



L'association des utilisateurs de Falcon

3

FALCONtact

Fanzine

Bon Envol '94

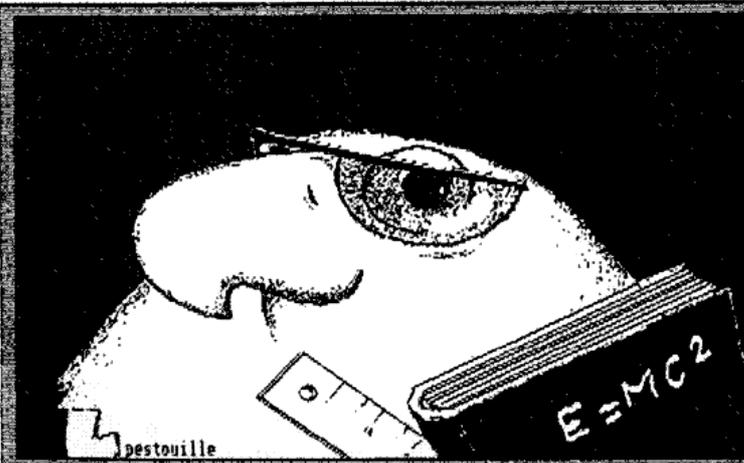
ENFIN!!

**EXPLOITER
LE SON DU
FALCON**

EN

- la mémoire vidéo
- Courrier des lecteurs
- Flabbergasted: Ishtar 2

V
R
A
C



**PLUS DE LA
TECHNIQUE!**

Janvier-Février 1994.

Bonne année 94 Souvenez vous, il y a dix ans, 1984...Apple lance le Macintosh et la souris révolutionnait le monde de l'informatique, Amstrad sortait le CPC qui allait participer activement au boom de la micro familiale, Los Angeles accueillait les J.O. d'été, Dire Straits sortait le mythique Money for Nothing et j'avais dix ans (mais ça vous vous en souvenez !) Qu'en reste-t-il dix ans après ? Le Mac 128 fait pâti face au Quadra 840 AV et autres Power PC, Amstrad ne fait plus que des PC et des antennes paraboliques, on jasse toujours Money for Nothing à la radio, le Falcon s'envole petit à petit vers la gloire et Faucontact fait peau neuve: beaucoup plus de technique, d'articles d'adhérents, nouvelle maquette... Le fanzine prend sa vitesse de croisière avec une augmentation notable (et durable) de la qualité par rapport aux numéros précédents et j'ai vingt ans (mais ça encore vous vous en tapez !) Que restera-t-il de tout ça dans dix ans ? En 2004 il y aura des Falcons partout, le RTC (qui ouvre d'ici au prochain numéro) sera en 128000 accès, avec téléchargement à 64800 bauds, Faucontact Magazine International fera 400 pages, tout en couleurs et sera tiré dans le monde entier à des millions d'exemplaires, Pset sera à la tête du pays (du monde ?), Worraps à celle de la plus grande banque mondiale, Splash sera le boss d'Atari Corp., Stef celui d'Apple (le traître !), et moi... j'aurais 30 ans (mais ça une fois de plus vous n'en aurez rien à secouer !!).

EDOX

Directeur de la publication:

Frank "WORRAPS" Cathelain

Rédacteurs et collaborateurs:

Pascal "PSET" Martin, David "SPLASH" Carrère, Julien "EDOX" Barret, Nicolas "BESTBUG" Perret, Christian "ENITALP" Hnau, CASUS, François "EROSTRATE" Jacquemin.

Excellents dessins de:

Serge "PESTOUILLE" Tortet

Conseils Xpress de:

Stéphane "STEF30" Darras

Pour nous contacter écrivez à l'adresse ci-dessous, ou laissez un message en bal FAUCONtact sur 3615 STMAG ou 3614 ENTROPIE. Secte FAUCONtact sur 3615 STMAG *SL FAU

L'envoi de textes, photos ou documents quelconques implique l'acceptation par leur auteur de leur libre publication dans le fanzine. La rédaction décline toute responsabilité quant aux opinions formulées dans les articles, celles-ci n'engageant que leurs auteurs.

Toutes les marques citées sont déposées par leur propriétaire respectifs.

Ce fanzine a été entièrement réalisé sous Calamus sur un Falcon 030.

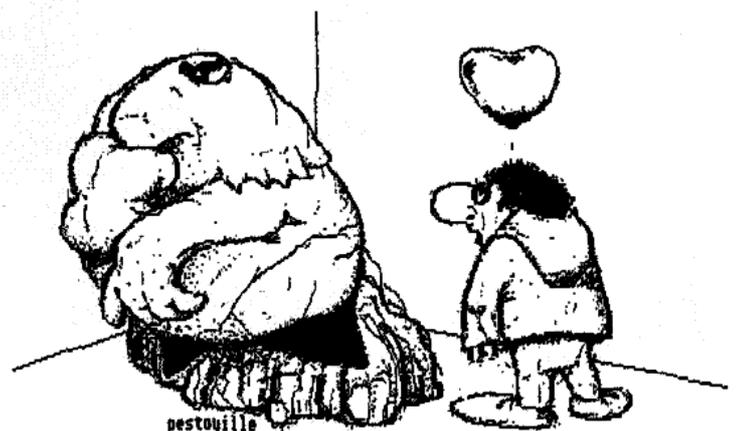
FAUCONtact fanzine est édité par l'association FAUCONtact (JO du 17/08/1993)

FAUCONtact,

Mas de Crès,

30140 Boisset et Gaujac.

RUBRIKASSO	3
GRAINS DE POUSSIERE	4
MULTIMÉDIA	6
EXPLOITEZ LE 68030	7
LEÇON DE SON	13
ROUTINES GRAPHIQUES	16
INITIATION AU C	18
FLABBERGASTED	20
COURRIER	21
DOMAINES PUBLICS	22
BULLETIN D'ABONNEMENT	24
BON DE COMMANDE DES DPs	24



RUBRIKASSO



PARLONS DE NOUS

FAUCONtact fanzine évolue et voici un des fruits de son évolution: la rubrique de l'association. En effet, plutôt que de vous faire la rubrique intro qui devait parler de l'air du temps, nous avons pensé qu'il serait plus judicieux de vous parler de l'association, de son évolution ainsi que de ses initiatives.

A la demande de certains adhérents il a été décidé la mise en place d'un système de parrainage. Ainsi toute personne nous amenant un autre adhérent pour six mois a droit à deux disquettes DP gratuites, et si l'adhésion est d'un an, ce sont trois disquettes DP gratuites. Certains vont trouver que ce n'est pas grand chose ... quand nos moyens financiers seront plus larges, et surtout, quand nous auront bien cerné les variations de nos dépenses, on fera mieux, c'est promis. Mais comment faire pour parrainer quelqu'un ? Ce n'est pas compliqué: la personne qui adhère inscrit au dos de son bulletin d'abonnement le nom et l'adresse complète de son parrain (qui doit être, bien sûr, déjà adhérent à FAUCONtact). Simple, non ?

Autre décision concernant, là aussi, la demande de certains adhérents, qui désiraient que l'on publie la liste des adhérents de l'asso. dans le fanzine. Hé bien, c'est non, non et non ! Nous n'avons pas le droit de publier les coordonnées de nos adhérents sans leur consentement car ce sont des informations considérées comme confidentielles par la loi. De plus rien ne nous prouve que cette liste d'adresse restera entre les mains d'adhérents de FAUCONtact, et que l'on n'en fera pas un usage autre que celui auquel elle est sensée être prévue. Si vous voulez avoir des contacts, je vous rappelle que vous

avez la rubrique Recherche de Compétence à votre disposition et qu'elle est gratuite. D'ailleurs, on va peut être créer aussi une rubrique "petites annonces" plus adaptée à la recherche de contacts...

Enfin, comme vous ne l'avez peut être pas encore remarqué, le fanzine est passé à vingt quatre pages. En effet ce mois-ci nous avons eu pas mal d'articles, notamment au niveau programmation. Et encore, nous avons du compresser pour tout faire rentrer dans le fanzine (non, non pas du JPEG !). De cela découle deux choses: d'une part on espère que certains d'entre vous ne seront pas trop surpris par

Pour terminer on va vous demander plein de pardons parceque l'on n'a pas toujours fait parfaitement le travail que l'on attend de nous:

Pardon pour le retard des premiers parainages !

Pardon pour le retard d'envoi des DPs aux auteurs d'articles du fanzine !

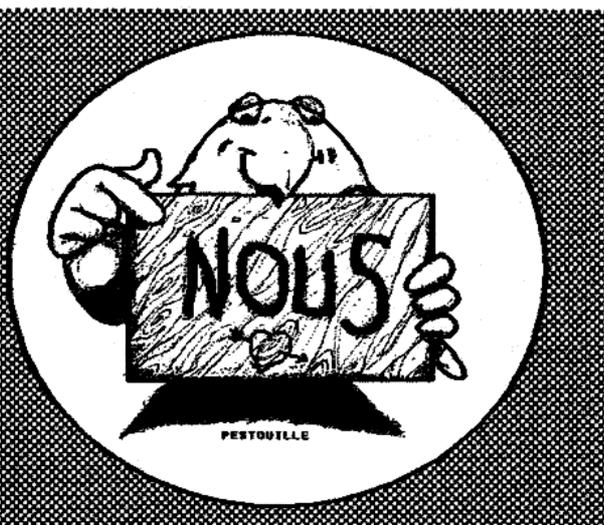
Pardon pour nos réponses incroyablement lentes au courrier de nos adhérents chéris ! on va vous répondre promis mais laissez nous respirer !! il nous faut nous renseigner pour vous répondre ! On ne voudrait pas vous dire n'importe quoi !

Pardon Bestbug pour avoir oublié de te citer dans la liste "Rédacteurs et collaborateurs" du mois dernier ! (sa signature était tout de même présente à la fin de l'article !).

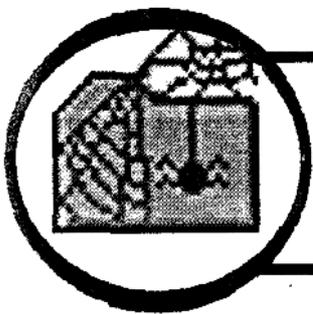
Je vous rapelle que ce fanzine est aussi votre moyen d'expression... si vous avez des articles à écrire n'hésitez pas. Non seulement ça vous rapporte des DPs gratos, mais en plus on n'a pas à les écrire !! Car si en plus de s'occuper de l'asso. il faut qu'on gratte tout le fanzine, là on nous enterre demain ! Et puis on ne connaît pas tous les domaines du Falcon, voyons !

Pour conclure, FAUCONtact est une association: ce sont donc des personnes qui mettent en commun leur connaissances et leur passion. Nous (les dirigeants) nous nous débrouillons pour que l'association tourne. Mais ce n'est pas une raison pour que vous restiez sans bouger ! Cette association c'est surtout vous qui la faites vivre ! Ne l'oubliez pas !

Splash.



cette arrivée en fanfare de la programmation (cela nous surprendrait d'ailleurs), mais cela est nécessaire pour que le Falcon ait des logiciels. D'autre part, ce constat entraîne une remarque: ce mois-ci, certaines rubriques ont du être diminuées au profit d'autres car la place nous manquait. Il est clair que si cela continuait, nous devrions encore augmenter le volume du fanzine pour le plus grand plaisir de tous. Mais plus de pages dit plus de coûts... donc pour l'instant le prix de l'adhésion reste intouché mais si le fanzine gonflait trop, elle risquerait d'augmenter (pas beaucoup, je vous rassure !).



GRAINS DE POUSSIERE

LE MONDE A VOS PIEDS

GOOD NEWS

Tel est le nom d'un petit fanzine qu'édite la S.A. Ados, un revendeur Belge. Intitulé "La bible du Falcon", ce fanzine date un peu (Février 1993) mais il possède une présentation assez complète du Falcon et de ses possibilités. Il présente aussi certains accessoires vendus, ainsi que des goodies Atari rarement vus ! (des pin's, des T-Shirts, des tasses à café, des boucles de ceintures !). Il y a aussi une banque de domaines publics, mais ceux-ci semblent surtout destinés au ST... des Midifiles et des illustrations toutes prêtes sont aussi disponibles. Pour conclure, un fanzine sympa mais dont le contenu n'est pas vraiment extraordinaire. Néanmoins il est offert gratuitement à tous nos adhérents, il suffit de communiquer à Ados... leur adresse. Or comme vous le lirez dans la rubrique de l'asso la loi nous l'interdit sans votre autorisation. Les adhérents qui sont intéressés par ce fanzine doivent donc nous envoyer une lettre avec le texte suivant : "J'autorise l'association FAUCONtact à divulguer mon adresse auprès de la S.A. ADOS (Bruxelles) afin de recevoir leur fanzine intitulé "Good News Magazine". Si par hasard mon adresse venait à circuler par le biais de cette société, l'association FAUCONtact ne pourrait en aucun cas en être tenue pour responsable.". Vous signez et datez le tout, et hop vous nous l'envoyez ! votre adresse sera transmise dans les plus brefs délais à Ados. (nb: cette formalité n'est pas une preuve de méfiance particulière à l'encontre de la S.A. Ados, mais juste une précaution afin d'éviter tout ennui d'ordre juridique à l'association).

MINI F1

Bon, après les hurlement qui se sont dressés de part et d'autres de la foule, lors de la réception de la version 1.01 de MINI F1, FAUCONtact et le programmeur Gilles Audoly se sont concertés. Le résultat en est une version 1.03 avec de magnifiques graphismes en True Color de notre cher président PSET, d'excellent effets agrémentant le jeu (rotation 3D mappée des circuits avant d'y jouer, drapeaux plus gros en 3D, un peu de digit), et une jouabilité améliorée. C'est sûr que par rapport à la précédente version, ça fait du bien: MINI F1 peut donc être qualifié de "bon pour le service !". Quand à nos adhérents qui ont acheté la version 1.01 (et plus généralement pour tous les acheteurs de MINI F1) il vous suffit de renvoyer votre original à l'éditeur MSI BA (en précisant votre adresse niak niak...) qui vous l'échangera contre la version 1.03. Alors, qui va se plaindre après ça, hein ? Bande de râleurs ! C'est pas sur PC que l'éditeur vous

aurait refait un jeu parce que vous n'en étiez pas satisfaits ! Ensuite, à propos des adaptateurs qui vous sont offerts avec le jeu (uniquement pour les adhérents cette fois-ci), mille excuses si vous ne les avez pas encore reçus: quelques problèmes de fabrication ont retardé leur sortie, mais la promesse sera tenue ! Voici l'adresse de MSI B.A.: 1, Rue de la gendarmerie, 26230 Grignan.

