

## SYSTEME 411 VERSION 3

*Cet article de gestion d'arborescence pour HP-28S nous est parvenu des U.S.A. par le réseau UNIX, reliant plusieurs milliers d'ordinateurs de par le monde. Thomas Affinito, son auteur, a donné l'autorisation aux Clubs de le publier. Paul Courbis l'a donc traduit, et c'est avec plaisir que nous vous le présentons ce mois-ci.*

### INTRODUCTION

Une mémoire à structure de stockage arborescente peut fournir une méthode agréable pour organiser la toujours croissante myriade de variables qu'un utilisateur actif crée pour son usage personnel. La plupart du temps, l'utilisateur passe trop de temps à utiliser ORDER, à monter et descendre les branches de l'arborescence pour chercher à retrouver à quel endroit sont situées certaines variables. Mon premier projet de programmation HP28S était de dompter cette hiérarchie à l'aide d'un système basé au sommet de l'arborescence de manière à libérer l'utilisateur des commandes transversales.

Mon idée était de mettre en oeuvre un concept d'espace de travail (comme en APL) avec des propriétés spécifiques :

1- l'utilisateur doit disposer de commandes simples pour créer (BUILD) et détruire (CRUSH) automatiquement un espace de travail.

2- l'espace de travail doit se comporter comme un banc d'essai pour le développement et les essais de nouveaux programmes, de nouvelles idées.

3- les programmes achevés et les variables doivent pouvoir être supprimés du menu USER (HIDE) et rangés dans un menu parent dans lequel il sera toujours accessible ; de plus les programmes, les variables et, plus généralement, les utilitaires pourront toujours être repoussés dans un menu parent plus élevé qui sera visible depuis tous les espaces de travail créés (STALL, abréviation de *install*).

4- les commandes du système principal peuvent être placées dans le menu *custom* pour en faciliter l'usage ; de plus, l'utilisateur peut créer des labels spécifiques qui seront toujours chargés quand l'espace de travail particulier sera activé (SLAB) ; les labels généraux (présents pour tous les espaces de travail) sont aussi possibles (GLAB) ; ainsi l'utilisateur peut-il cacher (HIDE) ou installer (STALL) tous ses programmes, et nommer de manière locale ou globale (SLAB ou GLAB) les applications principales, ce

qui laissera tout en dehors de l'espace USER, gardera tous les utilitaires opaques et donnera une certaine transparence aux applications.

5- l'utilisateur peut passer d'un espace de travail à un autre avec des commandes simples ; les passages transversaux sont totalement éliminés.

6- le système ne prend pas trop de mémoire.

Le système a été minutieusement débogué et est plein d'utilitaires intéressants ; il est facilement personnalisable et occupe à peine 2 Ko (un faible prix à payer avec la 28S). Voici le système que j'utilise pour toute ma programmation. Je serai intéressé par tous les commentaires, toutes les critiques et tous les compléments à ce système...

### VERSION 3 ?

Après avoir programmé sur la 28C pendant 6 mois, j'ai tressailli quand est sortie la 28S et je l'ai immédiatement achetée. 411 fut créé pour diriger mon style d'écriture de programmes. Après avoir lu le livre de Wlodek, j'ai réalisé que je devais placer mon système dans un sous-menu du menu HOME pour pouvoir conserver PEEK et POKE dans le menu principal (\*). Ainsi naquit la version 2.

Il y a trois jours, j'ai acheté le livre de Bill Wickes, *HP 28 Insights* que je recommande vivement. C'est une extension des manuels, avec d'intéressantes suggestions... Certains de ces programmes étaient similaires aux miens : de manière à rendre 411 plus accessible, j'ai changé certains de mes programmes pour reprendre ses noms, ses algorithmes à la place des miens. A savoir :

UP page 83 de son livre ainsi que dans le manuel HP. Mon programme UP laissait dans la pile le chemin qu'il venait de quitter... Ainsi mes programmes avaient des paires de UP et de DOWN (qui exécutait le chemin dans la pile) quand j'effectuais des chemins transversaux.

DOPATH page 84. Mon programme faisant cela s'appelait DOWN mais j'ai changé le nom pour être en accord avec Bill ; Mon algorithme n'est pas le même... Le mien utilise des utilitaires non-inclus dans son livre et est plus général car il exécute une liste ou un nom.

Purge et CLUSR. J'ai un seul programme (en fait une chaîne) qui fait un nettoyage complet d'un menu (appelé XCL pour *eXtra CLear*) Le Purge de Bill fait la même chose, ainsi ai-je utilisé ses noms et algorithmes... Avec l'exception que CLUSR est placé dans une chaîne d'objets. J'ai fait cela comme protection ; avoir une touche qui peut vider toute la











