d'interfaces graphiques en quelques clics de souris. Ces versions appelées "Visual" on été de plus adoptées comme langages macro dans les applications bureautique ce qui en accroît la démocratisation et l'utilisation par toute une population qui n'aurait jamais abordé autrement la programmation.
Nous passerons ici en revue les différentes versions disponibles sous Linux, ou du moins les plus

populaires, et nous en tenterons un comparatif.

# Un peu d’histoire

*Extrait et traduit de l’historique d’Andrea M. Marconi, consultable avec la documentation de kbasic*

## La naissance du BASIC

Le langage BASIC (en anglais Beginners All-Purpose Symbolic Instruction Code) est né en 1964 au collège de Dartmouth au New Hampshire (USA), où il a été développé par John G. Kemeney (1926-93) et Thomas E. Kurtz (1928-). Kemeney, a travaillé sur le projet Manhattan Project (1945) puis, plus tard (1948-49) en tant qu’assistant d’ Albert Einstein. c’est là qu’il rencontre Kurtz à Dartmouth en 1956.

Ils commencent tout deux à travailler sur un nouveau langage de programmation simplifié, et après être passés par des versions appelées Darsimco et DOPE, ils s’orientent vers un langage répondant aux spécifications suivantes:

1. D’utilisation générale
2. Facile à utiliser
3. Extensible
4. Interactif
5. Avec un système d’aide au débogage
6. Efficace
7. Indépendant du matériel
8. Indépendant du système d’exploitation

Pour cela, ils partent de FORTRAN et ALGOL. la version finale est appelée "Dartmouth BASIC" et comporte 14 instructions. Le "Dartmouth BASIC" était compilé et plutôt rapide pour l’époque.

Les efforts de Kemeney et Kurtz sont récompensés le 1er mai 1964 à 4 heure du matin, deux programme BASIC tournent simultanément sur l’unité centrale General Electric 225 du Collège de Dartmouth.

## La croissance de BASIC

Kemeney et Kurtz ne prirent aucun brevet et ne protégèrent pas leur invention afin de la laisser dans le domaine public. Cela en permit la croissance mais aussi la multiplicité. Parmi les premiers utilisateurs, on compte General Electric qui avait vendu le GE-255 à Dartmouth.

Gordon Eubanks (le président de Symantec) est à l’origine de plusieurs versions de BASIC dont le BASIC-E en 1970. Il utilisait un pseudo code comme Java aujourd’hui. Puis vint CBASIC et beaucoup d’autres versions, ce qui incita en 1974 l’ANSI à établir des standards qui ne furent jamais vraiment

suivis puisque l'utilisation de BASIC était déjà répandue quand les standards apparurent en 1978.

Entre temps (1975),TinyBASIC est créé par Bob Albrecht et Dennis Allison. Il peut tourner avec 2K de mémoire RAM. Le premier BASIC interprété apparaît aussi. Il est développé par William H. Gates III (1955-, plus connu sous le nom de Bill) et Paul Allen (1953-). Les producteurs de matériel informatique introduisent aussi une copie de BASIC sur des ROM de leurs machines. A la fin des années 70, les premiers ordinateurs personnels ont leur version de BASIC:

- Radio Shack Level 1 BASIC (TRS 80)
- Apple Integer BASIC (Apple II, 1977)
- Timex Sinclair 1000 BASIC (Sinclair ZX80, 1980)
- Sinclair ZX81BASIC (Sinclair ZX81, 1981)
- PET BASIC (Commodore PET, 1977)
- Atari BASIC (Atari 400/800, 1978)
- Commodore BASIC (VIC 20 in 1981 & C64 en 1982)
- TI-BASIC (Texas TI-99)
- etc...