ENCORE DES SOFTS MOINS CHERS ?

Bon, là c'est un sondage. Si un nombre suffisant d'adhérents est intéressé par le logiciel Studio Photo, nous pourrions l'avoir pour beaucoup moins cher. Disons... moins de 400 francs à peu près. Alors si vous êtes intéressés vous nous envoyez un courrier comme quoi...ça vous intéresse. Suite à quoi, si le nombre requis de demandes est atteint, vous serez contactés par nos soins afin de confirmer votre demande. Notez que le fait de nous dire que vous êtes intéressés n'est pas un engagement officiel, mais bon, on espère que si la chose se faisait, 99% des demandes se concrétiseraient... sans quoi le nombre ne serait plus atteint ! Cette offre se veut bien sûr pour un logiciel en tout point identique à la version disponible actuellement dans le commerce. Alors, intéressés ?

SOFTS !

Nous venons de l'apprendre, Z'ANIMOS et D2M sont disponibles. Le premier est un logiciel éducatif en plein de couleurs, le second un logiciel de dessins dont les preversions nous font beaucoup espérer. On va tâcher d'en savoir plus ! Ces deux logiciels sont édités par PARX Software. Tiens à propos de logiciels dispos, Semprini est enfin disponible depuis quelques temps. Rapellons que cet utilitaire se propose de remédier à beaucoup de maux de vos disquettes et disques durs. D'ailleurs il a tiré l'association plusieurs fois du pétrin ! (comme par exemple les données orphelines d'AHDL...). Non, non on n'a pas été payé pour en parler, on le trouve sympa c'est tout !

ASSEMBLEE GENERALE

Hé oui, il faudrait peut être y penser. Comme on est lent à se

mettre en place, on préfère s'y prendre à l'avance ! Cette assemblée se déroulera donc cet été durant un week-end complet (jour ET nuit non stop !). Nous commençons à réfléchir un peu aux activités qui y seront organisées. On pense faire une réunion d'une heure ou deux où on vous parlera un peu de l'asso, on vous demandera également votre avis sur certaines décisions, etc... puis pour le reste, on pense répartir tout ça en plusieurs activités: coding, jeux, etc... avec peut être des concours. Si vous avez des suggestions à faire, c'est le moment !

EXCELLENT !

Hé oui, il existe une bande de fanatiques d'Atari qui ont décidé de se regrouper afin de faire vivre leurs machines préférées. Ils éditent un fanzine sympa qui s'appelle ST MAG, et qui n'est pas mal même si parfois... on rapelle leur adresse: ST Magazine, 5/7 Rue Raspail, 93100 Montreuil. Bon, allez trêve de plaisanteries, merci STMAG d'avoir parlé de nous c'est sympa ! et vous achetez-le car c'est un des rares magazines pour le ST,TT et Falcon qui soit bien et qui subsiste encore !

NOUVEAUTES

Turtle Bay nous informe de l'arrivée prochaine d'une nouvelle carte de digitalisation en 256 couleurs et en temps réel pour environ 2000 francs. Son nom est Screen Eyes (le fanzine sus mentionné en a parlé dans son dernier numéro !). Un nouveau Tower Falcon est également annoncé. Espérons que pour une fois il alliera pratique et esthétique !

RECHERCHE DE COMPÉTENCES

Dans la série j'ai besoin de l'âme sœur, MC JEE nous écrit.

Il recherche des graphistes pour un jeu de course de voitures (Dans le style VROOM par exemple...) en 3D Mappé. Tout cela en True Color bien entendu.

Il propose aussi un catalogue de modules personnels pour demo ou autre...

Nous vous rappelons que cette rubrique ne mord pas, elle vous est ouverte pour vous permettre de faciliter la réalisation de vos projets.

INTERVIEW



MONSIEUR DPs

FAUCONtact: Alors Pascal, tu t'appelles comment ?

Pascal: Heu... je m'appelle Pascal

FCT: Et tu fais quoi dans la vie ?

Président: Je suis graphiste de Mini F1...

FCT: Ha bon ?!! Et tu fais quoi à tes moments perdus ?

Maître: Heu... je fais des portes !

FCT: Quels sont tes projets ?

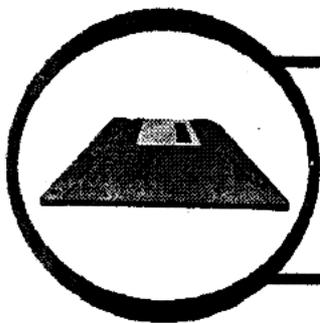
Seigneur: Je vais aller au Mc DO.

FCT: Ha, et quelque chose d'autre à déclarer ?

Dieu: Merci les adhérents pour la 106 !!

FCT: Et bien, merci Capitaine Flamme pour cette interview.

La secte
FAUCONtact
3615 ST MAG
*SL FAU



MULTIMÉDIA

POURQUOI TANT DE HAINE !?

MAIS QU'EST CE QUE C'EST DONC !?

Je ne vous apprendrais rien en vous disant que vous possédez une machine pouvant profiter des applications du Multimédia. Un bien beau mot le Multimédia, mais créer, échanger et manipuler des informations n'est pas un concept nouveau, il existe même depuis nos plus illustres ancêtres. Aujourd'hui les ordinateurs sont les meilleurs conducteurs de cette nouvelle industrie. Mais arrêtons le Multimédias Multitruks où les règles sont dictées par les enjeux économiques et où le consommateur se perd. Exemples de stupidités...

MULTIMÉDIAS MULTISUPPORTS

Quand on parle Multimédias nombreux sont ceux qui pensent au CD pour ses capacités de stockage. Jusque là tout le monde est d'accord, mais la douleur se fait sentir quand on s'aperçoit que le CD (Ou la qualité CD.) est représentée par pas moins de huit supports différents (Et j'en oublie.). Du simple et presque banale CD, au Mini-Disc en passant par la DCC, la DAT (Digital Audio Tape) pour le son, le Laserdisc (Ex CDVidéo.) bientôt suivi par le Vidéo-CD (Tout numérique) pour la Vidéo, le CDI, le CD Photo et le CD-ROM. Autant de supports pourraient être regroupés en trois grands groupes, un pour le son, l'autre pour la vidéo (Et pour le son qui va avec bien sur!) et un dernier pour les CDI, ROM, et Photo. Mais d'ici à ce que tout ce beau monde se décide à faire ça ou un CD pour tout, c'est pas demain la veille que cela nous arrivera et d'ailleurs la multitude de lecteurs CD qui se trouvent sur le marché ne vont pas simplifier l'uniformisation du produit.

On oublie aussi le principale défaut du CD. On ne peut écrire dessus à volonté, d'autant plus que les graveurs ne sont accessibles aux

particuliers qu'à partir de de 40000 Francs, ce qui limite pour l'instant le nombre de possesseurs potentiels.

On néglige aussi l'existence du disque magnéto-optique qui bien que moins capacitif qu'un CD offre tout de même de grandes possibilités malgré un prix assez élevé.

Quoi qu'il en soit et quoi que nous pensions, le CD est le support des prochaines années mais d'ici à leur grandes diffusions nos bon vieux DD ont de beaux jours devant eux.

MULTIMÉDIAS MULTILOGICIELS

Le Multimédias n'existe que par les applications qui le caractérise. La retouche photo, le graphisme, la vidéo, le son, j'en passe et des meilleurs, sont pour notre plus grand plaisir disponible sur notre machine. Le seul problème, hormis peut être leur prix, réside dans le fait qu'aucune de ces applications ne sont à proprement parler des applications Multimédias (Chaque logiciel est en fait un grain de sable dans le bac à sable du Multimédia.). D'accord, faire un logiciel regroupant dessin, retouche photo, PAO, animation, montage vidéo... n'est pas chose facile (Mais ça serait sympas.). En espérant que ce type de logiciel existe un jour, je souhaite que leur prix ne soit pas excessif. En écrivant ces lignes je viens même à rêver d'un DP de ce type (Prions mes frères.).

MULTIMÉDIAS MULTIMACHINES

Là, se pose le plus gros problème, car l'uniformisation des supports et des programmes est en route, mais l'uniformisation des machines n'est prête d'aboutir. La guerre entre les PC, Mac, Atari et Amiga est bien loin de voir apparaître un compromis. Le monopole des PC pourrait voir la véritable généralisation de ces machines mais elles sont loin d'être conviviales, contrairement au Mac qui lui l'est, mais qui est beaucoup moins

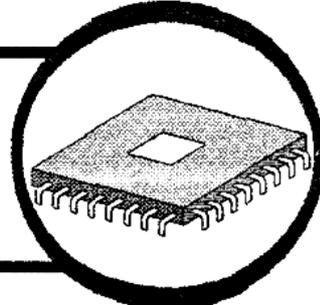
répandu. Quand à Atari, Amiga et les autres, ils ont parfois de bonnes idées mais ce n'est qu'une goutte d'eau, il faut être réaliste. De toute Façon l'apparition d'un "Standard" Micro n'est pas pour demain les enjeux économiques étant important, et la crise n'arrangeant rien aux choses...

MON MULTIMÉDIA À MOI

Ma conception du multimédia c'est mon Falcon. Tout ce qui l'exploite me ravi. Mais qu'en sera t'il dans quelques années quand il sera dépassé. Nul ne le sait. J'ai fait cet article juste pour vous rapeller que le Multimédia est pour l'instant un vaste futoir dans lequel nous sommes plongés et qu'il faut tout faire pour en sortir, et nous nous en sortirons en produisant de belles choses tout en se démarquant de ce que font les autres. Le Multimédia n'est qu'un argument de vente utilisé pour relancer le marché, se qui est légitime mais il y a des fois où le ras-le-bol se fait sentir. On nous bassine avec tout le temps, ce qui a pour effet de vulgariser le terme Multimédia alors que nous n'aimons l'informatique que par ce qu'il représente. Nous devons donc faire attention quand nous l'utilisons, et devons également nous mettre au travail, pour que d'ici peu le Multimédia devienne vraiment la possibilité pour tous d'accéder à tout type d'information avec un minimum de connaissance. A bon entendeur, salut et bonne année.

WORRAPS

68030



Lo touterikiki maousekosto !

Pourquoi pas un article sur le 68030 ? L'assembleur intéresse sûrement un certain nombre d'entre nous, et tout le monde ne connaît pas forcément malgré les livres tous les détails de l'assembleur fabuleux du processeur principal du Falcon. Mais je ne suis pas un programmeur professionnel, mon activité habituelle est tout autre; si bien sûr les gens de métier trouvent matière à indulgence envers ce travail, les autres pourront peut-être y trouver quelque intérêt.

Le 68000 nous avait donné coutûme d'une écriture simple, rapide, pleine de facilités grâce à un jeu d'instructions riche, à une syntaxe logique, des possibilités étendues. Le 68030 passe presque inaperçu tant il est aisé de se reposer sur la compatibilité ascendante pour assurer celle des machines. Pourtant, il offre de nouvelles possibilités, tant au niveau des instructions que de l'environnement matériel, avec de nouveaux registres dont il faut bien tenir compte.

Mais comment le présenter ? J'ai une spécialité, dans ma cuisine informatique : celle des inutilitaires. Toutes sortes de programmeurs s'achament à produire des utilitaires en tout genre qui encombrant les bibliothèques de programmes du domaine public, ou les collections de boîtes à outils des éditeurs. Je n'encours pas ce risque : je programme surtout des inutilitaires. (Peut-être faudrait-il que je songe à demander à Dave Small ce qu'il pense de ceux de mon espèce et dans quelle catégorie les classe-t-il selon Myers-Briggs). Voilà au moins une occasion d'en caser un et de le rendre, sinon utile, tout au moins public... C'est mieux que rien.

Un programme produit un résultat. Ce résultat parfois ne sert à rien, mais c'est un résultat tout de même. Celui-ci en a un puisqu'il copie la table des vecteurs d'exception dans un fichier ASCII, autorisant sa lecture et son impression. Et même dans une série de dix fichiers, si l'opération est répétée à plusieurs reprises. Il inscrit ces fichiers sur le lecteur AA systématiquement et s'il rencontre un fichier par lui déjà constitué, il en écrit un autre, incrémente le numéro

de fichier, qui est compris dans le nom sous lequel il apparaît au répertoire. S'il trouve le dixième, il écrase le premier. Il a une particularité: rédigé par un paresseux, il est prévu pour ne tourner QUE sur une machine équipée d'un 68030. C'est assez facile d'en donner une version 68000 et même de le rendre compatible avec tout processeur de la série 680xx, mais voilà, je ne l'ai pas fait, pour me contenter d'indiquer comment faudrait-il s'y prendre.

Je me propose, par contre, de commenter quelques-uns des points nouveaux rencontrés avec le 68030 :

- 1 - Quelques-uns des nouveaux registres et comment les utiliser.
- 2 - Les nouveaux modes d'adressage ou leur nouvelle syntaxe et les possibilités offertes.
- 3 - Les nouvelles instructions, au moins quelques-unes.

Rien de tout cela ne paraîtra bien neuf à ceux qui sont des virtuoses du 68020, ou aux possesseurs de TT, je le sais bien, mais je reste encore émerveillé pour ma part de ces capacités nouvelles et j'espère bien partager cela avec ceux qui, comme moi, en sont à les découvrir avec le Falcon.

QUELQUES NOUVEAUX REGISTRES

Parmi les nouveautés de l'environnement matériel du 68030, outre la PMMU, organe bien compliqué qui permet une très puissante pagination mémoire et toute une jonglerie qui va de l'adresse logique, celle du programmeur, à l'adresse physique, celle de la machine, et autorise le système d'exploitation à protéger des zones de mémoire, en mettre d'autres en commun, optimiser l'utilisation de la mémoire soit dans le sens de la rapidité soit dans celui de la place occupée, il y a, indispensable dans une situation de mémoire ainsi gérée, deux caches, l'un d'instruction, l'autre de

données. C'est ce qu'on appelle des antémémoires. Le code ou les données qui s'y trouvent sont lus par le processeur en lieu et place, et bien plus rapidement, que les équivalents en mémoire, mais en posant un problème: modifier les doublons contenus dans les caches ne modifie pas les originaux en mémoire, d'où la fin des programmes automodifiables... A moins d'inhiber les caches, bien entendu. Lors que le processeur cherche à lire l'instruction suivante à l'adresse suivante, il commence par se produire toute une série d'opérations dont le but est de chercher d'abord dans le cache, puis dans la mémoire: on aboutit à un succès-cache ou à un échec-cache.

Il existe justement deux registres de contrôle des caches étudiés pour, ainsi que d'autres opérations d'ailleurs. Ces deux registres sont CAAR et CACR.

