

Après l'énorme succès de Turbo Pascal sous MS-DOS et CP/M, Borland s'attaque au marché du Macintosh avec une nouvelle version de son produit vedette. Conserver au système Turbo Pascal ses performances et sa simplicité d'utilisation, tout en lui ménageant une large ouverture sur l'environnement Macintosh, voilà qui pouvait sembler un pari difficile. Le résultat est pourtant très réussi. Depuis les programmes simples en Pascal classique, jusqu'aux applications dans le plus pur style du Mac, en passant par les accessoires de bureau, Turbo Pascal sait tout faire, ou presque, en deux coups de cuillère à pot.



TURBO PASCAL

SI PAR BIEN DES POINTS, LE Turbo Pascal Macintosh se distingue des versions MS-DOS et CP/M, la philosophie générale du système reste la même. Son objectif principal : réduire le plus possible le temps qui sépare la saisie de la dernière ligne du programme de son exécution. Classiquement, lorsqu'on utilise un langage compilé, après avoir tapé et archivé le programme source dans un éditeur, il faut quitter cet éditeur, invoquer le compilateur qui va fabriquer un programme dit « relogeable » ; on appellera alors un éditeur de lien (ou linker) qui en fera un programme exécutable. Si une ou plusieurs erreurs se sont glissées dans votre programme, vous devrez reprendre tout le processus autant de fois que nécessaire pour les débuser et les éliminer.

Pour Macintosh

La multiplication des temps d'attente et des diverses manipulations à effectuer peut devenir horripilante, surtout pour un néophyte. Turbo Pascal réduit au minimum ces désagréments en intégrant sous une seule application éditeur de programme, compilateur et éditeur de liens.

Tout étant résident en mémoire, il n'y a pas de délai dû aux accès disques et à la fermeture ou à l'ouverture des différents fichiers. La rapidité de compilation proprement dite est si

LOGICIEL TESTÉ :

Version de présérie en anglais, avec documentation provisoire en anglais.

impressionnante que, pour des programmes de plusieurs centaines de lignes, tout est bouclé en une dizaine de secondes. Turbo Pascal peut compiler vos programmes en mémoire ou bien sur disque. La compilation sur disque est à peine plus lente que celle en mémoire et génère une application qui est ensuite directement exécutable à partir du Finder. La compilation en mémoire permet de tester les programmes en cours de mise au point. Il est possible d'exécuter le programme compilé en mémoire sans quitter Turbo Pascal. Comme dans la version MS-DOS, en cas d'erreur à l'exécution, le programme est immédiatement recompilé, pour retrouver la position de l'erreur à partir de la valeur du pointeur de programme au moment de l'erreur. Turbo Pascal signale alors dans le programme

source où et pourquoi il y a eu une erreur, facilité en général réservée aux langages comme Basic où l'exécution est suivie pas à pas par un interpréteur. En quelque sorte, Turbo Pascal offre à la fois les avantages d'un langage interprété, c'est-à-dire la facilité et la souplesse de mise au point, et ceux d'un langage compilé, c'est-à-dire un programme objet rapide et indépendant (il n'y a plus besoin de Turbo Pascal pour faire marcher un programme une fois compilé).

Le langage

Il y a plusieurs façons d'employer Turbo Pascal sur le Mac. Dans un premier temps, vous pouvez écrire du Pascal classique ou transférer des programmes depuis une autre machine, sans vous préoccuper de menus, fenêtres et autres gadgets Macintosh. Dans ce cas, le compilateur se charge de tout et simulera pour votre programme un écran 25 x 80 caractères de terminal classique, avec possibilités graphiques. Le Turbo Pascal est également interfacé à la plupart des sous-programmes de la boîte à outils située dans la ROM du Macintosh, grâce auxquels tous les logiciels Mac ont un air de famille. Plusieurs exemples de programmes simples ayant recours à ces nombreuses fonctions du Mac sont inclus sur la disquette Turbo Pascal et vous permettront de mieux comprendre comment créer et déplacer des fenêtres, utiliser les « ascenseurs »... Pour exploiter réellement ces possibilités, il

celles ayant trait aux conversions de type. Les nouveaux noms sont ceux des procédures standards fournies dans la boîte à outils du Mac. Par exemple, la fonction Str() de la version MS-DOS, qui convertit un nombre en chaîne de caractères, est remplacée par RealToString() et IntToString(). Un très gros progrès concerne le traitement des « unités » du Pascal Standard. Une « unité » Pascal, c'est une bibliothèque de sous-programmes utilitaires et de variables déclarées qui est compilée séparément. Turbo Pascal propose au départ 17 bibliothèques standards, dont plus de la moitié contiennent les programmes d'interfaces avec la ROM. Ainsi pour utiliser des fonctions graphiques, vous incluez au début la ligne : uses quickdraw.

Quickdraw est le nom d'une unit standard qui contient toutes les fonctions de dessin de la boîte à outils. Vous pourrez ainsi les appeler comme bon vous semble. L'avantage principal de ces bibliothèques est qu'elles sont déjà compilées. Quand votre programme utilise l'une d'elles, celle-ci n'est donc pas compilée à nouveau, comme le serait un fichier inclus dans votre fichier source. Cela est un facteur supplémentaire de rapidité du système. En plus des fonctions d'interface avec la ROM, les unités standards du Turbo Pascal permettent la gestion du pseudo-terminal 25 x 80 caractères de l'environnement Pascal classique, le graphisme 3D, la synthèse vocale (en anglais) et des calculs en virgule fixe, moins précis, mais bien plus rapides que les

permettent de spécifier ce découpage ainsi que l'utilisation des ensembles de sous-programmes. D'autres directives de compilation permettent d'interfacer votre programme à des procédures extérieures, par exemple, des sous-programmes en assembleur, comme le ferait un éditeur de lien classique. Vous pouvez également insérer du code machine hexadécimal au milieu de votre programme en Pascal. De même, une directive spéciale du compilateur permet de spécifier que l'on compile un accessoire de bureau et non pas une application ordinaire. Vous pourrez ainsi compléter votre menu pomme par des petits utilitaires de votre cru.