### L'évolution du BASIC

Depuis le début des années 80, l'histoire de BASIC est intimement liée à celle des ordinateurs. Deux produits dominent le marché. IBM fournit le BASICA avec PC-DOS, interprété et sur ROM, mais extensible. Son analogue pour MS-DOS est le BASIC Gee-Witz (ou GW).
En 1984 apparaît le compilateur BASIC de Microsoft, suivit de près par de nombreuses versions dont la série QuickBASIC commencée en 1985 et achevée en 1990 par le Microsoft BASIC Professional Development System 7.1

Une nouvelle fois, ce sont les systèmes d'exploitation qui amènent des changements dans le langage avec l'introduction d'interfaces graphiques. Ainsi naît Visual BASIC, conçu pour élaborer des interfaces graphiques (GUI). Visual BASIC prétend être un langage objet, ce que contestent de nombreux programmeurs. On notera cependant qu'une étude récente estime à 90% le taux d'utilisation de Visual BASIC dans les programmes développés pour Windows 9x.

## Les BASIC aujourd'hui

Si l'on tente de recenser le nombre de BASIC disponibles pour Linux, on trouve facilement une bonne demi-douzaine de projets à divers stades d'avancement. Il existe même sur sourceforge une "Basic Foundry" qui établit un classement basé sur le nombre de téléchargements réalisés:

TOP DOWNLOADS sur Sourceforge.net

1. XBasic
2. SmallBASIC

3. wxBasic
4. GNU/Liberty Basic
5. YaBASIC
6. X11-Basic

On apprend aussi que Gnome et KDE on chacun leur projet destiné à remplacer Visual Basic pour les raisons de démocratisation expliquées ci-dessus De plus Richard Stallman lui même, fait un commentaire sur le besoin d'une alternative libre à VB dans une lettre à "the register", le 10 février 2002.
Anthony Liguori (ajl13-at-bellatantic.net), auteur de GLBCC (GNU/Liberty Basic Compiler Collection) le seul projet basic labellisé GNU, réitère cette demande de manière plus précise sur le site GLBCC hébergé par sourceforge.

Passons donc en revue les différents BASIC pour nous faire une idée du "look and feel" de chacune des interfaces et des possibilités du langage:

# XBasic

XBasic est sûrement l'environnement de développement dont l'interface utilisateur est une des plus abouti. Il intègre un IDE, c'est à dire un outil de construction d'interface graphique, un débogueur et un compilateur. Un système de "dot commands" permet aussi de commander au clavier l'interface sans utiliser la souris en tapant des commandes précédées d'un point, directement dans la zone de saisie en haut à gauche de la fenêtre principale:

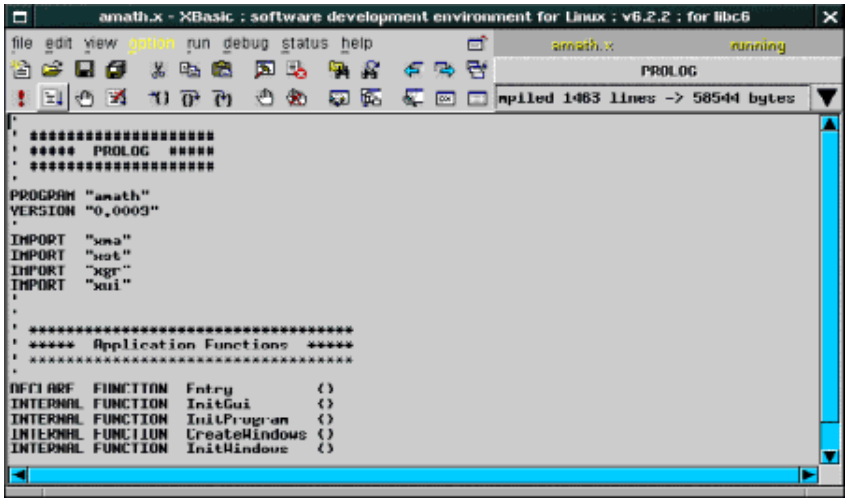


Figure 1a: La fenêtre principale

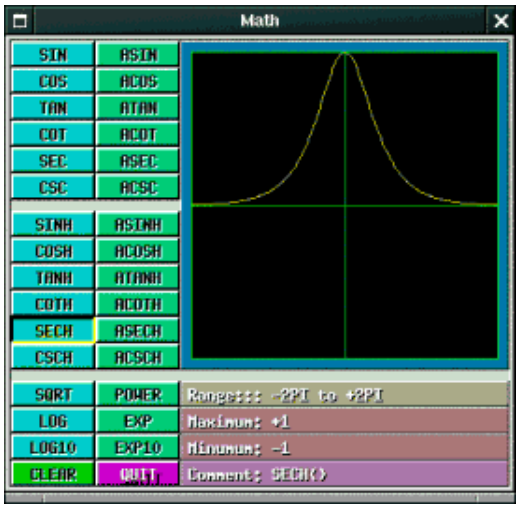

Figure 1b: L'application Math

Coté possibilités, XBasic possède toutes les bibliothèques nécessaires à la programmation d'interfaces graphiques et beaucoup d'extensions. On citera la possibilité d'appeler des fonctions écrites en langage C. On retrouve d'ailleurs beaucoup des possibilités du langage C comme des déclarations de type, des unions de variables ou la création de bibliothèques
Enfin, Xbasic est disponible sous license GPL pour les environnements Windows ou Linux à l'adresse :http://www.xbasic.org

# SmallBASIC

SmallBASIC est un interpréteur en mode texte pour Win32, Linux et PalmOS. La partie développeur est largement documentée afin d'encourager le portage sur d'autres OS. Côté installation l'interpréteur peut être compilé pour plusieurs interfaces:

- SVGALIB
- Frame Buffer
- SDL

L'exécution peut se faire en mode texte ou en mode graphique. L'exemple ci-dessous exécute le programme System_infos.bas:

## Mode console:

```
$ sbasic  System_info.bas
SmallBASIC version 0.8.2, use -h for help
http://smallbasic.sourceforge.net

VMT Initialization...
File: System_info.bas
```

```
Section: Main
PASS1: Line 24; finished
SB-MemMgr: Maximum use of memory: 30KB

PASS2: Node 3/3
Creating byte-code...
Variables  20
Labels     0
Proc/Func  0
Code size  707

System Information

OS:     Unix/Linux version 2.4.18-6mdk (quintela@bi.mandrakesoft.com)
       (gcc version 2.96 20000731 (Mandrake Linux 8.2 2.96-0.76mdk))
       #1 Fri Mar 15 02:59:08 CET 2002 204018
SB:     802
Display 99x58
Colors  16
Font:   1x1

Total free memory: 127728 KB
Stack size: 127728 KB
Largest free memory block: 127728 KB

 * DONE *

SB-MemMgr: Maximum use of memory: 30KB
$
```

## Mode graphique

```
$ sbasic -g  System_info.bas
SmallBASIC version 0.8.2, use -h for help
http://smallbasic.sourceforge.net


VMT Initialization...
File: System_info.bas
Section: Main
PASS1: Line 24; finished
SB-MemMgr: Maximum use of memory: 30KB

PASS2: Node 3/3
Creating byte-code...
Variables  20
Labels     0
Proc/Func  0
Code size  707

SB-MemMgr: Maximum use of memory: 30KB
$
```

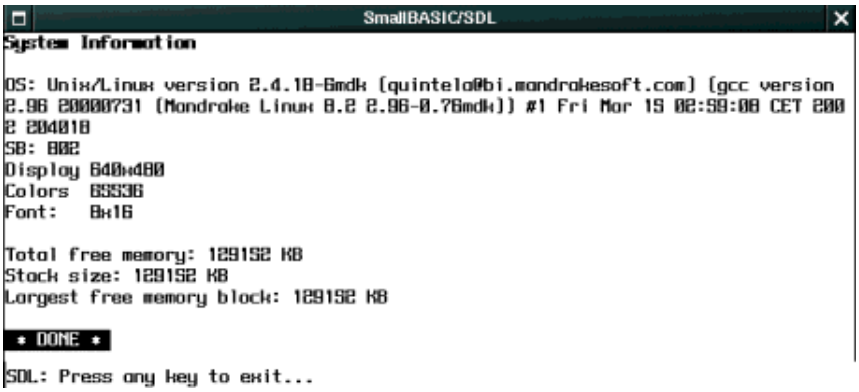


Figure 2: SmallBASIC. Mode graphique SDL.