CAAR = CAche Adress Register

CACR = CAche Control Register

C'est CACR qui permet d'invalider ou d'activer les caches. Ce registre est sur 32 bits dont voici la structure :

31 à 14 : 0

13 : WA (write allocation) si à 1, recopie en écriture possible.

12 : DBE si à 1 autorise le mode "burst" (rafales) pour le cache de données.

11 : CD (clear data) si 1, met à 0 les bits de validité du cache de données

10 : CED (clear entry data) à 1, met à 0 le bit de validité du mot long pointé par CAAR pour le cache de données

9 : FD (freeze data) à 1, gèle une mise à jour si l'essai donne lieu à un échec. La mise à jour est effectuée si succès.

8 : ED (enable data) à 1, le cache donnée est activé.

7 à 5 : 0

4 : IBE (instruction burst enable) cf. DBE

3 : CI (clear instruction) cf. CD

2 : CEI (rien à voir avec la Russie) cf. CED

1 : FI cf. FD

0 : EI cf. ED

On voit que ce champ de bits comporte deux sections, l'un pour le cache d'instruction, l'autre pour le cache de données, strictement parallèles.

Deux de ces bits retiennent toute notre attention: ED et EI, qui permettent d'activer ou d'inhiber le cache, facilement et sans risque de ne pouvoir restaurer la situation à l'identique après s'en être servi. Car le délicat mécanisme de lecture des caches, et de mise à jour, ne doit pas être perturbé imprudemment. Je suis trop paresseux pour en donner le détail aujourd'hui et ici d'ailleurs. Il faut savoir que le système d'exploitation, à chaque fois qu'il est appelé, s'occupe de valider à nouveau les caches... Nos efforts inverses n'ont de sens que tant qu'on ne met pas de d'appel système en jeu sur le Falcon.

Un autre nouveau registre est VBR. Celui-là est génial. C'est le vecteur idéal pour créer sans peine un virus sur Falcon. [NDLR: Glups ! c'est de l'humour, hein que c'est de l'humour ? HEIN ?] C'est un registre qui contient une variable et une seule. C'est parfait pour un fainéant de ma trempe. Ce registre-là contient tout simplement l'adresse de la table des exceptions, ou plus exactement un offset à ajouter à l'adresse \$0 pour l'obtenir.

Ce qui nécessite tout de même une petite explication: sur 68000, donc sur ST, la table des exceptions avait une adresse fixe. On la cherchait à compter de l'adresse \$0. Tous les quatre octets, on trouvait un vecteur de la table à chaque fois qu'il était initialisé. Rien de tel sur Falcon avec le 68030, non plus que sur TT d'ailleurs. Le registre VBR contient une adresse, celle de la table des exceptions. Mais au juste, qu'est ce que la table des exceptions ? A chaque fois que s'exécute un trap, par exemple, le système recherche dans la table des exceptions l'adresse de la routine système à laquelle il correspond. Le trap #1, autrement dit n'importe quelle fonction Gemdos, correspond à une routine mais peut être appelé par une simple instruction assembleur, sans préjudice de sa position réelle en mémoire, notion classique. Le vecteur numéro 33 (qui est le 34ème car la numérotation commence à 0) contient l'adresse du trap #1. C'est une routine qui examine le numéro de fonction gemdos, cherche l'adresse du sous-programme qui lui est dévolue dans une table, exécute, et s'achève par un RTE. On peut détourner un trap en modifiant son vecteur: ce qui permet, par exemple, d'ajouter des fonctions au système, comme le fait Mint, ou comme le font les

débugueurs. Eh bien, avec le 68030, on peut avoir autant de systèmes qu'on veut, et donc le réécrire entièrement, ou faire coexister en RAM plusieurs versions du TOS rien que par ce geste simple: inscrire une adresse dans VBR. Les deux premiers vecteurs restent cependant au même emplacement, c'est à dire aux adresses 0 et 4. Tout le reste est translatable.

Evidemment, reloger en RAM un TOS suppose de corriger toutes les références à des adresses absolues qui en seraient modifiées et pas celles qui restent vraies. Modifier VBR ne suffit pas.

Comment calculer l'adresse d'un vecteur dans ces conditions? Simple: numéro du vecteur * 4 + VBR = adresse du vecteur, qui lui-même pointe sur la routine d'exception que l'on cherche à localiser. Dans le cas du 68000, alors que VBR n'existait pas, numéro du vecteur * 4 donnait l'adresse. Du coup, mais oui, les malins ont déjà deviné, VBR contient 0 dans le cas du Falcon! Compatibilité entre les machines et les TOS oblige. Compatibilité toute relative, mais tout de même... Ainsi, ne pas oublier cette possibilité très intéressante, de sauter d'un système d'exploitation à un autre avec un seul registre et une seule instruction avec le 68030, ou de détourner d'un coup toutes les exceptions. Peut-être faudra-t-il en reparler un jour.

Bon, là, je trouve que ça fait assez pour les registres et qu'on peut passer à la suite.

LES NOUVELLES SYNTAXES

Bien des instructions déjà connues peuvent s'écrire autrement, et les assembleurs acceptent tous pratiquement les nouvelles formes, dès qu'ils acceptent le 68030, même si on écrit un programme destiné au 68000. Mais en outre, de nouveaux modes d'adressage sont apparus, qui vont faciliter la vie des programmeurs en assembleur à un point tel qu'on se demande parfois pourquoi continuer à s'embêter avec les langages de haut niveau alors qu'il est si facile d'écrire des sous-routines en 68030. Ce qui peut en limiter la portée, c'est évidemment le problème de la compatibilité. Les nouvelles formes, c'est compatible parce que le code machine n'a pas changé. Les nouveaux modes ça ne l'est pas, parce que ça introduit de nouveaux codes

machine. C'est fort intéressant, puisqu'on verra qu'on peut avoir un adressage indirect... par registre de donnée! Plus fort, qu'on peut directement accéder à une donnée par un registre qui contient l'adresse d'une variable qui contient son adresse, autrement dit d'un registre qui pointe sur son pointeur! Plus fort, qu'on peut associer tout ça et multiplier par un facteur d'échelle! Et rajouter un offset en plus! La manipulation des objets GEM en une instruction assembleur, quoi... De quoi sourire finement lorsqu'on voit des gens se faire de la purée de neurones pour leurs vieux jours en s'acharnant sur des machins 86 qu'on n'ose même pas nommer.

Le 68030, c'est le pied.

On commence par les nouvelles formes d'anciens modes d'adressage. Ainsi:

68000 → 68030

d16(An) → (d16,An)

d8(An,Xn) → (d8,An,Xn)

d16(PC) → (d16,PC)

d8(PC,Xn) → (d8,PC,Xn)

d16 est une constante sur seize bits, d8 une constante sur 8 bits, An est un registre d'adresse compris entre A0 et A7, et Xn un registre de données compris entre D0 et D7. Jusque là, on est dans le compatible. Ça facilite l'écriture, c'est tout.

On introduit tout d'abord trois nouveaux trucs: bd qui est un déplacement sur 16 ou trente deux bits (en fait sur 16 bits étendus à trente deux par extension du signe au moment du calcul de l'adresse...), od qui est un offset qui s'ajoute également en 16 bits également étendu à trente deux par extension du signe au moment du calcul de l'adresse effective, et FE qui est le facteur d'échelle, limité à 1, 2, 4, ou 8. Et on obtient les modes suivants:

([bd]) si bd = \$45E, adresse directement l'adresse de la mémoire vidéo utilisée si non nulle à la prochaine VBL. Double indirection...

([An]) Double indirection: adresse l'objet sur lequel pointe le pointeur dans An. Si An contient \$45e, va chercher à l'adresse \$45e sur quoi elle pointe par exemple \$78000 et modifie le contenu de \$78000...

([bd, An]) Double indirection sur la somme An + bd.

([bd], od) Double indirection sur bd, puis on rajoute l'offset. Si bd = \$45e et od = \$10, opère sur \$78010 dans le cas précédent.

[[An], od) Idem avec un registre adresse...

[[bd, An], od) on ajoute bd + An, on pratique la double indirection, puis on ajoute od. Si bd = \$5e, An=\$400, od=\$10, comme précédemment on atteint \$78010.

[[Xn*FE] OÙ Xn est n'importe quel registre de données et FE le facteur d'échelle 1, 2, 4 ou 8. De la même manière, double indirection. Si Xn contient, par exemple, \$45e et FE 1, on pointe sur le premier octet placé en \$78000.

[[bd, Xn*FE] même chose, la double indirection est appliquée après ajout de bd.

[[An, Xn*FE] C'est nouveau: double indirection sur An + Xn*FE

[[bd, An, Xn*FE] Même procédé.

[[bd, An, Xn*FE], od) La double indirection est appliquée entre les crochets, l'offset est ajouté ensuite.

[[An, Xn*FE], od)

[[bd, Xn*FE], od)

[[Xn*FE], od) appliquent le même principe. Ainsi, si DO = Xn et contient \$45e, si FE = 1 et od = \$10, on adresse \$78010 dans l'exemple précédent.

Mais on peut faire aussi :

[[bd], Xn*FE) OÙ l'indirection double ne porte que sur l'adresse bd, Xn*FE jouant le rôle d'un offset.

[[An], Xn*FE)

[[bd, An], Xn*FE)

[[bd], Xn*FE, od)

[[bd, An], Xn*FE, od)

[[An], Xn*FE, od)

[[bd], od)

[[PC]) OÙ l'indirection double porte sur le compteur de programme !

[[bd, PC])

[[bd, PC], od)

[[PC], od)

...On se demande quelles combinaisons ne sont pas possibles... [[bd, Xn*FE)

[[PC, Xn*FE]) et [[PC, Xn*FE], od)

[[bd, PC, Xn*FE], od)

[[PC], Xn*FE) et [[PC], Xn*FE, od)

[[bd, PC], Xn*FE, od)

On s'aperçoit que toutes les combinaisons peuvent se résumer de la manière suivante :

[[bd, An ou PC, Xn*FE, od) avec des []

marquant la double indirection sur tout ou partie sauf od, n'importe quel de ces éléments étant facultatif pourvu qu'il en reste un en dehors de od qui est lui-même facultatif.

Bien sûr, tous ces modes d'adressage ne sont pas disponibles comme opérande valable de toute instruction, et il devient difficile de s'y repérer d'emblée. Mais la plupart des instructions, même nouvelles, même les plus spécialisées, offrent une large palette des ces modes d'adressage pour au moins un de leurs opérandes: l'assembleur du 68030 est finalement très puissant, beaucoup plus puissant que celui du 68000, pour peu qu'on ose déroger à la compatibilité ou bien alors rédiger des modules distincts selon que le programme reconnaît qu'il tourne sur l'un ou l'autre processeur.

LES NOUVELLES INSTRUCTIONS

Je commence très sérieusement à faire l'objet de pressantes attaques de paresse. Aussi après tout ça, va-t-on se contenter de parler de celles qu'utilisent le listing.

D'abord, il y a MOVEC. C'est une instruction en mode privilégié, qui sert à transférer des données de ou vers un registre de contrôle, à partir d'un registre de donnée ou d'adresse. On aura donc soit:

movec Dn, Rc

movec An, Rc

movec Rc, Dn

movec Rc, An

La taille n'est pas indiquée: ce sont les 32 bits du registre de contrôle qui sont transférés.

Les registres de contrôles valables sont : VBR, CACR, CAAR, SFC, DFC, ISP, MSP, USP. Apparaît clairement la façon dont on va s'y prendre pour inhiber les caches...

Une autre instruction très intéressante est UNPK dont le pendant est PACK. Cette instruction, dont la spécialité est le décimal codé binaire, permet de réduire considérablement la traduction de l'hexa en ASCII par exemple. On va voir comment :

Deux syntaxes :

UNPK -(Ax), -(Ay),

#donnée_immédiate16bits ou UNPK

Dx, Dy, #donnée_immédiate16bits

Cette instruction agit sur un octet de la

source et deux octets de la destination. Soit UNPK DO, D1.

Au départ, on a DO = \$FFFF1020 et D1 = 00000000 avec une donnée immédiate de #\$3030. UNPK sépare en deux quartets l'octet bas de DO et répartit chaque quartet dans les quartets bas des deux premiers octets de D1, soit D1 = \$00000200, puis fait un ADD avec la donnée immédiate et on a D1 = \$00003230, soit très exactement les codes ASCII de 2 et 0. En une seule instruction, on a donc bien traduit un octet hexa en ASCII, de façon à pouvoir l'afficher ou l'imprimer. Reste que pour toute valeur supérieure à 9, il faut encore ajouter #'a'-\$39. On voit par ailleurs tout l'intérêt de l'adressage indirect pré-décrémenté pour balayer rapidement de grandes zones mémoires.

L'instruction PACK fait exactement l'opposé, avec là aussi, mais préalable au compactage, l'ajout d'une donnée immédiate sur 16 bits. Ainsi, PACK DO, D1, #\$1010 donnera, sur DO = \$FFFF2143 tout d'abord \$FFFF3153 puis D1 = \$00000013.

Voilà c'est tout pour aujourd'hui.

Hin hin, exercice! C'est un listing pour-rave, mal optimisé, qui ne se sert même pas de toutes les beautés exposées ci-dessus. L'exercice est simple: optimiser ça, et vérifier si le cookie _MCH (ou _CPU) est présent, en déduire si c'est 68000 ou 68030, et tester à chaque sous-routine utilisant les spécificités du 68030 une variable qui branche plutôt sur une routine adaptée au 68000 dans le cas où ce soit nécessaire... Quoi ? N'importe qui peut avoir tripoté le cookie-jar ? Vrai, très bien, dans ce cas, détourner le vecteur instruction illégale, et voir si une instruction 68030 provoque son exécution en se débrouillant pour retourner sans encombre au programme...

Et s'ils y en a qui doutent de l'utilité, ils ont raison. Pour ce programme là, ça ne vaut pas le coup. Mais c'est un schéma très réutilisable.