Quelques ombres viennent un peu ternir ce tableau. Tout d'abord, si la compilation est très rapide, le code généré n'est pas aussi performant surtout si votre programme fait de nombreux calculs en virgule flottante (3 à 8 fois moins rapides que Turbo Pascal sur IBM PC). Il faut cependant dire que les principaux fautifs sont les sous-programmes de calcul réel du système Macintosh qui travaillent sur des nombres de 80 bits, soit 18 à 19 chiffres significatifs. Une telle précision est souvent exagérée et ralentit considérablement l'exécution. Second problème, toujours lié à l'architecture sous-jacente du Mac : la taille d'une structure (tableau, enregistrement) est limitée à 32 Ko et, de surcroît, l'encombrement total de toutes les variables déclarées ne peut pas non plus excéder 32 Ko (un tableau de réels 100 x 100 occupe déjà 40 Ko). Sur un

```

File Edit Search Format Font Compile Transfer
SimulLig Run %R
To Memory %M
To Disk %K
Check Syntax %V
Find Error %E
Get Info %I
Options...
PLength = 524.0;
Avogadro = 6.02e20;
var
Fragment : array [1..MaxFragment] of st
ConcLin : array [1..MaxFragment] of rea
ConcCycl : array [1..MaxFragment] of rea
ConcTemp : array [1..MaxFragment] of rea
Len : array [1..MaxFragment] of rea
Product : array [1..MaxFragment,1..MaxF
maxIter, iter, lmax, numFragment, numFragIni
l, volume, delta_T:real;
s:string;
function cyclic(l:real):real; (returns cyclization probability of a DNA fragment
according to Levene & Crothers model)
var
x, dummy:real;
begin
x:=1/270.0;
if x < 0.85 then dummy:=0.0
else if x < 2.4 then dummy:=4.914e-3*x*x*x-4.9328e-2*x*x+0.14438*x-8.763899e-
else if x < 3.0 then dummy:=1.1354e-2*x*x*x-9.2965e-2*x*x+0.24288*x-0.16
else if x < 7.0 then dummy:=-1.7449e-4*x*x*x+3.1504e-3*x*x-2.2554e-
else begin
dummy:=l*(LLength*Length+PLength*PLength)/(LLength*(l+PL

```

La compilation d'un programme Turbo Pascal s'effectue directement depuis l'éditeur.

```

File Edit Search Format Font Compile Transfer
File.R
* FileResDef -- Resource input for sample application named File
* Written by Macintosh User Education
File.Rsrc
File.Pas
BEGIN
(When the window is resized, the scroll bar needs to be stretched to fit.)
WITH MyWindow.portRect DO
BEGIN
HideControl(vScroll)
ShowControl(vScroll, right-15, top-1);
SizeControl(vScroll, 16, bottom-top-13);
ShowControl(vScroll)
END
END;

```

Il est possible de travailler sur plusieurs fichiers simultanément : par exemple le programme principal, les ressources et le fichier.

est toutefois indispensable de posséder une documentation plus approfondie sur le fonctionnement interne du Mac que le manuel de Turbo Pascal (par exemple « Inside Macintosh » publié chez Addison-Wesley).

Le Turbo Pascal dans sa version Macintosh est plus « standard » que les versions MS-DOS et CP/M. Il reste encore quelques bizarreries, comme la gestion des fichiers qui ne se fait pas avec les fonctions get() et put(), mais toujours à l'aide des instructions writeln() et readln() qui sont étendues à la lecture et à l'écriture sur disque. En revanche, la gestion des pointeurs se fait de façon standard avec les procédures new() et dispose() et non plus mark() et release(). Un certain nombre de fonctions ont changé de nom, en particulier

calculs standards en virgule flottante. Il est possible d'écrire, de compiler et de rassembler ses propres sous-programmes en une bibliothèque d'utilitaires qui viendra compléter l'ensemble existant. L'utilisation de bibliothèques est, par ailleurs, un moyen de segmenter les programmes, ce qui devient nécessaire si le programme objet devient trop gros. En effet le Memory Manager du Macintosh (le programme système qui gère la mémoire) impose que tout programme soit découpé en « tranches » de 32 Ko au maximum. Le compilateur du Turbo Pascal propose un certain nombre d'options de compilation qui

► Pour tout Macintosh. Prix : 995 F HT. Edité et distribué par Borland International.

Mac Plus disposant de 1 024 Ko de mémoire vive, c'est un comble. On peut heureusement contourner cette limitation en utilisant des pointeurs sur des tableaux au lieu de déclarer les tableaux eux-mêmes, mais cette façon de faire est plus complexe, surtout pour un néophyte. Malgré ces quelques contraintes, Turbo Pascal constitue un environnement privilégié pour s'initier à la programmation dans le style Macintosh. Le débutant n'est pas rebuté par une trop grande lourdeur de mise en œuvre et peut réaliser assez vite des programmes qui fonctionnent. Le programmeur chevronné trouvera également en Turbo Pascal un outil de développement complet particulièrement efficace et agréable.

Frédéric NEUVILLE