

AProf

COLLABORATORS

	TITLE : AProf		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AProf	1
1.1	AProf.guide	1
1.2	Contacts	1
1.3	Droits de copie	2
1.4	Qu'est-ce qu'un profiler	2
1.5	Procédure d'installation	2
1.6	Système requis	2
1.7	Compilateur requis	2
1.8	Utiliser le profiler	3
1.9	Interface utilisateur	3
1.10	Menu: Files/Fichiers	3
1.11	Menu: Action	4
1.12	Menu: Data/Données	4
1.13	Menu: Sort/Tri	4
1.14	Menu: Move/Déplacement	4
1.15	Menu: Misc/Autre	4
1.16	Cycle: sous routines incluses/exclues	5
1.17	Cycle: Temps en pourcentage/millisecondes	5
1.18	XTime	5
1.19	Ligne de bouton	5
1.20	Bas de la fenêtre	5
1.21	Configurability	6
1.22	Caveats	6
1.23	Profile basé sur le Workbench	6
1.24	Module startup spéciaux	6
1.25	setjmp()/longjmp()	7
1.26	Horloge CIA	7
1.27	Overlays	7
1.28	Limitation du Runtime	7
1.29	Fonctions statique	7

1.30	Traphandlers	7
1.31	Fonctions switch et de lancement	7
1.32	Utilitaires	8
1.33	Utility: StripB	8
1.34	Utilitaire: Sym	8
1.35	Utilitaire: Seg	8
1.36	Systèmes testés	9
1.37	Manx Aztec C V3.2 - V5.2	9
1.38	SAS C Vx.x	9
1.39	Maxon C++ 1.2.1	9
1.40	PCQ Pascal Compiler Version 1.2a + A68k V2.61 + BLink V6.7	10
1.41	DICE C V2.06.21 version non enregistrée	10
1.42	GNU C/C++ Vx.x	10
1.43	AmigaGuide (TM)	10

Chapter 1

AProf

1.1 AProf.guide

Documentation pour l'Amiga Profiler V3.2
© 1993 Michael G. Binz

Qu'est-ce qu'un profiler?
Système requis
Compilateur requis
Procédure d'installation
Utiliser le profiler
Configurability
Interface utilisateur
Systèmes testés
Caveats
Utilitaires
Contacts
Droits de copie

1.2 Contacts

Contactez moi à :

EMAIL: sun411@informatik.fh-augsburg.de

Si les Emails ne sont pas possible, écrivez moi à:

Michael Binz
c/o Gisela Fahrner
Schießgrabenstr. 8a
8900 D-Augsburg

(NDT : les codes postaux allemands ont changé depuis le 01/07/93.
celui-ci est donc faux)

Je suis très intéressé par tous les tests qui seront faits avec
n'importe quel système de programmation ainsi qu'avec différentes
configurations Amiga.

Si vous faites un test avec un compilateur non listé dans 'systems', envoyez moi SVP un fichier démo, l'exécutable compilé (avec les symboles hunk) et quelques informations sur la création de ces hunks avec votre compilateur. Si nécessaire, envoyez une description des problèmes et actions que vous avez résolues.

1.3 Droits de copie

Ce programme est freeware. Vous pouvez le copier, distribuer et l'utiliser, tant que vous ne modifiez pas le programme et que vous ne distribuez que le fichier archivé (AProfilerX_XX.lzh). Vous ne pouvez pas demander plus de taxes que celle faites par Fred Fish pour la copie et la distribution.

© 1993 Michael G. Binz

1.4 Qu'est-ce qu'un profiler

Un Profiler est un outils de développement. Il lance un programme spécifique et collecte des informations pendant son exécution. Ces informations peuvent être utilisées pour trouver des fonctions dans votre code, où l'optimisation aura le plus d'effet.

1.5 Procédure d'installation

Copiez tous les fichiers de l'archive AProfilerX_XX.lzh dans un répertoire et ajoutez ce répertoire dans votre chemin (PATH).

1.6 Système requis

Le profiler a besoin du WorkBench/KickStart 2.04 et un minimum de 500 Ko libre de mémoire.