EROSTRATE

Le listing

(à assembler sous Devpac 3)

Vu le caractère général et instructif de cet article nous publions exceptionnellement le listing qui l'accompagne en entier. Néanmoins cela ne se reproduira pas tous les jours, vu sa taille MONSTRUEUSE !

opt D-,S- ; pas de table de labels
; ni d'info de déboguage.

```
c_conws equ 9
c_conin equ 1
m_shrink equ $4A
supexec equ $26
f_sfirst equ $4e
f_next equ $4f
f_open equ $3d
f_create equ $3c
f_close equ $3e
f_write equ $40
f_seek equ $42
from_end equ 2
from_start equ 0
from_current equ 1
index equ 9
```

; une macro pour exécuter un trap #14

```
xbios macro
move #1,-(sp)
trap #14
lea \2(sp),sp
endm
```

; une macro pour exécuter un trap #1

```
gemdos macro
move #1,-(sp)
trap #1
lea \2(sp),sp
endm
```

; rendre la mémoire non-programme
; au système d'exploitation

```
start
move.l 4(a7),a3 ; adresse de basepage en a3
move.l $(a3),d0 ; taille de la section TEXT
add.l $14(a3),d0 ; additionner la taille de DATA
add.l $(a3),d0 ; puis celle de BSS
add.l #$100,d0 ; + de quoi faire large
move.l #ma_pile,a7 ; un nouveau pointeur de pile
move.l d0,-(sp) ; et hop...
move.l a3,-(sp)
clr.w -(sp)
gemdos m_shrink,12
pea tbexecp ; lit en ascii la table des exceptions
xbios supexec,6
pea stop_cache ; inhibe le cache
xbios supexec,6
```

; a-t-on déjà sauvegardé une table ?

```
cherche
lea fichier,a0
move.w #0,-(sp)
move.l a0,-(sp)
gemdos f_sfirst,8
cmpl.w #0,d0
bne create ; qu'à cela ne tienne...
```

trouve

```
lea fichier,a0 ; mais quelle énième est-ce ?
move.b (index,a0),d0 ; la syntaxe 68000 est index(a0),d0
```

```
addq.b #1,d0
cmpl.b #$39,d0
ble incr ; pas plus de neuf !
move.b #$30,d0 ; sinon on repart sur la première
move.b d0,(index,a0) ; syntaxe 68000 : d0, index(a0)
gemdos f_next,2
cmpl.w #0,d0
bne create ; en créer une
bra ouvrir ; ou l'ouvrir
```

```
incr
move.b d0,(index,a0)
bra cherche
```

```
create
lea fichier,a0
clr.w -(sp)
move.l a0,-(sp)
gemdos f_create,8
cmpl #0,d0
bit ouvrir
move.w d0,id_f
bra ecrire
```

```
ouvrir
move.w #$2,-(sp)
lea fichier,a0
move.l a0,-(sp)
gemdos f_open,8
cmpl #$0,d0
bit error
move.w d0,id_f
```

```
ecrire
moveq #0,d1
lea imprim,a1
lea liste,a2
move #from_end,-(sp)
move id_f,-(sp)
clr.l -(sp)
gemdos f_seek,10
move.l d0,compte
```

ecr1 ; écrit une ligne de liste

```
movem.l d1/a1-a2,-(sp)
moveq.l #0,d2
```

ecr2

```
move.b (0,a2,d2),d0 ; syntaxe 68000 : 0(a2, d2.l), d0
addq.l #1,d2
cmpl.b #0,d0
bne ecr2
subq.l #1,d2
move.l a2,-(sp)
move.l d2,-(sp)
move.w id_f,-(sp)
gemdos f_write,12
movem.l (sp)+,d1/a1-a2
addq.l #1,d0
add.l d0,a2
add.l d0,compte
movem.l d1/a1-a2,-(sp)
```

```

move.w #from_start,-(sp)
move.w id_f,-(sp)
move.l compte,-(sp)
germdos f_seek,10
movem.l (sp)+,d1/a1-a2
movem.l d1/a1-a2,-(sp)
lea (0,a1,d1)*8,a1 ; puis une ligne de vecteurs
; noter la syntaxe
; avec facteur d'échelle

```

```

move.l a1,-(sp)
move.l #8,-(sp)
move id_f,-(sp)
germdos f_write,12
add.l d0,compte
move #from_start,-(sp) ; puis une fin de ligne
move id_f,-(sp)
move.l compte,-(sp)
germdos f_seek,10
move.l #crif,-(sp)
move.l #2,-(sp)
move id_f,-(sp)
germdos f_write,12
add.l d0,compte
movem.l (sp)+,d1/a1-a2
addq.l #1,d1 ; on enregistre une ligne

```



```

movem.l d1/a1-a2,-(sp)
move #from_start,-(sp)
move id_f,-(sp)
move.l compte,-(sp)
germdos f_seek,10 ; on ajuste le pointeur de fichier
movem.l (sp)+,d1/a1-a2
cmpi #256,d1 ; tant que c'est pas fini...
bit ecr1

```

fermer

```

move id_f,-(sp)
germdos f_close,4
pea boot_cache
xbios supexec,6
bra fin

```

error

```

lea err_fois,a0
addq #1,a0
pea err_mes
germdos c_conws,6

```

```

germdos c_conin,2
lea err_fois,a0
move (a0),d0
cmpi #10,d0
ble cherche
pea boot_cache
xbios supexec,6
bra fin

```

stop_cache

```

movec CACR,d0
oriw #0808,d0
movec d0,CACR
rts

```

boot_cache

```

movec CACR,d0
andi.w #0F7F7,d0
movec d0,CACR
rts

```

tbexcep

```

movem.l d0-d1/a0-a2,-(sp)
lea table,a1 ; où lire les vecteurs d'exception
move.l #0,a2
moveq.l #0,d1

```

prec

```

move.l (0,a2,d1)*4,d0
move.l d0,(a1)+
addq.l #1,d1
cmpil #2,d1
bne prec ; offset relatif de deux
movec.l VBR,d2 ; VBR indexe la table
add.l d2,a2 ; a2 pointe le 3 vecteur

```

prec2

```

move.l (0,a2,d1)*4,d0
move.l d0,(a1)+ ; une de plus dans table
addq.l #1,d1
cmpil #100,d1
bne prec2 ; jusqu'au bout
subq.l #1,d1 ; on repointe le dernier vecteur
lea imprim+512*4,a2 ; second espace où réécrire

```

ascii

```

; tout entre 0 et 9 se retrouve en ascii
; agit sur l'octet : on répète 4 fois pour un vecteur
REPT 4
unpk -(a1)-(a2),#$3030
ENDR
dbf d1,ascii
lea imprim,a1
moveq.l #0,d1
move.l #8,d2
; on corrige pour A, B, C, D, E, F

```

retouche

```

move.l (0,a1,d1)*4,d0
cmpi.b #39,d0
ble suite1
addi.b #7,d0 ; ici

```

suite1

```

; on balaye un octet à chaque tour

```

```

roll d2,d0
cmpi.b #$39,d0
ble suite2
addi.b #$7,d0

suite2
roll d2,d0
cmpi.b #$39,d0
ble suite3
addi.b #$7,d0

suite3
roll d2,d0
cmpi.b #$39,d0
ble ouf
addi.b #$7,d0

ouf
roll d2,d0
move.l d0,(O,a1,d1*4)
addq.l #1,d1
cmpi.w #511,d1
ble retouche
movem.l (sp)+,d0-d1/a0-a2
rts

fin
gemdos 0,2

;*****

data
even

liste dc.b "RESET : valeur initiale de ISP = ",0
dc.b "RESET : valeur initiale de PC = ",0
dc.b "Erreur bus = ",0
dc.b "Erreur d'Adresse = ",0
dc.b "Instruction illégale = ",0
dc.b "Division par zéro = ",0
dc.b "Instruction CHK, CHK2 = ",0
dc.b "Instruction TRAPV, TRAPcc, cpTRAPcc = ",0
dc.b "Violation de privilège = ",0
dc.b "Trace = ",0
dc.b "Code %1010 ou ligne A = ",0
dc.b "Code %1111 ou ligne F = ",0
dc.b "Réservé = ",0
dc.b "Violation de protocole coprocesseur = ",0
dc.b "Erreur de format = ",0
dc.b "Interruption non initialisée = ",0
REPT 8
dc.b "Réservé = ",0
ENDR
dc.b "Interruption parasite = ",0
dc.b "Interruption auto-vectorisée niveau 1 = ",0
dc.b "Interruption auto-vectorisée niveau 2 = ",0
dc.b "Interruption auto_vectorisée niveau 3 = ",0
dc.b "Interruption auto-vectorisée niveau 4 = ",0
dc.b "Interruption auto-vectorisée niveau 5 = ",0
dc.b "Interruption auto-vectorisée niveau 6 = ",0
dc.b "Interruption auto vectorisée niveau 7 = ",0
dc.b "TRAP # 0 = ",0
dc.b "TRAP # 1 = ",0
dc.b "TRAP # 2 = ",0
dc.b "TRAP # 3 = ",0
dc.b "TRAP # 4 = ",0
dc.b "TRAP # 5 = ",0
dc.b "TRAP # 6 = ",0
dc.b "TRAP # 7 = ",0
dc.b "TRAP # 8 = ",0
dc.b "TRAP # 9 = ",0
dc.b "TRAP #$A = ",0
dc.b "TRAP #$B = ",0
dc.b "TRAP #$C = ",0
dc.b "TRAP #$D = ",0
dc.b "TRAP #$E = ",0
dc.b "TRAP #$F = ",0
dc.b "FPCP condition non ordonnée du copro = ",0
dc.b "FPCP résultat inexact = ",0
dc.b "FPCP division par zéro = ",0
dc.b "FPCP dépassement vers le bas = ",0
dc.b "FPCP Erreur d'opérande = ",0
dc.b "FPCP Dépassement vers le haut = ",0
dc.b "FPCP Signal NAN = ",0
dc.b "Réservé = ",0
dc.b "Erreur de configuration PMMU = ",0
dc.b "Opération 68851 illégale = ",0
dc.b "Violation de niveau d'accès 68851 = ",0
REPT 5
dc.b "Réservé = ",0
ENDR
REPT 192
dc.b "Interruption utilisateur = ",0
ENDR
even

fichier
dc.b "a:\table_,$30,.lst",0

even
err_mes
dc.b "Ca marche pas : vérifiez la disquette",13,10
dc.b "ou mettez une disquette formatée en A:",0

crif dc.b 13,10,0

even
dc.b $bd,"F. Jacquemin - 1993 - La Charite sur Loire ",0

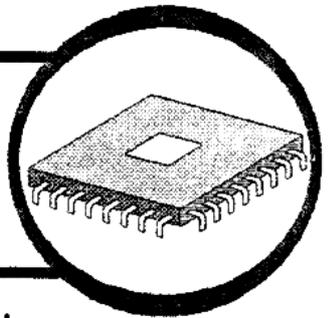
bss

even
table ds.l 256
imprim ds.l 512
id_f ds.w 1
err_fois ds.w 1
compte ds.l 1
ds.l $60
ma_pile ds.l 1

end

```

LEÇON DE SON



COMMENT PISSER DANS SON FALCON POUR EN TIRER DU SON !

Grand titre pour une série d'articles qui se propose et qui, je l'espère vous permettra de comprendre enfin l'utilisation d'éléments tel que la matrice du Falcon. Le but est ici de comprendre le fonctionnement de chacune des nouvelles fonctions Xbios mettant en oeuvre le son. Nous allons commencer par détailler ces fonctions, puis nous aborderons leur implementation en C et en GFA, puis nous terminerons par l'utilisation de tout ce que l'on aura appris (enfin je l'espère) en les mettant en oeuvre dans un programme du style STreplay.

LA MATRICE

Késako ? La matrice du Falcon pourrait être comparée à votre chaîne hifi et plus particulièrement à son préampli. Nous avons d'un côté tous les éléments qui la composent tel que magnéto, laser etc, et tous les éléments vers quoi nous pouvons diriger le son (enceintes, magnéto d'enregistrement, equaliser) Dans le cas de la matrice du Falcon nous avons.

En entrée

- ADC
- EXTINP
- DSPXMIT
- DMAPLAY

En sortie

- DAC
- EXTOUT
- DSPRECV
- DMAREC

Dans la figure ci-contre (si le rédac à fait son boulot) vous pouvez voir l'organisation logique de cette matrice, comme vous pouvez le remarquer tout les éléments peuvent être reliés à tous les autres éléments.

- L'ADC permet de rentrer dans votre Falcon toute source sonore

analogique externe, en fait c'est comme si sur votre ST vous aviez une cartouche STreplay en standard dans votre machine.

- DSPXMIT permet de rentrer dans votre Falcon un signal sonore en provenance du DSP (voir explication plus loin)

- DMAPLAY idem DSPXMIT mais cette fois ci c'est en provenance du canal DMA.

- LE DAC permettant la conversion d'un son digital vers un son analogique compréhensible par toute chaîne hifi ou enceinte extérieur. En fait tout le son audible passe automatiquement par là sinon il ne pourrait être écouté (A moins d'être très riche et de posséder un ampli digital et des enceintes digitales).

- DSPRECV permet d'envoyer au dsp un son pour traitement.

- DMAREC pareil que le DSPRECV, mais vers le canal DMA.

Ce qu'il y a de très intéressant avec la matrice, c'est que tout est permis, tout est automatique, et que le Falcon se charge tout seul comme un grand du mixage. Par exemple un délire comme celui qui suit est autorisé.

1/ Connexion du ADC-DAC permettant un écoute de toute source sonore reliée au Falcon (Option LISTEN de STreplay)

2/ Connexion du ADC-DSPRECV pour que le DSP puisse recevoir des données pour un traitement tel qu'un

harmoniseur ou une réverbération.

3/ Connexion du DSPXMIT-DMAREC. Là on sort le son traité par le DSP vers le DMA pour qu'il puisse être échantillonné puis rejoué avec 5 secondes de décalage.

4/ Connexion du DMAPLAY-DAC Phase finale permettant d'écouter le son traité par le DSP puis rejoué avec un décalage.

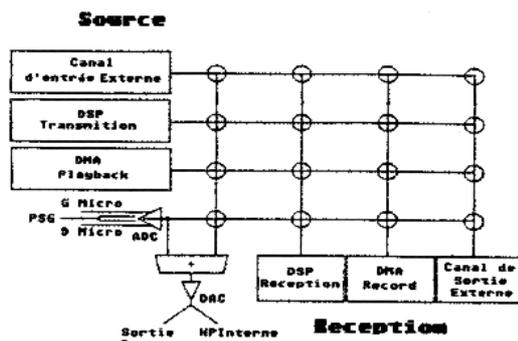
Dans cette description d'entrées/sorties je passe volontairement sur les deux external (in et out) nécessitant une adjonction hard pour leur utilisation (Port SCI du DSP).

Des questions ? Si oui écrivez au journal et j'y répondrai dans le prochain numéro. En attendant je vais vous décrire rapidement les nouvelles fonctions Xbios. Si vous remarquez des "trous" dans la numérotation c'est normal car je passe volontairement sur certaines fonctions buggés ou hors sujet mais ne vous inquiétez pas j'y reviendrais.

XBIOS(128) LOCKSND.