Le langage de SmallBASIC est assez simple et proche des fonctions standard que l'on peut attendre d'un BASIC. Même les fonctions graphiques ne sont pas innovantes, on retrouve les classiques RECTANGLE et CIRCLE qui ont l'avantage de s'exécuter indifféremment sur n'importe lequel des OS suscités. Les variables ne sont pas typées.
Par contre, SmallBASIC est compatible avec les anciens TINYBasic et QuickBasic et il est très bien intégré à PalmOS.
Ce logiciel est téléchargeable sur http://smallbasic.sourceforge.net

# wxBasic

Le wxBasic se décrit comme ayant les possibilités de Quick Basic associées à des spécificités Unix comme les tableaux associatifs de awk. C'est un interpréteur de petite taille, il tient sur une disquette. La documentation qui l'accompagne est complète sous forme d'un manuel au format pdf de 138 pages. Le langage comporte une bibliothèque graphique qui permet d'écrire des programmes pour X Window ou Windows. Ce langage se rapproche fortement des langages objet comme le C++, tout au moins pour la construction de l'interface graphique. Par contre, les interfaces graphiques doivent être construites "à la main". Aucune interface de développement intégrée ne semble exister pour ce BASIC.

```
// My first wxBasic demo...
option explicit

// create the window
dim frame=new wxFrame(0,-1,"wxBasic App",wxPoint(10,10),wxSize(300,200))
frame.Centre()

// place a panel in the window
dim panel = new wxPanel(frame, -1)

// add a status bar
dim sBar = frame.CreateStatusBar( 1 )
sBar.SetStatusText("wxBasic Frame Demo")

// attach a menubar to the window
dim mBar = new wxMenuBar()
frame.SetMenuBar(mBar)

// build the "File" dropdown menu
```

```
dim mFile = new wxMenu()
mBar.Append(mFile,"&File")

// populate it
mFile.Append(wxID_NEW, "&New", "Creates a new file")
mFile.Append(wxID_OPEN, "&Open", "Loads an existing file from disk")
mFile.Append(wxID_SAVE, "&Save", "Saves current file")
mFile.Append(wxID_SAVEAS, "Save &As", "Saves current file with new name")
mFile.AppendSeparator()
mFile.Append(wxID_EXIT, "&Exit", "Exit Application")

// build the "Edit" dropdown menu
etc.....
```

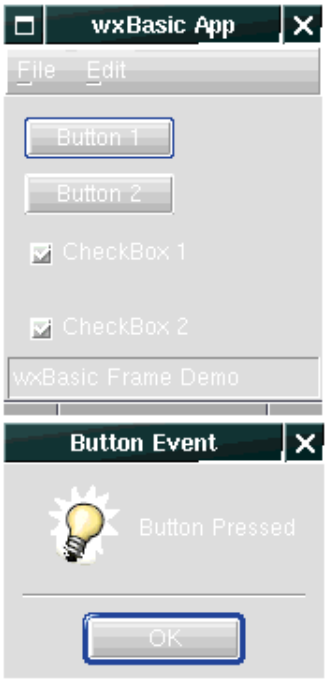


Figure 3: wxBasic: un des programmes de demo

site web: http://wxbasic.sourceforge.net

# GNU/Liberty Basic

Aussi appelé GLBCC, pour GNU Liberty Basic Compiler Collection, c'est uniquement un compilateur ou plutôt une interface au compilateur C GNU GCC qui convertit le Basic en langage C un peu comme font certains compilateurs C++ qui transforment le C++ en C avant compilation. On l'aura compris d'après le titre, ce compilateur est dédié à un langage 100% compatible Liberty Basic. Le Liberty Basic est un des BASIC apparus dans les années 90 sur les plates-formes Windows et qui a connu un relatif succès du fait de sa diffusion libre (d'où son nom). Pour en savoir plus à son propos, le site lui est dédié et vante de nombreuses qualités. Ce langage n'est pas libre, mais une version gratuite de Liberty BAsic pour Windows est téléchargeable sur le site.