1.7 Compilateur requis

Votre compilateur doit être capable de créer des 'symbol hunks' Amiga. Les 'symbols' inclus dans les hunks doivent seulement adresser les adresses de départ des fonctions dans votre code (le 'code symbols' ne sont pas pris en compte par le profiler). Si votre compilateur crée des 'symbols' en plus, ils doivent être enlevés avec 'Misc/Mask Filter' (regardez dans 'Systems', peut-être que votre système y est listé).

1.8 Utiliser le profiler

Si vous utilisez le compilateur à partir du CLI (ce qui est recommandé), lancé le avec

```
1.DEV:c/app> AProf application
```

Le profiler se lance, affiche son interface graphique charge les tables de 'symbols' du fichier 'application' et attend les actions que vous voulez faire. Si le nom de fichier manque, il sera demandé.

```
@{ " Utilisation sous Workbench " link ca-wb }
```

1.9 Interface utilisateur

```
Menus:  @{ " Files " link m-files } @{ " Action " link m-action } @{ " Data " ↔
        link m-data } @{ " Sort " link m-sort } @{ " Move " link m-move } @{ " Misc " ↔
        link m-misc }
```

```
Gadgets: @{ " In/Exclusive timevalues " link gad-inex } @{ " Percentual/ ↔
          Millisecond timevalues " link gad-percmil } @{ " XTime " link gad-xtime }
          @{ " Button row " link gad-buttons }
```

```
Menus:  @{ " Fichier " link m-files } @{ " Action " link m-action } @{ " Données ↔
          " link m-data } @{ " Tri " link m-sort } @{ " Dépl " link m-move } @{ " Autre ↔
          " link m-misc }
```

```
Boutons: @{ " Temps In/Exclus " link gad-inex } @{ " Temps pourcentage/ ↔
           Millisecondes " link gad-percmil } @{ " XTime " link gad-xtime }
           @{ " Ligne bouton " link gad-buttons }
```

Description des entrées de la table:

Fonction: C'est le nom du 'symbol'

HitCnt: C'est le nombre de fois que la fonction a été appelée

Average: Temps moyen d'exécution pris par la fonction

Over: Temps maximum d'exécution pris par la fonction

Min/Max: Temps d'exécution minimum/maximum.

```
@{ " Bas de la fenêtre " link gad- ↔
  bottom } }
```

1.10 Menu: Files/Fichiers

Open:	Ouvre:	lit un nouveau fichier
Save:	Sauve:	Sauve le table de 'symbol' avec le même nom et un suffix .pro
Save as:	Sauve sous:	Sauve la table de 'symbol' vers un fichier choisit
Reset:	ReInit:	Remet à zéro toutes les valeurs
Print:	Imprimer:	Pas encore implémenté

Exit: Sortir: Quitte le programme
About: A Propos de: Affiche la version, copyright

1.11 Menu: Action

Start: Lancer: Lance le programme. Si vous le relancé, les temps sont ajoutés.

1.12 Menu: Data/Données

Exec détails/Détails d'exécution:

Dans cette requête, vous pouvez entrer une ligne de commande (sans le nom de la commande) et la taille de la pile pour le programme à profiler

Exemple de ligne de commande:

Exécutable chargé: 'list'
ligne de commande: 'c:#? all'

1.13 Menu: Sort/Tri

Pas encore implémenté

1.14 Menu: Move/Déplacement

Find: Recherche: Entrez une chaine à rechercher et commence la recherche
Find next: Suivant: Recherche la chaine suivante
Top: Haut: Se met en haut de la table de 'symbols'
Bottom: Bas: Se met en bas de la table de 'symbols'
Page up/down: Page avant/suivante: une page plus haut ou plus basse.

1.15 Menu: Misc/Autre

Help: Aide:

Lance le système hypertext @{ "AmigaGuide" link l-amigaguide }(TM)
(si il est accessible)

Symbol filter: Filtre 'symbols'

Entrez le modèle des 'symbols' à exclure

Exemple:

.#? Exclue tous les 'symbols' commençant par '.'
#?%#? Exclue tous les 'symbols' ayant '%' dans leur nom

(Pour une explication des modèles Amiga, reportez vous au manuel

```
utilisateur)
```

Refresh Window: Rafraichir la fenêtre:

```
Reaffiche la table des 'symbols'
```

1.16 Cycle: sous routines incluses/exclues

o Sous routines incluses

Ceci ajoute les temps des appels à des sous routines aux temps de ceux qui appelle.