Permet de bloquer le système sonore. Cette fonction n'est en aucun cas un véritable blocage mais une information que vous donnez au système disant que vous aimeriez utiliser le son. Si une autre application a bloqué le son avant, vous ne devrez pas utiliser le son. Dans le cas de l'utilisation de SAM ou AFM, toute modification de la matrice mettrait un joyeux bordel. C'est pour ça qu'il faut prendre l'habitude d'utiliser cette fonction avant toute utilisation pour vérifier votre bon droit (Programmons propre dans la mesure du possible). Pourquoi dans la mesure du possible? Ben tout simplement parceque des TOS tels que certains 4.01 (pas tous) ont la facheuse tendance de renvoyer n'importe quoi, ce qui fait que théoriquement on est jamais sur d'être le seul.

Le retour de cette fonction, si tout c'est bien passé, est TRUE.



XBIOS(129) UNLOCKSND.

Permet de débloquent le système sonore. Attention ici, la réponse normale est 0.

XBIOS(130) SOUNDCMD(mode,data)

Permet de définir des préférences. Avec "mode" on choisit une action et on transmet les paramètres dans "data". En retour vous aurez toujours l'état de "mode" après votre modification. Normalement le paramètre transmis et la valeur en retour doivent être égaux mais allez savoir, des fois que.

MODEDATA

0 Atténuation du volume gauche par incrément de -1.5 db

1 Atténuation du volume droit par incrément de -1.5 db

2 Atténuation du gain de l'entrée gauche par incrément de 1.5 db

3 Atténuation du gain de l'entrée droite par incrément de 1.5 db

4 Choisit la destination du signal 16 Bits entre l'ADC, la matrice ou les deux

Bit 7 6 5 4 3 2 1 0

0 0 0 0 0 0 M A

5 Permet de choisir pour la voie droite et gauche si c'est l'ADC ou le PSG. Si le bit est à 0 c'est l'ADC, et à 1 c'est le PSG

Bit 7 6 5 4 3 2 1 0

0 0 0 0 0 0 L R

6 Permet de définir le prédiviseur de fréquence lors de l'utilisation de DEVCONNECT

0 - Non utilisé

1 - Division par 640

2 - Division par 320

3 - Division par 160

XBIOS(131)

SETBUFFER(reg,begaddr,endaddr)

Cette fonction permet de donner une adresse de début et de fin où seront lus ou écrit les données lors d'un DMAPLAY ou d'un DMAREC

reg : 0 Définition d'un buffer de playback

1 Définition d'un buffer d'enregistrement

Exemple :

Taille%=100000

a%=MALLOC(taille%)

~XBIOS(131,1,La%,La%+taille%)

XBIOS(132) SETMODE(mode)

Défini le type de donnée pour l'enregistrement et la lecture.

mode : 0 8 Bits STEREO

1 16 Bits STEREO

2 8 Bits MONO

XBIOS(136) BUFFOPER(mode)

Défini l'opération à effectuer avec le buffer précédemment indiqué avec SETBUFFER

Bit 7 6 5 4 3 2 1 0

RR RE PR PE

RR Record en boucle

RE Flag record

PR Play en boucle

PE Flag play

Mode =0 permet d'interrompre toute les opérations en cours.

XBIOS(139)

DEVCONNECT(src,dst,srclk,prescale,protocol)

Sûrement la fonction la plus puissante car toutes les opérations sur la matrice se font avec elle.

scr -> Source sonore

3 ADC

2 EXTINP

1 DSPXMIT

0 DMAPLAY

dst -> Destination

8 DAC

4 EXTOUT

2 DSPRECV

1 DMAREC

srclk -> Horloge utilisé par la source

0 Interne 25,175 Mhz

1 Externe (Utilisation avec EXTINP et EXTOUT)

2 Interne 32 Mhz

prescale -> La fréquence d'échantillonnage est déterminé par la valeur de l'horloge source divisé par 256 puis divisé par ce paramètre.

Protocol -> Permet d'activer ou de désactiver le handshaking.

XBIOS(141) Buffptr(pointer)

Pointer est l'adresse d'une structure de 4 mots longs qui contiendront lors du lancement d'une opération avec BUFFOPER la position courante dans le buffer d'enregistrement ou de lecture. Voir l'exemple en GFA pour son utilisation est ses contraintes.

Structure (long) Play buffer

(long) Record buffer

(long) Reservée

(long) Reservée

L'UTILISATION EN GFA

Je vous donne ici à titre d'exemple un petit oscillo qui n'est là uniquement que pour vous montrer l'utilisation de la plupart des fonctions XBIOS.

Pour commencer nous allons nous créer toutes les procédures et fonctions XBIOS.

Procédure SOUNDCMD(mode,data)

~XBIOS(130,mode,data)

Return

Procédure

DEVCONNECT(src,dest,srclk,prescale,protocol)

~XBIOS(139,src,dest,srclk,prescale,protocol)

Return

Procédure SETMODE(mode)

~XBIOS(132,mode)

Return

Procédure SNDSTATUS(reset)

~XBIOS(140,reset)

Return

Procédure

SETBUFFER(reg,begaddr,endaddr)

~XBIOS(131,reg,L,begaddr,L,endaddr)

Return

Procédure BUFFOPER(mode)

~XBIOS(136,mode)

Return

Procédure BUFFPTR

~XBIOS(141,L,V:buf%(0))

Return

Function LOCKSND

a=XBIOS(128)

RETURN a

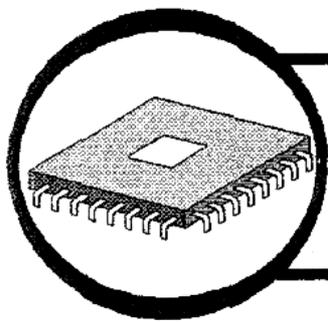
Endfunc

Procédure UNLOCKSND

~XBIOS(129)

Return

A ceci nous rajouterons une procédure init permettant de se créer des constantes.



ROUTINES

GRAPHISMES PREMIERE PARTIE

Les routines, quel nom barbare pour une chose qui a engendré d'innombrables heures de recherche, dont sûrement pas mal de nuits blanches. La précision, la qualité, et la rapidité de celles-ci sont à la base de la réussite de programmes tels que les démos, les jeux ou autres programmes de dessin. Cette série d'articles aura pour but de vous amener à réaliser vos propres routines. Vous n'aurez en revanche jamais de routines complètes prêtes à l'emploi, car beaucoup de personnes risquent de rager en voyant tout le monde disposer de belles routines rapides, alors qu'ils ont passé de nombreux mois à élaborer les leurs. De plus, une routine optimisée n'est jamais universelle. Chaque programme a besoin de routines spécifiques.

Maintenant que la présentation est faite, il faut attaquer le problème par un bout. Le langage utilisable pour l'élaboration de celle-ci sera de préférence l'assembleur, et peut-être le C. En revanche, le GFA basic ne peut vraiment convenir puisqu'il ne peut en aucun cas réussir à générer des routines plus rapide que celles fournies par le système. Cependant, il y aura probablement quelques applications à réaliser avec celui-ci. De plus, il est fréquent que les programmeurs en assembleur vérifient d'abord la validité de leurs algorithmes en GFA basic avant de se lancer dans une code en assembleur très lourd.

Maintenant que vous avez choisi votre langage de support (en fait vous n'avez pas énormément de choix), je dois m'assurer que vous connaissez bien certaines bases, telles que l'organisation de la mémoire vidéo, et bien sur les notations hexadécimales et binaires. L'apprentissage complet de ces bases serait très long, voilà déjà les équivalences entre les premières valeurs:

hexadécimal	binaires	décimal
\$0	%0000	00
01	\$1	%0001
02	\$2	%0010
03	\$3	%0011
04	\$4	%0100
05	\$5	%0101

06	\$6	%0110
07	\$7	%0111
08	\$8	%1000
09	\$9	%1001
10	\$A	%1010
11	\$B	%1011
12	\$C	%1100
13	\$D	%1101
14	\$E	%1110
15	\$F	%1111

quelques exemples:

$$16 = \$10 = \%10000$$

$$216 = \$114 = \%100010100$$

Pour convertir d'héxa (abréviation usuelle d'hexadécimal) en binaire, il suffit d'associer chaque digit du nombre en héxa, aux 4 bits lui correspondant dans la table, et le tour est joué. Pour convertir d'héxa en décimal, il suffit d'appliquer cette formule:

$$\text{nbre_héxa} = \$WXYZ$$

$$\text{nbre_décimal} = Z*Y*16 + X*256 + V*4096$$

$$= Z*\$1 + Y*\$10 + X*\$100 + V*\$100$$

$$= Z*16^0 + Y*16^1 + X*16^2 + V*16^3$$

De nombreuses calculatrices font les conversions, mais il vous faudra absolument maîtriser l'héxa et le binaire, ou sinon vous perdrez votre temps à taper sur votre calculatrice, et vous serez très vite agacé.

Le Falcon 030 possède deux différents types d'affichage: avec ou sans plan de bits. C'est à dire surtout le mode 256 couleurs et le mode dit "near true color" (65536). En mode "near true color", l'ordinateur ira un peu moins vite qu'en mode 256 couleurs, les données seront deux fois plus volumineuses, mais dans bien des cas les algorithmes d'affichage seront bien plus simples et bien plus rapides, grâce à la simplicité d'organisation de ce mode.

Voilà l'organisation de la mémoire vidéo en mode "near true color": Chaque pixel est codé sur 16 bits, soit 2 octets ou un mot (word, unsigned int):

$$\%RRRRRVVVVVBBBBB$$

Chaque pixel est défini par un niveau de rouge, de vert, et de bleu, lorsque RRRRR=%11111, le niveau de rouge est maximum, lorsque RRRRR=%00000, le niveau de rouge est minimum. Vous appliquez le même raisonnement pour le vert et le bleu. L'écran doit contenir la

définition de chaque point. Ceux-ci sont classés d'abord suivant leurs valeurs en Y, puis leurs valeurs en X (origine en haut à gauche). Evidemment, pour clarifier tout ça, prenons un petit exemple. Supposons que nous soyons en 320*200, et qu'il y ait sur un écran noir uniquement une ligne verticale jaune partant de (3,2) et allant en (3,4). Voilà ce que contiendrait l'écran: La couleur jaune est composée de rouge et de vert (bon à savoir), ce qui nous donne: %11111111100000=\$ffe0 (rappelez vous de la conversion bin-héxa).

Présentation: (offset par rapport au début de l'écran, en octets):

$$\text{\$val} \quad (0):0,(2):0, \dots ,(638):0$$

$$(640):0,(642):0, \dots ,(1278):0$$

$$(1280):0,(1282):0,(1284):0,(1286):0,\$ffe0,(1288):0, \dots ,(1918):0$$

$$(1920):0,(1922):0,(1924):0,(1926):0,\$ffe0,(1928):0, \dots ,(2558):0$$

$$(2560):0,(2562):0,(2564):0,(2566):0,\$ffe0,(2568):0, \dots ,(3198):0$$

$$(3200):0,(3202):0, \dots ,(127998):0$$

Donc, pour résumer, voilà une petite formule pour savoir où mettre un point de coordonnées (XY), sur un écran en Near True Color de largeur LAR, et de hauteur non définie (car pas utile pour la formule): offset = (X*Y*LAR)*2

Maintenant, dites vous bien que ce que vous avez vu était de la rigolade. Le mode en plans de bits est bien plus compliqué. De ce fait, nous allons l'aborder progressivement, en commençant par le plus simple, le mode monochrome (2 couleurs). En fait, dans ce mode, un pixel est soit allumé (couleur 1), soit éteint (couleur 0). Chaque pixel correspond un bit qui est soit à 1 (allumé, couleur 1), soit à 0 (éteint, couleur 0). Les bits sont classés, comme pour le mode NTC, d'abord suivant leur valeur en Y, puis en X, ce qui nous donne la formule suivante pour connaître le numéro de bit: numéro = X*Y*LAR. Voilà, ça à l'air simple ? Et bien non, ce n'est pas tout. La mémoire est organisée d'octets, et pas de bits. Chaque octet, comme l'indique son nom, est composé de 8 bits. Nous avons donc maintenant ces deux formules: offset = (X*Y*LAR)/8
numéro_de_bit_provisoire = (X MOD 8)+1
(V MOD D) est le reste de la division

entière V/D. Mais pourquoi provisoire ? Le bit de gauche d'un octet est le 8ème bit, et pas le 1er. Nous devons donc faire la conversion: $\text{numéro_de_bit} = 9 - \text{numéro_de_bit_provisoire} = 9 - (X \text{ MOD } 8) - 1 = 8 - (X \text{ MOD } 8)$

Passons aux modes avec plusieurs plans de bits. Nous allons prendre le mode 16 couleurs. 4 plans de bits, $2^4=16$, $4=\text{Ln}(16)/\text{Ln}(2)$ (ar!arf! 2 logarithmes népériens, et ça fait de suite plus technique). Pour commencer, il va falloir décomposer l'index de couleur en 4 bits. $\text{couleur}=\%VXYZ$ Nous allons commencer par déterminer l'offset pour un écran monochrome. Mais en fait, il nous faut considérer que nous avons 4 écrans monochromes, et donc positionner sur chacun des 4, le bit à 0 ou à 1, suivant la valeur de V, X, Y et Z. Si vous positionnez ces 4 écrans les uns à la suite des autres, vous aurez un joli écran pour le mode 16 couleurs de l'amiga. Sur Atari, on voit les choses différemment (c'est surtout pour se contenter d'un shifter moins complexe). Le véritable écran est composé de 4 écrans monochromes imbriqués les uns dans les autres. On dit aussi qu'ils sont entrelacés. Voilà la composition:

premiers octets du 1er écran: A,B,C,D,E,F
 premiers octets du 2ème écran: G,H,I,J,K,L
 premiers octets du 3ème écran: M,N,O,P,Q,R
 premiers octets du 4ème écran: S,T,U,V,W,X
 premiers octets du véritable écran 16 couleurs: S,T,M,N,G,H,A,B,U,V,O,P,I,J,C,D,W,X,Q,R,K,L,E,F
 Vous avez compris? Ils sont insérés par 2 octets.

