

Ressources

COLLABORATORS

	<i>TITLE :</i> Ressources		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 16, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Ressources	1
1.1	Les Ressources de l'Amiga	1
1.2	Emulation d'autres systèmes	1
1.3	Shared Libraries	2
1.4	Device Drivers	2
1.5	Les Formats de Fichiers	3
1.6	Interchange File Format	3
1.7	8-bit Sampled sound Voice	3
1.8	Formatted TeXT	4
1.9	IFF Interleaved Bitmap	5
1.10	IFF Simple Musical Score	6
1.11	Joint Photographic Experts Group	7
1.12	Moving Pictures Experts Group	8
1.13	DCS	9
1.14	EPS	10
1.15	TIFF	10
1.16	Exec	10
1.17	Inter-Process Communication	10
1.18	Intuition	10

Chapter 1

Ressources

1.1 Les Ressources de l'Amiga

Emulation d'Autres Systèmes

Les Formats de Fichiers

Shared Libraries

Device Drivers

Exec

Inter-Process Communication

Intuition

Retour Au Menu Principal

1.2 Emulation d'autres systèmes

En plus de son propre système d'exploitation, l'Amiga peut aussi fonctionner avec :

UNIX

AT&T UNIX System V Release 4.

MINIX.

LINUX.

Apple Macintosh (émulateurs hardwares)

A-Max, Emplant.

MS-DOS (émulateurs hardwares et/ou logiciels).

Cartes Commodore

Vortex Atonce, Golden Gate

KCS Power PC Board

GVP AT-500 16 Mhz

PC-Task

Commodore 64 (émulateur logiciel).

Sinclair Spectrum (émulateur logiciel).

Sinclair QL (émulateur logiciel).

Atari ST (émulateur logiciel et/ou hardware).

Chamäleon, Medusa, ST Emulator

Certains de ces systèmes peuvent fonctionner parallèlement au système Amiga.

1.3 Shared Libraries

L'Amiga possède des libraries (bibliothèques) qui peuvent être ouvertes et utilisées par des programmes d'application. Cela diminue la taille des programmes exécutables et du Kernel.

Une library est un ensemble de routines de programmation réunies en un fichier. Les programmeurs peuvent utiliser ces libraries et les routines qu'elles contiennent pour éviter, en fait, de faire un travail déjà réalisé.

Par exemple, la programmation d'un requester peut être évitée en utilisant la routine adaptée présente dans l'Asl.library par exemple.

1.4 Device Drivers

L'objectif des Devices est de contrôler des éléments périphériques en évitant aux programmeurs la nécessité d'inclure ces fonctions de contrôle dans leurs productions.

Ils offrent surtout un standard en matière de communication avec les périphériques.

Tous les devices sont accessibles de manière logique par l'utilisateur ou le programmeur. Par exemple, envoyer une sortie vers un fichier s'effectue de la même façon que pour l'envoyer vers une fenêtre, le port parallèle, le synthétiseur vocal, ou tout autre device. Les E/S synchrones et asynchrones sont supportées par tous les devices. De nouveaux drivers peuvent facilement être ajoutés au système sans nécessiter la recompilation du Kernel.

1.5 Les Formats de Fichiers

Interchange File Format

JPEG

MPEG

EPS

DCS

TIFF

1.6 Interchange File Format

Créé en 1985 par Electronic Arts, le format IFF offre aux applications un format de fichier standard pour les graphismes, les animations, les sons numérisés, les données MIDI ou d'autres types de données. Une bibliothèque (iffparse.library) offre les fonctions pour manipuler les données.

Un fichier correspondant à ce standard commencera par l'entête FORM suivie un peu plus loin par le format du fichier :

8SVX	8-bit Sampled sound Voice
ACBM	Amiga Continuous BitMap
ANBM	ANimated BitMap,
ANIM	ANIMation
FNTR	Raster FoNT
FNTV	Vector FoNT
FTXT	Formatted TeXT
GSCR	General-use musical SCoRe
ILBM	InterLeaved BitMap
PDEF	Deluxe Print page DEFinition
PICS	Macintosh PICTureS
SMUS	Simple MUsic Score
VDEO	Deluxe Video Construction Set ViDEO

1.7 8-bit Sampled sound Voice

Date de création : 7 février 1985

Auteurs : Steve Hayes et Jerry Morrison de Electronic Arts

Avec l'aide de : Bob "Kodiak" Burns, Commodore-Amiga
R. J. Mical, Commodore-Amiga
Greg Riker, Electronic Arts
Steve Shaw, Electronic Arts
Barry Walsh, Commodore-Amiga

Référence : Le standard "EA IFF 85"

Le 8SVX est un ensemble de données contenant un son échantillonné numériquement sous un format 8 bits.