Dans les programmes C, le temps pour 'main()' est proche du temps d'exécution général, parce que normalement toutes les fonctions sont appelées de main().

o Sous routines exclues

Ici, le temps d'une fonction exclue le temps nécessaire d'une sous routine appelée.

Si vous ajoutez toutes les valeurs de la colonne 'Over' de la table de 'symbols', la somme doit être proche du temps d'exécution général du programme profilé.

1.17 Cycle: Temps en pourcentage/millisecondes

o Precentual timevalues: Temps en pourcentage:

Tous les temps sont en pourcentage (%) pour tous les programmes profilés.

o Millisecond timevalues: Temps en millisecondes

Tous les temps sont en millisecondes.

1.18 XTime

Temps d'exécution général d'un programme profilé en millisecondes

1.19 Ligne de bouton

Une façon rapide d'atteindre les entrées menus souvent utilisées.

1.20 Bas de la fenêtre

En bas de la fenêtre, le compilateur affichera les messages utilisateurs.

1.21 Configurability

Toutes les options configurables d'AProf doivent être sauvées dans les ToolTypes de l'icone AProf.info
Le fichier .info doit être dans le répertoire que le profiler.

- o WINDIM=gaucche/haut/largeur/hauteur
Utilisez ça pour positionner la fenêtre du profiler.
(Dans la version 3.2 'largeur' doit être donné, mais la valeur n'est pas utilisée).

1.22 Caveats

Voici une liste de construction qui peuvent donner des problèmes, si vous essayez de profiler des programmes les contenant :

```
@{ " Profil basé sur le Workbench " link ca-wb }
@{ " Modules Startup non-standard " link ca-startup }
@{ " setjmp()/longjmp() " link ca-sljmp }
@{ " Utilisation des horloges CIA " link ca-timers }
@{ " Overlays " link ca-overlays }
@{ " Limitation du Runtime " link ca-runmax }
@{ " Fonction statique " link ca-static }
@{ " Traphandlers " link ca-traps }
@{ " Utilisation de fonctions switch et de lancement " link ca-swila }
```

1.23 Profile basé sur le Workbench

Si lancé à partir du workbench, le profiler utilise le message startup envoyé par le Workbench. Les programmes-workbench à profiler hérite de l'environnement du profiler, attendront indéfiniment ce message déjà utilisé.

Utilisez si possible le plus simple module startup possible. Dans la plupart des cas, vous éviterez votre programme d'attendre ce message. D'un autre coté, vous avez aussi à analyser la ligne de commande dans votre code.

1.24 Module startup spéciaux

Les programmes à profiler doivent être capable de tourner dans le même environnement que le profiler. Donc il n'est pas possible d'utiliser des codes startup spéciaux pour détacher un programme ou pour le mettre résident.

1.25 setjmp()/longjmp()

Si les combinaisons `setjmp()/longjmp()` sont utilisés dans le programme à profiler, essayez d'exécuter un rts depuis la fonction cible de `longjmp`. La logique du profiler le demande pour connaître quelle portion de programme il exécute.

1.26 Horloge CIA

Les horloges CIA ne sont pas disponible pour les programmes à profiler.

1.27 Overlays

Profiler un programme 'overlayed' n'est pas possible.

1.28 Limitation du Runtime

L'horloge du profiler peut mesurer des temps maximum de 99 mins.

1.29 Fonctions statique

Les fonctions à mesurer DOIVENT être dans la table de 'symbols'. Ce n'est pas le cas des fonctions statiques dans la plupart des environnement de programmation.

1.30 Traphandlers

Si votre programme utilise des 'trap handlers' privés, les 'traps' n'étant pas dans un 'handler' privé doivent être propagé au 'handler' précédent.
Profiler n'est pas possible si vous utilisez des 'trace traps' (#9).

1.31 Fonctions switch et de lancement

Les programmes à profiler qui utilise des membres `tc_Switch` et `tc_Launch` dans une structure de tâche Exec DOIVENT propager l'exécution aux 'handlers' précédent.