Pour la formule finale, nous travaillerons sur des mots (16 bits). L'offset est toujours indiqué en octets.
 $N = \text{numéro de plan de bits de l'index si coul}=\%VXYZ$
 V est le 4ème bit
 $\text{Offset} = ((X+Y*LAR)/16)*8+(N-1)*2$
 $\text{numéro_de_bit} = 16 - (X \text{ MOD } 16)$

Pour le mode 256 couleurs, c'est la même chose, sauf qu'il y a 8 plans de bits au lieu de 4. soit P le nombre de plans de bits

$\text{Offset} = ((X+Y*LAR)/16)*2*P+(N-1)*2$
 $\text{numéro_de_bit} = 16 - (X \text{ MOD } 16)$

Pour bien faire la synthèse, voilà 2 programmes utilisant des routines d'affichage de point en assembleur pour les modes NTC et 256 couleurs, qui ne devraient pas être difficiles à convertir en C, voir en GFA basic. Dans ce dernier cas, il n'y a aucun intérêt, car la fonction

"plot" ira surement plus vite. Ces routines ne sont bien sur pas optimisées, cela vous donnera l'occasion de commencer à réfléchir sur des méthodes d'optimisation. Ce sera tout pour ce premier article. Il vaut mieux vous laisser le temps de bien digérer l'organisation de la mémoire vidéo, pour attaquer dans les prochains articles des routines plus complexes, telles que l'affichage de sprites, les scroll-texts, les lignes, les polygones pleins, et sûrement bien d'autres.

Si vous avez des questions, passez par FAUCONtact qui transmettra.

; affichage de points en 640x256 NTC ; n'oubliez pas de sélectionner le bon mode video

```

LAR=640 ;largeur de l'écran
move.w #3,-(sp) ;Logbase
trap #14 ;adresse de début
addq.l #2,sp ;de l'écran
move.l d0,phy
moveq #0,d5
move.w #200-1,d7 ;d7=Y
nouvelle_ligne:
move.w #320-1,d6 ;d6=X
nouveau_point:
bsr point
dbra d6,nouveau_point
addq.w #1,d5
dbra d7,nouvelle_ligne
move.w #7,-(sp) ;Crawcin
trap #1 ;juste pour admirer
addq.l #2,sp
clr.l -(sp)
trap #1
point: ;d5=couleur d6=X d7=Y
movem.l d6-a0,-(sp)
move.l phy,a0
add.w d6,a0
add.w d6,a0
mulu.w #LAR*2,d7
add.l d7,a0
move.w d5,(a0)
movem.l (sp)+,d6-a0
rts
BSS
phy:
ds.l 1
END
    
```

; affichage de points en 640x256 couleurs ; n'oubliez pas de sélectionner le bon mode video

```

LAR=640 ;largeur de l'écran
PLANS=8
move.w #3,-(sp) ;Logbase
trap #14 ;adresse de début
addq.l #2,sp ;de l'écran
move.l d0,phy
moveq #0,d5
move.w #200-1,d7 ;d7=Y
nouvelle_ligne:
move.w #320-1,d6 ;d6=X
nouveau_point:
bsr point
dbra d6,nouveau_point
addq.w #1,d5
dbra d7,nouvelle_ligne
move.w #7,-(sp) ;Crawcin
trap #1 ;juste pour admirer
addq.l #2,sp
clr.l -(sp)
trap #1
point: ;d5=couleur d6=X d7=Y
movem.l d0/d1/d6-a0,-(sp)
move.l phy,a0
mulu.w #LAR,d7
ext.l d6 ;d6 est-il bien sur
32 bits ?
add.l d6,d7
divu.l #16,d7
mulu.l #2*PLANS,d7
add.l d7,a0
and.w #15,d6 ;modulo à 16...
moveq #15,d7
sub.w d6,d7 ;15-n (le 1er bit
est le 0)
moveq #PLANS-1,d0 ;nombre de
plans à traiter
moveq #0,d1 ;numéro de plan
nouveau_plan:
move.w (a0),d6
bstst d1,d5
bne.s mettre
bcir d7,d6
bra.s ok
mettre:
bset d7,d6
ok:
move.w d6,(a0)+
addq.w #1,d1
dbra d0,nouveau_plan
movem.l (sp)+,d0/d1/d6-a0
rts
BSS
phy:
ds.l 1
END
    
```



INITIATION AU C

PREMIERE PARTIE

La dernière fois je vous ai montré pourquoi il vaut mieux développer en C/ Assembleur sur FALCON. Aujourd'hui, nous allons passer à la pratique. Tout d'abord nous allons voir comment faire une bibliothèque (en C ou en Assembleur) afin de l'utiliser en C, et ensuite nous commencerons l'étude du C proprement dite.

CREATION D'UNE BIBLIOTHEQUE :

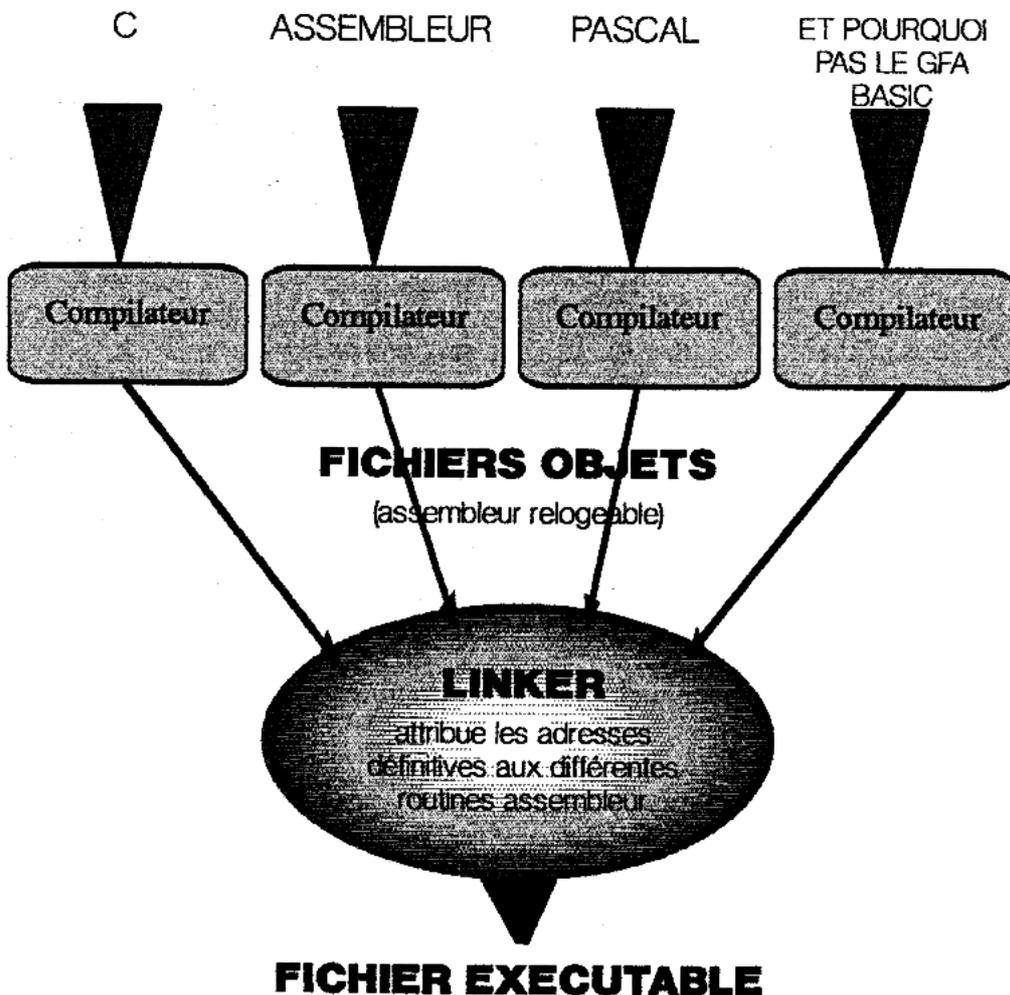
Pour bien comprendre ce qu'est une bibliothèque en C, il faut regarder ce qui se passe quand on compile puis linke un source C afin d'obtenir un exécutable. Le fait de compiler un programme C (ou tout autre langage) produit un fichier dit objet (Dont l'extension est *.obj ou *.o ...) dans lequel le programme sera transformé en assembleur, mais toutes ces adresses absolues, et ces zones de données seront relogeables (C'est à dire qu'à ce stade le programme n'a aucune obligation quand à l'emplacement en mémoire où il s'exécutera. Un fichier objet n'est donc pas exécutable). Pour obtenir un

programme exécutable il faut linker le (ou les) fichier(s) *.obj, c'est à dire lier ce (ou tous ces) fichier(s) pour en obtenir plus qu'un seul (*.prg *.tos *.app *.ttp *.acc ...).

Le gros intérêt est que l'on peut théoriquement linker ensemble des programmes ayant été compilés pour n'importe quel langage, si les fichiers "objet" ont le même format. En fait il devient facile d'imaginer qu'un de ces fichiers "objet" est le fichier principal, alors que les autres ne sont rien d'autre que des bibliothèques utiles à celui-ci. En fait chaque bibliothèque est constituée de fonctions, (des sous programmes en quelque sorte) et dans certains cas de variables globales (c.a.d. valables tout au long du programme et qui peuvent être organisées en structure...). Toutes les fonctions (et les zones de données) qui pourront être utilisées par l'utilisateur, devront être déclarées comme tel avant, la compilation (le plus souvent déclaration en extern, ou export ...). Ce qui permettra lors de la compilation de les différencier de façon à ce que le linker sache quoi lier avec quoi. En C pour permettre la compilation, toutes les fonctions qui n'existent pas dans le fichier à compiler devront être déclarées (en extern ou non) c'est le but des fichiers *.H.

En PureC, on a la possibilité de faire un fichier projet (*.PRJ) qui n'est rien de plus qu'un fichier texte dans lequel on indique au compilateur et au linker comment ils doivent compiler et linker le programme principal (Noté main() en C) [NDLR: les hiéroglyphes { et } sont des accolades.] ainsi que les bibliothèques. Pour rendre exécutable un programme C simple qui n'utilise que les bibliothèques classiques livrées avec le PureC il faut utiliser DEFAULTPRJ. Sinon on peut faire un fichier MAKE... ce qui est équivalent.

Par contre les fichiers *.OBJ ne sont pas tous au même format, il n'est par



exemple pas possible de linker un fichier objet de PureC avec un fichier objet fait avec GENST ou même avec le LATTICE C. Pour le GFA Basic/GFA Assembleur je n'ai pas essayé et je pense que ce ne doit pas être le même format que le PureC, ni celui de GENST...

LE C

Dans cette première partie nous allons parler des types de données les plus importants.

Mais pour commencer il faut parler de la fonction principale du C :

```
int main(int argc, char *argv[])
```

*/*argc donne le nombre de paramètres*/*

*/*argv donne la liste des paramètres*/*

```
{
.../ *ici le reste du programme*/
return(0); /*retour au systeme avec un
           code d'erreur*/
}
```

Voici la définition complète de la fonction main, correspondant au programme principal.

Tout programme C doit contenir une et une seule fonction "main" avec ou sans paramètre. On utilisera le plus souvent qu'une seule partie de sa définition :

```
void main(void)
{
.../ *ici le reste du programme*/
}
```

Void signifiant ici: aucun paramètre

Pour programmer en C il faut connaître les types de données, leur utilité, et la façon dont on les utilise. Il faut d'abord savoir que toutes les variables utiles à une fonction doivent obligatoirement être définies au début de celle-ci (variables locales), ou en dehors de toute fonction (variables globales) avec le type de données auquel elles correspondent.

Voici un tableau qui récapitule les types de données les plus courants :

int entier 16 bits
char entier 8 bits ou caractère
long entier 32 bits
float réel simple précisions
double réel double précisions
void rien aucun type

Tous ces types sont normalement signés, pour les déclarer non signés il suffit de faire précéder la déclaration par unsigned. Une variable lorsqu'elle est définie à une valeur quelconque. Pour indiquer au compilateur que cette variable est déclarée dans un autre fichier objet (bibliothèque) il suffit de faire précéder la déclaration par extern.

Voici quelques exemples de travail avec des tableaux et des pointeurs de données :

int tab[10]; définit un tableau de 10 entier 16 bits.

tab[5]=0; met 0 dans la 5ème case du tableau.

int *ptab; définit un pointeur pointant sur un entier.

ptab=&tab[0]; initialise le pointeur à l'adresse du début du tableau tab.

ptab=tab; la même chose que précédemment (on verra cela plus tard)

***ptab=0;** met 0 à l'adresse de ptab.

***(ptab+5)=0;** si ptab pointe sur tab, c'est la même chose que tab[5]=0;

int *ptab[10]; définit un tableau de pointeurs sur des entier.

ptab[0]=&tab[0]; initialise le premier pointeur du tableau à l'adresse du tableau.

Il existe aussi les structures de données, les pointeurs de structure, les pointeurs de fonction... bref on peut tout faire en C mais on verra tout cela plus tard en détail (avec leurs correspondances en assembleur).

On peut remarquer qu'il n'y a pas de type qui définisse les chaînes de caractère car le C fonctionne comme en assembleur : une chaîne est une zone de mémoire terminée par un zero. Pour finir voici un programme C tout simple qui initialise divers types de variables. (les commentaires en C sont entre */* commentaire */*)

```
/*on inclus la déclaration de printf définie dans stdio.h*/
#include <stdio.h>

void main(void)
{
/*déclaration d'un entier i, et d'un tableau de 10 entiers itab*/
int i=0,itab[10];

/*declaration d'un reel pi, et d'un tableau de 10 reel ftab*/
float pi=3.14,ftab[10];

/*déclaration et initialisation d'un tableau de char faisant office de chaîne
de caractères, dans ce cas là pas besoin d'indiquer la taille du tableau, le C
le calcule tout seul (et oui c'est moderne)*/
char txt[]="vive le Falcon";

for(i=0;i<10;i++) /*for i=0 to 10 en basic */
{
itab[i]=i; /*la case i de itab est initialisée à i*/

ftab[i]=pi*i; /*la case i de ftab est initialisée à pi*i */

/*on affiche tout avec printf mais on verra cela plus tard */
printf("%s %d %f",txt,itab[i],ftab[i]);
}

/*c'est fini*/
}
```

On peut remarquer quelque chose d'important dans ce listing, chaque instruction en C se termine par un ; et on peut faire ce que l'on veut avant ce ; . Par contre **for** ne se termine pas par un ; car il concerne un bloc d'instruction caractérisé par { } .

Conclusion :

Voilà c'est terminé, pour aujourd'hui la prochaine fois on entrera plus dans le détail de la programmation en C, mais attention on ne verra pas tout du C, et comme il me tient à coeur de vous montrer comment faire des programmes C/Assembleur, je vous conseille de vous procurer un livre sur le C (comme "C facile" de marabout, pour 39f on ne se ruine pas) même s'il ne parle pas du FALCON mais du PC de toutes façons, le C est le même... car sinon avec un FAUCONTACT tous les deux mois, on en a pour 10 ans pour tout faire sur le C. [NDLR: on a toute la vie!]