Le format 8SVX est prévu pour être restitué sur du matériel utilisant des échantillons 8 bits et autorisant un contrôle du volume pour obtenir un bon rapport signal/bruit. Ainsi un FORM 8SVX stocke des échantillons 8 bits et un niveau de volume. Pour des résolutions supérieures (12 ou 16 bits), un format différent sera nécessaire.

Le 8SVX peut stocker une enveloppe, un contour d'amplitude pour enrichir des notes musicales.

Le 8SVX stocke toutes les données de la courbe sonore dans un chunk appelé "BODY". Il stocke les paramètres nécessaires à son exécution dans le chunk de tête de fichier "VHDR". "VHDR" et tous les chunks optionnels comme "NAME" (nom du fichier), "(c)" (copyright), "AUTH" (nom de l'auteur) et "ANNO" (remarques) doivent apparaître avant "BODY".

Il y a deux façons d'utiliser FORM 8SVX :

Comme un son numérisé directement ou comme un instrument de musique numérisé qui joue des notes.

Un son numérisé directement est une série d'échantillons de données sonores accompagnées de leur vitesse de restitution et de leur amplitude. Le programme utilisant ce fichier peut ajuster ou moduler ces variables.

Pour les instruments de musiques, l'idée est de stocker une onde sonore échantillonnée qui sera paramétrée par le ton, la durée et l'amplitude pour jouer chaque note. L'enveloppe peut aussi être stocker pour moduler la courbe sonore.

Le format 8SVX peut supporter un algorithme de compression. le plus courant est le Fibonacci-delta écrit par Steve Hayes. Cet algorithme code chaque delta en 4 bits. La donnée compressée fait la moitié de la taille du delta original plus 2 octets de valeur initiale. Cependant, l'utilisation de la compression engendre souvent une perte de qualité de l'échantillon. Son utilisation doit donc être évitée en ce qui concerne les instruments de musique.

1.8 Formated TeXT

Date de création : 15 novembre 1985

Auteurs : Steve Shaw et Jerry Morrison de Electronic Arts
Bob "Kodiak" Burns de Commodore-Amiga

Un fichier FTXT est un principe d'archivage et d'échange prévu pour trois usages. Le plus simple est pour une représentation minimale d'un texte avec des caractères de positionnement espace "SP" et des fins de lignes "LF". Il ne tient pas compte des caractères de déplacement du curseur. La deuxième utilisation est prévue pour les textes qui contiennent des informations de formatage explicites comme les fontes (famille, taille, style...). La

troisième contient la base de la structure du document en plus des informations de formatage.

Ces trois utilisations sont bien sûr facilement transférables et lisibles par n'importe quel programme reconnaissant ce standard.

Le texte est stocké dans un ou plusieurs chunks "CHRS" dans un FTEXT. Chaque CHRS contient une chaîne de texte 8 bits compatible avec les standards ISO et ANSI. FTEXT utilise le jeu de caractères central des standards ISO/ANSI.

L'apparence des caractères est stockée en séquences de contrôle dans les chunks CHRS.

Le contenu principal du FORM FTEXT se trouve dans sa chaîne de caractère "CHRS". Des chunks de formatage de texte peuvent aussi apparaître. La seule propriété de formatage définie pour le moment est "FONS", qui précise la fonte. Un FORM FTEXT sans CHRS représente une chaîne de caractère vide.

1.9 IFF Interleaved Bitmap

Date de création : 17 janvier 1986

Auteur : Jerry Morrison de Electronic Arts

Un fichier ILBM est un principe de stockage prévu pour trois usages. Tout d'abord, une simple image qui précise exactement comment elle doit être affichée (résolution, taille, palette...). Ensuite, une image prévue pour être incluse dans une image plus grande qui possède ses propres résolution, palette, etc... Et enfin, une image vide avec une palette. ILBM est aussi une base pour des fichiers d'animations. L'une des utilisations d'ILBM sera de préserver le plus d'informations possible d'un environnement à l'autre. Une autre utilisation sera de stocker des données utilisables par plusieurs programmes.