Le compilateur GLBCC est disponible pour les environnement Windows et Linux et il est capable de produire des exécutables autonomes qui sont annoncés comme aussi rapides que ceux programmés avec n'importe quel autre langage de programmation. Les auteurs clament aussi assez fort que du code Liberty Basic compilé avec GLBCC est capable de ridiculiser VisualBasic en terme de vitesse.

L'installation est assez simple sous Linux et ne requiert que les classiques dé-compactage avec "tar" et installation avec "make install".
En mode normal, le programme s'utilise en ligne de commande et un "glbcc hello.bas" produira l'exécutable hello comme suit:

```
$ glbcc
/usr/local/bin/lbpp -I/usr/local/lib/glbcc-lib/0.0.7/include -o out.c hello.bas
gcc -g -I/usr/local/lib/glbcc-lib/0.0.7/include 'gnome-config --cflags gnomeui'
    -o hello out.c /usr/local/lib/glbcc-lib/0.0.7/lib/lbcrt0.o
    -L/usr/local/lib/glbcc-lib/0.0.7/lib -lLB
    -lm 'gnome-config --libs gnomeui'
$ ls -l hello*
-rwxr-xr-x  1 john  john  339671 oct 13 21:55 hello
-rw-r--r--  1 john  john      22 avr 14 17:41 hello.bas
$ cat hello.bas
print "Hello, world!"
$ ./hello
Hello, world!
```

S'il est utilisé sans paramètre, glbcc ouvrira une fenêtre de dialogue graphique et demandera à l'utilisateur le nom d'un fichier basic puis éventuellement le nom de l'exécutable à produire. par défaut, le nom du programme entré servira en sortie, additionné d'un .exe sous windows et de rien sous Linux:

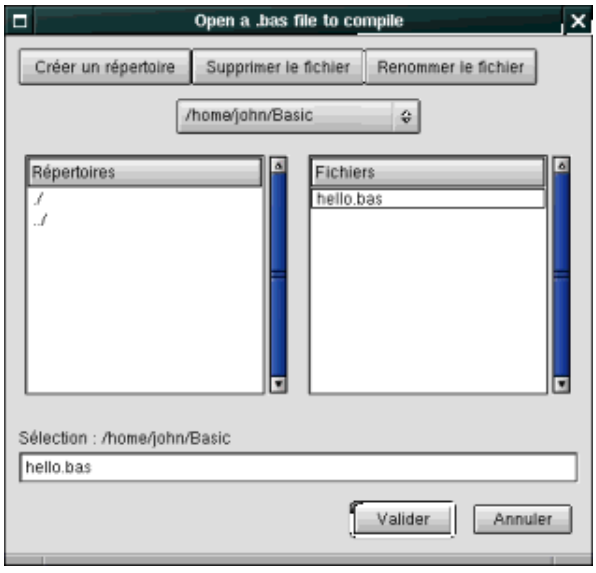

Figure 4: GNU/Liberty Basic

Coté langage, ce basic est complet et l'interface graphique est accessible au travers de la bibliothèque GTK. Chose amusante, glbcc est lui même écrit en Basic.

## YaBASIC

*Yet another Basic for Unix and Windows*

Bien qu'il se comporte comme un interpréteur, yabasic n'en est pas un. C'est plutôt un compilateur: en lui donnant un code source à exécuter, il le compile, le transformant ainsi en code machine puis l'exécute immédiatement. yabasic s'utilise en ligne de commande. Avec un nom de fichier comme argument, il exécute ce fichier comme décrit ci-dessus. Sans argument, il entre en mode interpréteur comme dans l'exemple ci-dessous:

```
$ yabasic
Enter your program, type RETURN twice when done.
for i=1 to 10
   print i,"hello, world"
next i

1hello, world
2hello, world
3hello, world
4hello, world
5hello, world
6hello, world
7hello, world
8hello, world
9hello, world
10hello, world
$
```

Ce fonctionnement permet aussi d'utiliser YaBasic pour écrire des scripts Unix ou même des CGI en plaçant en première ligne du fichier le "#!/bin/yabasic" classique des shells Unix et en déclarant le fichier exécutable.