BESTBUG

FLABBERGASTED

ISHAR 2, LE RETOUR

Alors c'était bien !? Moi je me suis amusé comme un fou. Il faut dire que jusqu'à maintenant sur Falcon on s'amuse d'un rien. Mais c'est vrai que Ishar 2 est tout de même très attractif. Et comme promis ce mois-ci, je vous livre des patchs de ce jeu que certains approuvent, que d'autres redoutent et que d'autres réprouvent (Hein Splash!). Notez que ceux qui réprouvent une solution aussi rapide d'un jeu qui a, à peine deux mois, ont peut être raison

mais j'avais fait une promesse dans le dernier numéro et je tiens à la tenir. Sans plus attendre, lançons-nous dans le vif du sujet.

Donc, je récapépète; d'abord sauvegarde, ensuite éditeur de secteurs, puis aller à l'adresse 3E4 pour modifier vitalité et remplacer valeur en hexa par 7F00 (Et faites de même pour les autres personnages tant que vous y êtes!). Ça va déjà mieux, non? Après cela, direction 3EE pour remonter vos finances en mettant

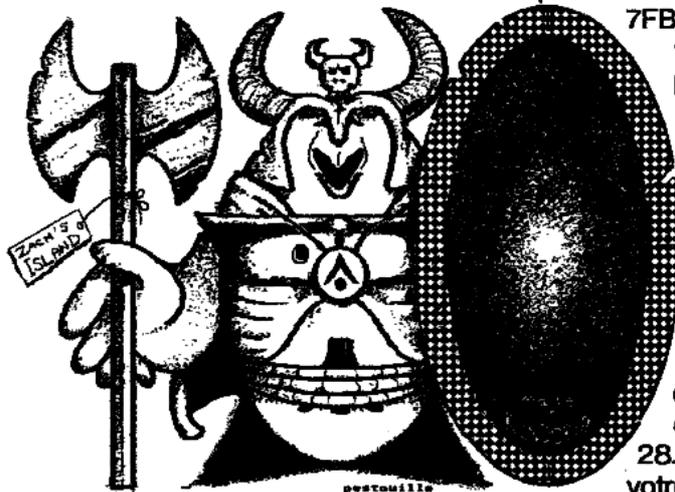
7FBB par exemple (Evitez la valeur 7FFF sinon la prochaine fois que vous gagnerez des sous le jeu se mettra à planter comme dans Ishar.). Aux adresses 510, 515, 51A, 51F et 524 vous modifierez respectivement votre force, votre constitution, votre sagesse, votre intelligence, votre agilité et enfin votre. Et pour tout ça vous mettez la valeur

28. Pour finir, à l'adresse 529, votre psychique vous modifierez

et en 52E votre physique sera rétabli par la valeur 64.

Voilà, j'espère que vous vous amusez bien. Grâce à ces patchs (Ou peut être à cause.) Ishar 2 n'a tenu qu'une semaine entre mes mains à raison de quatre heures par jour. Tiens en passant, vous l'avez sûrement remarqué, de nombreuses fautes de frappe étaient présentes dans le dernier numéro, notamment dans Flabbergasted où "un Yag" s'était transformé involontairement en "une Yag", mais vous avez dû corriger de vous-même. Voilà j'ai bavé mon rond de chapeau. Allez, à plus..

WORRAPS



BON DE COMANDE MINI F1

Version 103
truecolor

Je désire une version de MINI F1 (plus un adaptateur au choix), pour la modique somme de 230Fr, Deux cents trente francs*

- Un adaptateur port imprimante.
Ou Un adaptateur 4 joysticks STE.
Ou Un adaptateur pour joystick analogique.

ci-joint mon règlement par chèque à l'ordre de Association FAUCONTACT.

A retourner à l'adresse suivante :

FAUCONTACT, Mas du Crès, 30140 Boisset & Gaujac

*Frais de port compris.

COURRIER



CHERS LECTEURS...

Comment peut-on intégrer les fontes type1 provenant du PC ou du Mac dans Speedo GDOS ? (...) Le Falcon a la particularité d'accepter tous les types de disque dur IDE ou ASCII. Pourquoi les revendeurs Atari pratiquent-ils des prix prohibitifs ? On trouve les mêmes DD en magasin PC à 20 % de moins ! J'ai l'intension d'en acheter un mais je ne sais pas que choisir !! Où puis-je trouver un driver SpeedoGdos pour une HP Deskjet 550C ? Savez vous pourquoi et comment on peut remédier au refus de lecture des protections sur cartouche pour les logiciels PRO24 et MASTERSCORE ? N'existe pas sur Falcon un problème de gestion de la mémoire ? (...) j'ai eu l'impression soit qu'il ne me rendait pas correctement la main, soit qu'il était incapable de se retrouver dans les appels (deux bombes). J.Hilaire.

En effet il paraît que Speedo Gdos est capable de gérer les fontes Adobe Type 1. Malheureusement jusqu'à présent tous nos essais se sont avérés infructueux. Nous avons essayé des fontes venant du Mac...et nous allons essayer avec des fontes en provenance du PC...nous poursuivons donc nos investigations ! Si certains ont trouvé quelque chose, qu'ils nous le fassent savoir. Ensuite concernant le disque dur...mettons les choses au point: le Falcon accepte les disques durs IDE (internes), et les disques durs SCSI 2. Ces deux types sont des standards, avec des normes strictement définies. Donc que ce soient des disques dur PC, Mac ou ce que vous voulez, du moment qu'ils appartiennent aux catégories sus-citées, il fonctionneront sans aucun problème sur votre Falcon 030 moyennant tout de même un petit reformatage. Concernant les disques dur SCSI ça doit aussi fonctionner sans problème mais il est probable qu'il faille un câble particulier pour les brancher sur la prise SCSI 2. Enfin

concernant les disques dur ASCII (l'ancien format Atari quoi !), là par contre il faut une interface qui convertisse les signaux ASCII en signaux SCSI, ce qui existe (il me semble que l'asso Contact'St en vend une mais je ne sais plus trop). Pour ce qui est des prix, la seule explication que nous voyons est que les magasins PC vendent beaucoup plus de disques dur que les revendeurs Atari. Ils peuvent donc se permettre des marges plus serrées. Pour le driver pour HP 550C...heu...on sait pas ! Speedo Possède un driver pour HP 500, en principe ça devrait déjà te permettre d'imprimer tes documents en noir et blanc. Par contre pour la couleur, il te faut effectivement un driver spécifique... on va chercher. Pour pro24 et Masterscore, on est désolé mais on ne voit pas trop la solution... si vous êtes plusieurs à connaître ce problème, dites le nous on verra ce que l'on peut faire... Bon et pour terminer, concernant la mémoire, c'est vrai que parfois certains programmes rendent mal la main. Cela ne vient pas forcément d'un bug de la ROM du Falcon, mais d'une mauvaise programmation...ceci dit les ROM du Falcon sont LOIN d'être parfaites, et cela ne nous étonnerait pas outre mesure qu'elles soient parfois fautives...

(...) sur mon Falcon (TOS 401) lorsque j'utilise un logiciel de direct to disk à une fréquence d'échantillonnage inférieure ou égale à 16 Khz, j'ai un grésillement important sur une des deux voies (toujours la même). D.Bregardis

Sur les premiers Falcons sortis, certains avaient quelques problèmes. Des grésillements apparaissaient lors de la restitution du son, et cela à certaines fréquences et d'un seul côté. Nous n'avons pas vu votre Falcon, nous ne pouvons donc pas nous permettre de donner une explication

formelle... Dans un premier temps essayez d'utiliser le logiciel sur d'autres Falcons que le votre et dans les mêmes conditions. Si les grésillements persistent, alors il est probable que cela vienne d'un bug du programme ou peut être d'un accessoire ou encore d'un programme dans le dossier AUTO (auquel cas réessayez mais sans aucun programme en AUTO ou en accessoire). Si par contre les grésillements disparaissent, là vous pouvez commencer à regarder votre Falcon d'un sale oeil...Vous ne pouvez pas conserver un Falcon avec un tel problème. Avez vous toujours utilisé le câble avec les résistances à 200 K Homs lors de la digitalisation ? Car sinon, le convertisseur CODEC a peut être pris mal... Le mieux à faire est de contacter votre revendeur (ou sinon directement Atari) en précisant clairement le problème. Si jamais vous renvoyez votre Falcon (ce qui semble être la meilleure chose à faire si les essais sur un autre Falcon se sont avérés infructueux), n'oubliez pas d'inclure les logiciels sur lesquels les problèmes se produisent. Si vous faites vous même l'emballage n'oubliez jamais de le faire bien costaud, plutôt discret (pas de logos ou autres) et de passer de préférence par un transporteur privé. Enfin, vérifiez si votre garantie est encore valable (sinon demandez un devis de réparation). Bon mais cependant les autres adhérents peuvent noter que ce n'est pas parce que leur Falcon fait des grésillements qu'il est bousillé...par exemple la Warum demo nous a fait des sacrées frayeurs...alors que c'est sa routine de player qui bugge...

Voilà, par manque de place c'est malheureusement tout. Ceux à qui on n'a pas répondu seront contactés personnellement ! Si jamais vous avez des remarques à faire, ou des réponses à apporter, écrivez nous, on fera partager aux autres !!



DOMPUBS

Hello !

J'ai une petite remarque à vous faire, certains programmes de cette rubrique ont été créés par des adhérents de Faucontact, or ils ne m'ont pas été envoyés par leur programmeurs, mais par d'autres personnes... étonnant non ?

Donc je vous le dis, n'hésitez plus, envoyez nous vos créations, vous serez rémunéré sous forme de DP, et en plus vous en ferez profiter tous les autres adhérents.

merci pour eux et pour le FALCON.

PSET

SON:

Réf.: FCT_S1

5TO5 : Convertisseur de samples, fonctionne sous Multitos et utilise le format DVSM (Fortune et Winrec). Mais aussi une floppée d'autres formats en provenance d'univers différents.

PAULA 22A : Player de modules, tourne en ACC idéal pour travailler en musique (nécessite FPATCH2.PRG).

FALCPLAY : Encore un player, mais pas en ACC.
MODULES : Une série de .MOD.

Réf.: FCT_S2

FALCONTRACKER : Un autre player de modules en ACC, ne fonctionne pas sous Multitos.

DAME : Logiciel de D2D très puissant, à utiliser si vous êtes un passionné. (Nécessite une partition entière de votre DD car il écrase votre partition à cause de sa façon d'échantillonner)

Réf.: FCT_S3

WINREC : Autre logiciel de D2D, c'est celui que je préfère. Il est fiable et propose pas mal d'options.

STARPLAYER : Player de modules en .prg, qui a l'avantage d'être très beau.

Réf.: FCT_S4

BACKTRACK : Un autre player de modules, en accessoire.

DESKTRACKER : Un player de modules, en accessoire avec oscilloscope et ou analyseur de spectre.

Réf.: FCT_S5

DIGITAL TRACKER : Un soundtracker 8 voix en pleine évolution, c'est du tout bon.

ZIKZIK : Un sampleur 8 et 16 Bits, bourré d'options, couplé à Digital tracker c'est le rêve.

DIVERS:

Réf.: FCT_D1

PICSWITCH : Très bon soft qui permet de convertir les images couleur en monochrome selon divers algorithmes afin de les imprimer ou de les sauver. Il gère les formats neo, degas, tny, art director, spc, spu, prism paint, img, aim, gif, ria, pcut2 16 & 256), iff, mac paint, startup screen, atari portfolio, plus divers formats 8 bits. Sauve au format IMG uniquement. Gère les imprimantes Epson 9/24 et HP Laserjet.

HAMB : Viewer d'images JFF et HAMB de l'Amiga.
SPEED OF LIGHT : Visualiseur d'images GIF très rapide.

TGA SHOW : visualiseur d'images .TGA très rapide aussi.

VIEW JPEG : J'ai tous dit dans le titre... non ?

WATCH IT : Visualiseur .GIF distribué avec les lers FALCON.

Réf.: FCT_D2 (pack anti bug FALCON, on vous l'offre.)

CHNGMON : Corrige un défaut du falcon, qui est de ne pas faire de reset lors du changement de moniteur.

CHNGNAME : Permet de renommer le volume d'une unité de disquette ou disque dur.

SHELFIX : Corrige la copie en mémoire du fichier NEWDESKINF (sur certain TOS)

FCOUNTER : Compte le nombre de dossier sur le DD et permet de configurer le système avec l'aide de FOLDRnnn.prg.

FOLDRnnn.prg : A utiliser avec FCOUNTER.

FPATCH2 : Indispensable pour faire tourner certains soft et/ou démos.

HP_OUT : Coupe le HP interne.

Réf.: FCT_D3

AUTOSORT : Pour classer les .PRG dans le dossier auto.

BOOT.DMV : Permet de configurer le newdeskint depuis le dossier auto, plus quelques autres services.

DIVFILE : Permet de partager un gros fichier en plusieurs petits fichiers.

INIT : Tout seul dans votre dossier auto, il vous permet de choisir les PRG que vous désirez lancer au boot. (Dis pascal, à quand le 2 hein hein dit...)

XTERMINE : Anti virus pour disquette et DD très puissant.

Réf.: FCT_D4

STE2FALC : Permet de faire tourner des soft ST. Il se débrouille comme il peut, et il faut reconnaître que dans certains cas il est très efficace. Il y a deux versions: une pour le bureau GEM, l'autre plus spécifique aux jeux.

MAGI : Idem STE2FALC. Sa technique de programmation différente lui permet de lancer certains jeux que STE2FALC refuse de lancer (et vice versa)

BACKWARD : Idem MAGI mais moins concluant.

VCB 1,4 : Après de nombreux essais, il c'est avéré être le plus puissant.

Réf.: FCT_D5

MOUSE WIND : Sélectionne la fenêtre sous la souris.

GERMAN2ENGLISH : Traducteur allemand vers anglais. Bien pratique pour ceux qui captent rien à l'allemand !

GEMVIEW 2.43 : Dernière version de ce fabuleux utilitaire vous permettant de visualiser tous les formats de fichiers images existants (ou presque!).

Réf.: FCT_D6

FALCONSCREEN : Permet d'atteindre des résolutions plus élevées (le GEM en 1600*800 ? Mais oui c'est possible !! sur un moniteur RVB !!). Nécessite de la programmation, mais je peux vous le configurer si vous le désirez. (Nécessite Pure C)

SCREEN INIT : Utile pour repositionner l'écran, après le passage de FALCON SCREEN.

Réf.: FCT_D7

DESKPIC : Affiche une image GIF en fond du bureau. GENIAL ! Plus 2 images, celles qui ornent le bureau de splash et le mien.

AEYES 101 : Affiche des yeux dans des fenêtres. (complètement inutile !)