La définition "BMHD" ainsi que toutes les définitions optionnelles doivent apparaître avant le chunk "BODY".

BMHD

l'indispensable définition "BMHD" contient le BitMapHeaDer qui décrit les dimensions et l'encodage de l'image, ainsi que les données nécessaires à la compréhension du chunk BODY qui suit.

CMAP

La définition "CMAP" stocke la palette de couleur sous la forme d'un trinôme décrivant l'intensité de rouge, vert et bleu

L'intensité des composantes de couleurs sont définies par un nombre allant de 0 à 255. le blanc correspond à (255, 255, 255) et le noir est (0, 0, 0).

GRAB

La définition optionnelle "GRAB" précise un point relativement à son angle supérieur gauche. Cela est utile pour les pointeurs de souris ou les brosses (paint brush).

DEST

Cette définition optionnelle est une façon de dire comment répartir zéro ou plusieurs bitplans dans une image de destination plus profonde (en terme de nombre de bitplans).

SPRT

La présence du chunk "SPRT" indique que cette image doit être considérée comme un sprite. C'est au programme qui lit le fichier d'en faire un sprite ou de le traiter en tant qu'image.

CAMG

Le chunk "CAMG" est spécifique à l'Amiga. Il stocke le "viewport mode". Ce qui permet de spécifier un mode d'affichage comme "dual playfield" et "hold and modify".

BODY

L'image source est stockée dans le chunk "BODY". Ce chunk contient tous les bitplans et les masques optionnels.

Le BitMapHeader, dans le chunk BMHD, précise les dimensions de l'image (w, largeur et h, hauteur) ainsi que les plans (nplanes). Il contient aussi les champs de masque qui indiquent si il y a un plan de masque et le champs de compression qui indique l'algorithme de compression utilisé. Cette information est nécessaire pour interpréter le chunk BODY, donc, le chunk BMHD doit apparaître en premier.

CRNG

Un chunk "CRNG" contient les informations sur le "color register range". Il est utilisé par le programme Deluxe Paint pour identifier les registres de couleurs adjacents pour les dégradés et les cyclages de couleurs. Il peut y avoir 0 ou plusieurs chunks CRNG dans un fichier ILBM, mais tous doivent apparaître avant le chunk BODY.

1.10 IFF Simple Musical Score

Date: 5 février 1986

Auteur : Jerry Morrison de Electronic Arts

SMUS est un format de données destiné à être utilisé pour échanger des partitions musicales entre programmes et stocker des musiques pour des jeux. FORM SMUS utilise le time encoding de la notation musicale commune (demies notes, notes pointées, silences...).

Une partition SMUS peut dire quels instruments sont supposés jouer quelles notes. Mais la partition est indépendante du système et du logiciel utilisé pour jouer les notes.

Comme tous les formats IFF, SMUS est un format d'archivage. Il est complètement indépendant de la représentation de la partition en mémoire, des opérations d'édition, de l'interface utilisée, des graphismes affichés, du matériel de traitement et de sonorisation. Comme tous les formats IFF, SMUS est extensible.

La partition est contenue dans des pistes (tracks). Il peut y avoir jusqu'à 255 pistes dans un FORM SMUS. Chaque piste est stockée dans un chunk TRAK. Le nombre de chunks TRAK est enregistré dans le chunk SHDR au début du FORM SMUS. Les chunks TRAK apparaissent dans leur ordre numérique 1, 2, 3...

Les différentes pistes d'une partition peuvent avoir des tailles différentes.

Un FORM SMUS contient un "SHDR" nécessaire, suivi par n'importe quel nombre de chunks de données "track". Les chunks optionnels comme "NAME", copyright "(c)", et les références d'instruments "INS1" peuvent aussi apparaître.

Le "SHDR" contient un SScoreHeader qui spécifie les informations globales sur la partition. Il doit apparaître avant les TRAKs dans un FORM SMUS.

Le chunk optionnel "NAME" précise le nom de la partition.

Le chunk optionnel "(c)J" contient le copyright de la partition.