Coté langage, YaBasic est un BASIC standard (entendez par là, proche de Quick Basic) avec des variables non typées. Il faut seulement faire la différence entre chaînes et nombres grâce à la présence où non du caractère $ en fin de nom de variable. Ici point de programmation objet mais on y trouve quand même la possibilité de faire ses propres bibliothèques, des instructions qui donnent accès aux modes graphiques X11 et Windows selon le système d'exploitation utilisé. A cela s'ajoutent quelques fonctions bien pratiques comme le split() que l'on retrouve dans Perl ou PHP, les boucles FOR, WHILE ou REPEAT UNTIL et le IF ELSE ENDIF, ces derniers étant présents dans le langage BASIC depuis les années 80 quand est apparue la programmation structurée dans Basic.

# X11-Basic

X11-BASIC est un interpréteur de Basic Structuré ayant des capacités graphiques pour X11. Il utilise les fonctionnalités du GFA Basic de l'Atari ST. Il s'agit en fait du portage de GFA BASIC du système GEM/TOS des Atari ST (fin des années 80) vers l'environnement Unix. Il peut être utilisé comme interpréteur ou pour rédiger des scripts ou des CGI. Il existe un pseudo compilateur qui permet de créer des exécutables autonomes (mode statique) ou qu'il faut accompagner de la bibliothèque X11 basic qui pèse environ 200 Ko (mode dynamique). C'est un pseudo compilateur car le fichier résultant ne contient

pas du code machine directement exécuté par le micro processeur mais le langage BASIC sous une forme compressée et accompagné de l'interpréteur. Ce pseudo compilateur est lui même écrit en X11-Basic.
Le langage de ce basic est très riche, structuré et avec des variables typées (entiers, flottants, textes, tableaux, booléens). Il existe des instructions pour accéder à la mémoire comme le malloc() du C ou bien des instructions de multiplication de matrices applicables à des tableaux.
Les instructions graphiques sont restées les mêmes depuis le GFA Basic des Atari ST mais ont maintenant un résultat similaire sous X. L'instruction MENU crée un menu dans une fenêtre graphique par exemple.

Cet interpréteur est accompagné d'une documentation ainsi que de nombreux exemples malheureusement encore en allemand par endroit. Seul ombre au tableau, le portage n'est pas complètement terminé et des bogues sont susceptibles de surgir même avec les exemples fournis. On peut considérer que cet interpréteur se comporte encore beaucoup trop comme une version beta pour être facilement utilisable par la catégorie d'utilisateurs que vise le langage BASIC.

```
$ xbasic
************************************************************
*               xbasic                V. 1.07               *
*                    by Markus Hoffmann 1997-2002 (c)      *
*                                                          *
* version date:           Wed Sep 25 10:26:29 CEST 2002    *
* library V. 1.07 date:   Wed Sep 25 10:26:29 CEST 2002    *
************************************************************
```

X11 Basic: l'accueil de l'interpréteur

Site web: http://www-cip.physik.uni-bonn.de/~hoffmann/X11-Basic/

# HBasic

Voici enfin un BASIC qui fait bonne impression dès le début, tant par ses possibilités, que la qualité de sa documentation qui pèse tout de même 7.7 Mo. L'installation nécessite Qt-3.* qu'il faut charger sur le site de Trolltech (http://www.troll.no) si vous n'avez pas installé une distribution de Linux très récente. Il s'agit d'un environnement de développement vraiment complet qui malgré sa jeunesse (version 0.8) intègre déjà toutes les fonctionnalités dont un développeur peut rêver:

- Interface de développement intégrée, créateur d'interface graphique et éditeur de propriétés.
- Chargement et utilisation de paquetages prédéfinis pour adapter les formulaires et objets graphiques dans les programmes.
- Éditeur de code source avec coloration syntaxique, complétion et repliement des modules de code.
- Possède un interpréteur pour éviter la lenteur du cycle de compilation en phase de tests.
- Le compilateur est intégré et génère de véritables exécutables.
- Un compilateur de code .NET peut être exécuté en environnement .NET.
- Débogueur: Inclut des points d'arrêt, une fenêtre de visualisation du contenu des variables en cours d'exécution ou bien en déplaçant la souris sur le nom de la variable dans l'éditeur de code.
- Possibilité de créer des objets en C++ pour étendre les programmes HBasic ou l'interface de développement en cours d'exécution.