DISCPAC : Editeur de secteur, génial.

Réf.: FCT_D8

SLCMAI : Un selecteur de fichiers alternatif.

RUFUS : Permet de transférer des données par divers ports.

LET THEM FLY : Raccourcis clavier, fly dial.

Réf.: FCT_D9

MASTER BROWSE 2.8 : Un éditeur de textes en anglais.

GODEL : Processeur mathématique (demo).
TTARTIST V1.3 : Protecteur d'écran (à l'origine pour TT mais il tourne sur F030). Nécessite un coprocesseur arithmétique.

7UP : Un super éditeur de textes, nouvelle version tout en français. Mais... mais je rêve... je suis en train de l'utiliser.

Réf.: FCT_D10

CHEETAH : Copieur de fichier hyper rapide (quelques risques subsistent en copie de partition à partition).

STZIP23 : Dernière version de ce compacteur.
DESKCOPY et DESKFORMAT : Copier et formater des disquettes en tâche de fond sous Multitos.

Réf.: FCT_D11

GEMBENCH : Pour avoir une idée de la vitesse de votre machine.

QINDEX : Idem GEMBENCH. Résultats parfois fantaisistes.

SYSINFO : Vous donne plein de renseignements sur votre machine. (cookies, adresses diverses...)

WINLUPE : Un accessoire génial qui ouvre une fenêtre, avec zoom permanent sur la souris.

COPY 030 : Un copieur de disquettes, spécial FALCON.

Réf.: FCT_D12

FRACTAL : Générateur de fractales, qui utilise le DSP ET le 68030, ou le copro pour un calcul en nombres réels.

SNAPFALC : Snapshot qui fonctionne dans toutes les résolutions.

GTC74G : Un visualiseur d'images de différents formats, avec colorimétrie et autres effets.

BNR VIEWER : Visualiseur d'images en true color, nombreux effets.

Réf.: FCT_D13

POV 2 : Ce génial Raytracer, enfin dispo, en version 2.0, plus de nombreux utilitaires.

Réf.: FCT_D14

NCONVERT : Un convertisseur d'images pour 34 formats différents.

NVIEW : Visualiseur d'images pour une bonne trentaine de formats lui aussi.

JEUX

JEUX

Réf.: FCT_J1

OXYD_falcon : Fabuleux jeux de réflexion en 640*480 en 256 couleurs (uniquement en VGA).

Réf.: FCT_J2

METHACK : Jeux d'aventures sous GEM (style jeu de rôle, un peu comme ULTIMA quoi !). C'est génial j'y retourne !

Réf.: FCT_J3

MINE V2.2!! : Un démineur classique, sous GEM.
SHANGHAI : Sous GEM, programmé par l'auteur de 7UP.

MEMOS : spécial FALCON 256 couleurs, c'est mon coup de cœur. Un jeu de mémoire : associez les images entre elles ! incroyablement prenant !

Réf: FCT_J4
VERTICAL MAYEN : Superbe COLUMNS en 256 couleurs, musiques superbes voilà j'y retourne!
TRON : Un tron se jouant à deux joueurs, soundtrack et true color.

Réf: FCT_J5
DES LASER ET DES HOMMES : Une pré-version d'un jeu à la WOLFENSTEIN 3D. Réf: FCT_J6

MINI F1 v1.03 : Demo de la nouvelle version de ce jeu, en true color, avec plein d'effets et une jouabilité améliorée.

WATER.SHIP : Mini jeu, où vous pilotez un Zodiac à la recherche d'une arme, hard scroll et soundtrack.

DEMOS/SOFTS:

Réf: FCT_DSFT1
AVANT VECTOR DEMO: démonstration d'un logiciel de vectorisation.
ZOOM DEMO: Démonstration d'un magnifique éditeur d'icônes couleur.

Réf: FCT_DSFT2
DIGITAPE : logiciel de direct to disk
SOLUTION : logiciel de math.

Réf: FCT_DSFT3
DEMO du jeu "Space Junk". Très beau A voir.
NOUVEAU : une version avec animation video est dispo, mais sur 3 disquettes donc, **ATTENTION** le prix est de 45 Francs.

Réf: FCT_DSFT4
DEMO du jeu "HUMANS". On rit bien (enfin...), c'est en 256 couleurs et ça fait plaisir à voir ! **ATTENTION** Cette démonstration tient sur deux disquettes. Donc le prix est de 30 francs. Pensez-y lors de la commande !

Réf: FCT_DSFT5
APPLIC.SYS : Une grande partie des softs édités par Application Systems. (merci à eux)
+ BLOW UP et OUTSIDE.

Réf: FCT_DSFT6
D2M : La demo du soft de dessin, édité par la société PARX.
PIXART : Un soft de dessin en allemand.
LLAMAZAP : Démo jouable du jeu de JEFF
MINTER, nécessite un paddle Atari. (GENIAL avec un paddle NOIR... en attendant les GRIS!)

Réf: FCT_DSFT7
GFA_4 : éditeur sous GEM du Gfa 4.00.
KOBOLD.2 : Démo du génial soft de copie de fichiers.

SCRIPT_3 : Démo du traitement de textes, d'Applications Systems.

Réf: FCT_DSFT8
CHAGALL : Un soft de dessin en Allemand, son interface est fabuleuse.
CHARLY : Un autre soft de dessin en Allemand.
PAPILLON : Et hop encore un soft de dessin, et je vous le donne en mille... il est lui aussi en Allemand.

Réf: FCT_DSFT9
C.L.O.E : une démo de ce fabuleux raytracéur, avec la doc et des scripts.

DEMOS/FUN:

Réf: FCT_DFUN1
PLASMA : rotation et déformation plein écran en 3D et 50 images secondes. C'est en fait une version dérivée de la TERMINAL FUCK UP en plus performante.
BIG KART : un clone SUPER MARIO KART jouable et de 39 Ko !.

Réf: FCT_DFUN2
LEFT DONUT DEMO: une superbe musique (16 bits avec interpolation) et quelques objets 3D avec calcul d'éclairage en True Color. A voir.
SPHERE : une sphere composée de point, le tout calculé par le DSP.
NEW TREND : une démo standard, l'avis est partagé

au sein la rédaction.

Réf: FCT_DFUN3
CEBIT93 : démo réalisée à l'occasion du CEBIT 93. Son 16 Bits au rendez-vous et Déformation en True color aussi !
BOINKFAL: Une boule qui rebondit avec une image dedans ...

Réf: FCT_DFUN4
BRAINSTORM FLJ PLAYER DEMO: ben j'ai tout dit dans le titre !
3DIT: c'est de la 3D pour TT (non ?) qui tourne sur Falcon et en plus c'est de Brainstorm, c'est du tout bon !
INTEL: Démo avec le sens de l'humour: Fractales et logo "Motorola inside" qui tourne.

Réf: FCT_DFUN5
1600*800: on vous avait bien dit qu'on pouvait le faire (sur un écran RVB)
MODE7: De la 3D mappée aussi bien que la Super Nénesse de Nitengros, avec en plus un soundtrack en 16 bits ! Complètement tuant !

TERMIN: La TERMINAL FUCK UP livrée avec certains Falcons: celle qui nous a fait tous pleurer lorsqu'on l'a vue pour la première fois !!

Réf: FCT_DFUN6
SMOOTH VECTORS: De la 3D avec éclairages en True Color, c'est très beau.
OMEGAF30: Une démo pour STE 'adaptée' rapido sur Falcon: Un video clip avec du soundtrack (en VRAIE mono...)

GOURAUD: encore de la 3D avec éclairages, sauf que là il y a des trames très bien faites pour éviter qu'on voit les nuances de couleurs ! C'est réussi !

Réf: FCT_DFUN7
MCROVO: Trois démonstrations en 3D prévues à l'origine pour TT, avec des soundtracks géniaux en fond.
FRACTALRT: un zoom sur une fractale de Mandelbrot calculé en temps réel par le DSP !
DSPFRAC: Encore du zoom sur une fractale sauf que là vous indiquez où avec la souris !

Réf: FCT_DFUN8
SPECIAL VGA : Nombreux programmes faisant des Shades bobs et autres effets graphiques, pour moniteur VGA uniquement.

Réf: FCT_DFUN9
CRIME : Par DNT CREW, samples en boucle (ça prend la tête), shades bobs, logo 3D, fractales temps réel, cercle élastique (Hein splash que c'est génial les cercles... JND splash: OH-HH Oui !!), line dots...
COOL : Shades bobs plus une superbe musique, je plane...
COOOL : Des jolies spheres avec 8000 Pts grâce au DSP, musique c'est cool quoi...

Réf: FCT_DFUN10
WAITING THE ZIK DISK : par POSITIVITY, une dentro pleine d'humour arrosée d'un soundtrack sauce guignol (à la rédac, on l'adore)
YEAHHHHH2 : mais où est la un ??? Hard scroll en true color plus sprite et soundtrack.

Réf: FCT_DFUN11
PENTAGON : Musique très très techno, zoom, 3D et autre effets sympa.
TRUEDEMO : Un tableau de bord, et plein de vaisseaux spatiaux en 3D ombrée, le tout en TRUE COLOR.

Réf: FCT_DFUN12
680XX : image superbe agrémentée d'une musique sympa, plus de nombreux effets en line dots. (déplacez votre souris par hasard...)
DRAGON : Un scroll d'étoiles !!!!!
MODE7 : Nouvelles versions encore plus rapides.

Réf: FCT_DFUN13
CSC2 : Logo 3D de la Crystal Summer Convention 2.
BURP FALCON : Teste votre résistance au mel de

mer...

1800 PLOTS : Représentation spéciale de courbes mathématiques, modifiables par l'utilisateur.

MAD : Sphere en 3D mappée.
FF.DEMO : Paysages fractal, accompagnés de "What is love" (HADDAWAY).

Réf: FCT_DFUN14
BOUND 2 : Un ensemble de spheres animées, en raytracing temps réel au DSP
ROT 3D BMAP : Une rotation en 3D d'une image, du plus bel effet.
GOURAUD 2 : Objet 3D avec rendu de GOURAUD.
PLASTIC DREAMS : Musique techno, Start tunnel, rotation 3D. Sympa mai.
VARUM : Pourquoi... SPLASH est il encore accroché au plafond ????? Parceque c'est la meilleure du moment !

IMAGES:

Réf: FCT_IMG1
La vie des animaux vous intéresse ! Alors n'hésitez pas, ces images au format gif sont pour vous: BEARS.GIF; BUTTRFISH.GIF; CHOUETTE.GIF; COUGAR.GIF; CYOTE.GIF; FISHY.GIF

Réf: FCT_IMG2
De superbes photos venues de l'espace pour vous faire décoller: APOLLO.GIF; C2056059.GIF; EARTH.GIF; HORSE.GIF; JUPITER.GIF; JUPITER1.GIF; JUPITER2.GIF; M33.GIF; MARSFACE.GIF; MERCURY.GIF; MOON.GIF; NEPTUNE.GIF; NORTHAM.GIF; SATURNE.GIF; SUNFLARE.GIF; WALK.GIF

Réf: FCT_IMG3
(Pas de taboos chez FAUCONCONTACT/INDLR: Bon, a poil le président, allez hop !!) De beaux dessins accompagnés de jeunes femmes très légèrement vêtues et dans des positions un peu suggestives: AMBER.GIF; CITY.GIF; FRACTO08.GIF; FRACTO10.GIF; H.MISS05.GIF; JUSTINE.GIF; LUNE.GIF; OVNI.GIF; P.EMILY.GIF; SKYLINE.GIF; SUNSET.H.GIF; TORP208.GIF; VENUS3.GIF

Réf: FCT_IMG4
Ces paysages sont splendides ! Ce petit coin de nature est fait pour vous ! AUTUMN1.GIF; BIGLAKE.GIF; FLOWER.GIF; PAINT.GIF; REDROCK.GIF; YROSE.GIF

Réf: FCT_IMG5
Décidément le raytracing est une bien belle invention... BACKGRND.GIF; BILBOO5.GIF; LPOPS2.GIF; PICTURE0.GIF; PICTURE1.GIF; PICTURE2.GIF; PICTURE3.GIF; PLAZA.GIF; TILES1.GIF; VTEST1.GIF; WALKMAN.GIF; X.ARCHES.GIF

Réf: FCT_IMG6
Vive le raytracing... X.IONICS.GIF; X.LASER0.GIF; X.LOAK20.GIF; X.PALACE.GIF; X.PLANET.GIF; X.SKYVAS.GIF; X.TAKEOFF.GIF; X.WATERB.GIF

Bulletin d'inscription.

Voici le bulletin à renvoyer si vous désirez adhérer à l'association FAUCONtact. Remplissez le en lettres capitales afin qu'on n'écarte pas votre nom !!

NOM: _____

Prénom: _____

Adresse: _____

N téléphone (facultatif) : _____

Si vous possédez un B.A.L, indiquez-la: _____

Sur quel(s) serveur(s): _____

Quelles sont vos activités informatiques principales?

- GRAPHISMES.
 PROGRAMMATION.
 MUSIQUE. M.I.D.I
 JEUX.
 P.A.O./BUREAUTIQUE.
 AUTRES: _____

Quelles machines et quelle configuration possédez-vous (mémoire, périphériques, etc...)?

OUI, j'adhère à l'association FAUCONtact pour une durée de ...

- 6 mois. Ci-joint un chèque de 45 Francs (50 Francs pour les autres pays de la C.E.E.) à l'ordre de: Association FAUCONtact
 1 an. Ci-joint un chèque de 90 Francs (100 Francs pour les autres pays de la C.E.E.) à l'ordre de: Association FAUCONtact

Indiquez s'il s'agit d'un réabonnement.

Cette cotisation me permettra de recevoir le fanzine bimestriel de l'association. De plus, mes questions seront prioritaires et j'aurai accès à la banque de Domaines Publics et de démos de l'association.

En m'incrivant, j'autorise l'association FAUCONtact, à m'envoyer tous documents comportant mon nom (carte de membre, facture, etc...)

Si vous avez des suggestions concernant l'association, n'hésitez pas:

Renvoyez ce bulletin ainsi que le chèque à l'adresse suivante:

Association FAUCONtact,

Mas du Crès,

30140 Boisset et Gaujac. FRANCE.

BON DE COMMANDE DES DP

NOM :

Prénom :

Pseudo :

Adresse :

NOM et REFERENCE	PRIX	QUANTITÉ	PRIX

TOTAL :

Chaque disquette est facturée 15 francs port et disquette compris. Renvoyez ce bulletin ainsi qu'un chèque à l'ordre de Association FAUCONtact à l'adresse suivante:

Association FAUCONtact, (Rubrique DP), Mas du Crès, 30140 Boisset et Gaujac.