1.32 Utilitaires

Pendant le développement du profiler, j'ai écrit quelques utilitaires qui peuvent parfois être utiles:

```
@{ " StripB: " link util-stripb } Enlève les HUNK_SYMBOL et HUNK_DEBUG des ↵
    exécutables
@{ " Sym:      "      link util-sym } Affiche les HUNK_SYMBOL
@{ " Seg:      "      link util-seg } Affiche les listes de segments.
```

1.33 Utility: StripB

StripB enlève tous HUNK_SYMBOL et HUNK_DEBUG d'un exécutable Amiga.

Ligne de commande:

```
StripB fichierentree fichiersortie
```

1.34 Utilitaire: Sym

Utilisez Sym pour afficher le code 'symbols' (HUNK_SYMBOL) dans un exécutable Amiga. -----

Ligne de commande:

```
Sym app ...
```

Structure de la sortie de Sym:

```
Colonne:   1       2       3           4   5
Donnée : 39    ( 0 /      d90)          d3e  __Readfile__NoBuf
```

```
Colonne 1: nombre de 'symbol'
          2: nombre de 'hunk'
          3: taille du 'hunk'
          4: Adresse du 'symbol' relatif au 'hunk' courant
          5: Nom du 'symbol'
```

Message d'error:

```
Symbol out of hunk bounds      'Symbol' hors limite
La position des 'symbols' est hors des limites du 'hunk'
(utilisez par quelques compilateurs)
```

1.35 Utilitaire: Seg

Seg affiche le list des segment d'un exécutable Amiga

Ligne de commande:

```
Seg prog
```

1.36 Systèmes testés

Voici une liste de systèmes que j'ai testé avec le profiler. Si vous trouvez une erreur ou si vous avez testé un système non listé ici, envoyez moi un message.

```
@{ " Aztec C V3.2 - 5.2 " link manx-c}  
@{ " DICE V2.06.21 " link dice-c }  
@{ " GNU C/C++ Vx.x " link gnu-cpp}  
@{ " PCQ Pascal V1.2a " link pcq-pascal}  
@{ " Maxon C++ V1.2.1 " link maxon-cpp}
```

1.37 Manx Aztec C V3.2 - V5.2

Problèmes:

Aucun

Création des 'symbol hunks':

Utilisez l'option -w pour le linker

Autre:

Vous n'avez pas besoin d'enlever les 'symbols' appelés '_Hx_org'.

N'utilisez pas 'detach.o' avec les programmes que vous voulez profiler.

1.38 SAS C Vx.x

Aucun

1.39 Maxon C++ 1.2.1

Problèmes:

Parfois, mélanger du code et des 'data symbols'. Vérifiez la table de 'symbol' et enlevez les 'data symbols' via le menu Misc/Symbol Mask

Le compilateur ajoute des labels appemés L999 à la table de symbole. Ceci doit être enlevé.

Création des 'symbol hunks':

Pour les lignes de commandes, le compilateur utilise l'option -bs.

Dans l'environnement intégré, utilisez les options du compilateur.

Autre:

Si vous oubliez d'enlever les 'symbols' L999, un GURU va arriver.

Pas de reconstruction des 'symbols' C++ ('No unmangling of c++ symbols').

1.40 PCQ Pascal Compiler Version 1.2a + A68k V2.61 + BLink V6.7

Problèmes:

Parfois, mélanger du code et des 'data symbols'. Vérifiez la table de 'symbol' et enlevez les 'data symbols' via le menu Misc/Symbol Mask

Le compilateur ajoute des labels appelés *%. Ils doivent être enlevés.

Creation of symbol hunks:

Utiliser l'option -d pour l'Assembleur

1.41 DICE C V2.06.21 version non enregistrée

Problèmes:

Aucun

Création des 'symbol hunks':

Utilisez l'option -s pour DCC.

1.42 GNU C/C++ Vx.x

Problèmes:

Il semble qu'il n'y ai aucun problèmes. Je n'ai pas testé ce compilateur, seulement testé quelques exécutable créés avec GNU C

Création des 'symbol hunks':

??

1.43 AmigaGuide (TM)

AmigaGuide est un programme d'affichage avec des fonctions hypertextes.

AmigaGuide est FreeWare pour des Freeware non commerciaux et est disponible via FTP.