- Langage orienté objet pour les classes définies dans le code source HBasic ou des composants chargés comme des paquetages.
- Outils de gestion de bases de données intégrées qui permettent d'accéder aux données directement avec l'IDE (environnement de développement intégré) ou par programmation.
- Création et utilisation d'instance des classes QT
- Support .NET: permet d'utiliser la gamme complète des bibliothèques .NET pour accéder aux méthodes, propriétés ou champs des bibliothèques .NET
- Édition et compilation de programme C# depuis l'IDE HBasic.
- Feuilles de calcul et graphes (sont encore à l'état de version alpha).

Les auteurs de HBasic insistent quand même en disant: *"La version en cours de HBasic n'est pas encore assez stable pour être utilisée par des développeurs BASIC. Il leur faudra attendre la publication de la première version stable 1.0"..*

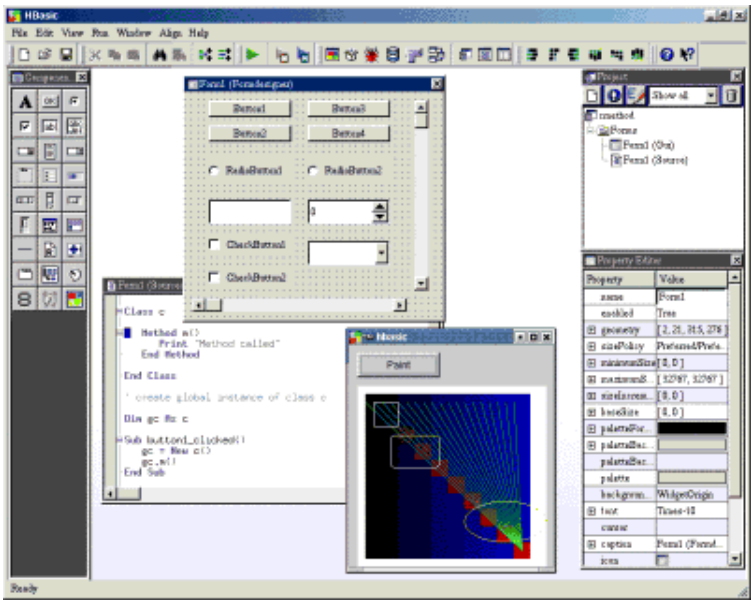

Figure 5:Hbasic

# GNOME Basic

GNOME Basic est un projet qui vise à être le remplaçant compatible à 100% des Visual Basic, VBA, VBScript, et autres langages associés. A l'origine il se dédiait à rendre les macros VBA exécutables dans gnumeric, mais à cause de dérapage des objectifs il exécute quelques formulaires, examine la plupart du code VB et exécute de simples ASPs. Avec une meilleure intégration de Bonobo, un environnement entièrement MS compatible devient réalisable
Pour l'instant, Gnome Basic est juste une tentative embryonnaire de fournir des fonctionnalités compatibles avec VB pour le projet GNOME, en particulier pour les application de type bureautique (VBA).
Le projet est toujours dans un état "pré-alpha" et doit être réservé aux développeurs du projet GNOME à ce stade.

Page d'accueil: http://www.gnome.org/gb/

# KBasic

KBasic est l'autre tentative de développer un BASIC compatible à Visual Basic. Les développeurs espèrent distribuer la première version 1.0 stable à l'été 2003. Il n'existe pour l'instant qu'une version instable réservée aux développements. A terme il semble que KBasic utilisera l'environnement de Kdevelop.

Voici un exemple de ce que donne actuellement la version téléchargeable du noyau dur du KBasic:

```
$ kbasic1 --help
Usage: kbasic1 [OPTION]... FILE
      --help                  display this help and exit
  -V, --version               print version information and exit
  -c, --copyright             print copyright information and exit
$ kbasic1 -V

KBasic version 0.8

Copyright (C) 2000, 2001, 2002 The KBasic Team
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE,
to the extent permitted by law.

Visit us on http://www.kbasic.org/

$kbasic1 hello.bas
-- scan --

LINE 00001 PRINT STRING "Hello, world!" EOL
LINE 00002 END

-- parse --

STRING in line 1
EOL in line 1
parsed PRINT

parsed END


-- interpret --

0000:  KBASIC   KBASIC
0006:  VER      1.000000
0011:  OPTION R OFF
0013:  JMP      36
0036:  DIM_STAT "Hello, world!", 1
0055:  JMP      18
0018:  PUSHS    1 , " Hello, world! "
0023:  PRINT
Hello, world!
0034:  END      execution ended
```

Comme on peut le voir, kbasic fonctionne mais est loin d'être opérationnel, les messages de debogage

de l’interpréteur lui même ainsi que le code assembleur généré sont inévitables pour l’instant.

Site: http://www.kbasic.org

# Conclusion

Ce tour d’horizon des différents interpréteurs et compilateurs BASIC nous apprend plusieurs choses. Tout d’abord, une demi surprise, BASIC, ce langage tant décrié par les développeurs des autres langages, n’est pas mort, même sur les sytèmes Unix où son utilisation reste pourtant toujours très discrète à cause de concurrents sérieux (perl, python, Tcl/Tk,php....) qui fonctionnent eux aussi en mode interprété. L’activité que génère ce langage dans le monde des développeurs libres est significative. Le nombre de projets est important et tous ces projets sont très actifs. A n’en pas douter la demande est sûrement forte et ce fait ne peut que favoriser le développement du logiciel libre en facilitant la migration de programmeurs BASIC vers les Unix libres.
Sur les systèmes Windows, VisualBasic prévaut du fait de son intégration dans le système d’exploitation de Microsoft, les asp et les suites bureautiques. Notons tout de même qu’une grande proportion des BASIC que nous avons testés fonctionnent sur les deux plates-formes voire d’autres.
Deuxièmement, BASIC traîne toujours quelques casseroles. L’absence de standard de ce langage a conduit à ces nombreuses versions qui ne sont pas du tout compatibles entre elles. La volonté de plusieurs projets récents comme GNOMEBasic ou KBasic de rejoindre le standard de fait qu’est VisualBasic irait dans le bon sens si VB était un standard libre, ce qu’il n’est évidemment pas. Il vaut mieux dans ce cas parler de monopole...
Pour le développeur BASIC, le choix existe et quelques outils sortent nettement du lot. En tant que système de développement intégré, HBasic est certainement le plus prometteur. En attendant la version stable, XBasic est actuellement le plus abouti. D’un autre coté, smallbasic, yabasic permettent facilement d’écrire des scripts ou des cgi à tous ceux qui viennent de migrer sous Unix et qui hésitent parmi toutes les solutions que ce système propose. GLBCC permet aussi cela avec l’avantage de créer des programmes compilés avec l’excellent GCC. Étant donné l’absence d’IDE et donc d’un débogueur facile d’abord, on ne pourra que difficilement développer et maintenir de gros programmes avec cet outil. Reste wx-Basic qui est assez inclassable mais qui a des atouts, en particulier pour l’écriture d’interfaces graphiques.
Un autre avantage de ces BASIC est la migration facile de programmes d’un système d’exploitation vers l’autre sans recompilation. Ils sont en effet, quasiment tous disponibles pour les environnements Win32 et Unix.
Il reste donc au développeur BASIC la lourde tâche de faire le bon choix parmi tous ces outils en fonction de l’utilisation envisagée.


| | |
|---|---|
| Site Web maintenu par l´équipe d´édition<br>LinuxFocus | Translation information:<br>fr --> -- : John Perr <johnperr/at/Linuxfocus.org> |

2005-01-14, generated by lfparser_pdf version 2.51