Le chunk optionnel "AUTH" précise le nom de l'auteur de la partition.

Le chunk optionnel "ANNO" contient n'importe quelle annotation voulue par l'auteur.

Le chunk "INS1" identifie les instruments à utiliser pour cette partition.

Les chunks de données TRAK contiennent le noyau de la partition. Le contenu d'un chunk TRAK est une suite d'évènements codés sur 16 bits comme "note", "rest" ou "set instrument". Ces évènements sont des ordres stockés par ordre de leur apparition. Les pistes peuvent être polyphoniques.

Chaque évènement est stocké comme un "SEvent" ("simple musical event"). Chacun des SEvent possède un champ de 8 bits appelé "sID" et un champ de données sur 8 bits.

1.11 Joint Photographic Experts Group

Le JPEG est un procédé standardisé de compression d'image. JPEG signifie Joint Photographic Experts Group, le nom original du groupe qui a écrit ce standard.

Cet algorithme se base entre autres sur l'échantillonnage de la chrominance, des transformations cosinus discrètes, et une routine de gestion de l'entropie Huffman (la plus courante) ou arithmétique (licence détenue par AT&T, IBM et Mitsubishi).

Le JPEG est prévu pour compresser aussi bien des images numériques en couleurs ou en nuances de gris représentant des scènes "naturelles". Le JPEG ne gère pas le noir et blanc (1 bit par pixel) ni la compression d'animations.

Le JPEG est "dégradant", signifiant par là que l'image résultant de la compression n'est pas exactement identique à l'originale. L'algorithme base l'essentiel de sa compression sur l'exploitation des limites connues de l'oeil humain, principalement le fait que les petits détails de couleurs ne sont pas perçus aussi bien que les petites nuances de lumière. Ainsi, le JPEG est prévu pour compresser des images qui seront observées par des humains. Si ces images sont analysées par une machine, les petites erreurs introduites par le JPEG pourront poser des problèmes, même si elles sont invisibles à l'oeil nu.

Une possibilité très utile du JPEG est que le degré de perte peut être modifié en ajustant les paramètres de compression. Cela signifie que l'on peut établir un compromis entre la taille du fichier et la qualité de l'image. Il est ainsi possible d'obtenir de très petits fichiers si la qualité de l'image importe peu.

Cette opportunité de pouvoir compresser une image couleurs de 2 Mo en un fichier de 100 Ko représente une différence énorme en terme de transmission ou d'archivage.

Le JPEG remplace d'ailleurs le format GIF sur les réseaux. Tout d'abord grâce à cette capacité de compression exceptionnelle (5 fois supérieure au GIF), mais aussi de par sa gestion d'images en 24 bits par pixel, soit 16 millions de couleurs.

Cependant, sur des images ne comportant que quelques couleurs différentes, le GIF fait mieux que le JPEG, en particulier si de vastes surfaces de pixels exactement de la même couleur sont présentes. Le JPEG a aussi du mal avec les contrastes trop prononcés comme une zone de noir adjacente à une zone blanche, le résultat étant souvent un adoucissement de ce contraste. Ce problème peut aussi apparaître avec du texte.

1.12 Moving Pictures Experts Group

Le MPEG (Moving Pictures Experts Group) est le nom du groupe d'experts qui a créé ce standard de numérisation vidéo et de compression audio.

MPEG commence avec une résolution relativement basse d'environ 352 par 240 points (pour le standard US) et un son de qualité laser. Les images sont en couleurs, mais converties en YUV, et les deux canaux de chrominance (U et V) sont réduits chacun à 176 par 120 pixels.

Le principe de base est de prévoir le mouvement d'une image à l'autre sur une échelle temporelle, et d'utiliser des DCT (Transformations Cosinus Discrètes) pour organiser les redondances au niveau spatial. Les DCT sont effectuées sur des blocs de 8x8, et la prévision faite sur le canal de luminance (Y) sur des

blocs de 16x16. En d'autres mots, étant donné le bloc de 16x16 dans l'image courante que vous essayez de coder, vous recherchez une ressemblance proche de ce bloc dans une image précédente ou à venir (il existe des modes de prévisions rétroactives où les dernières images sont envoyées en premier pour permettre une interpolation entre les images). Les coefficients DCT (soit des données présentes, soit de la différence entre ce bloc et la ressemblance proche) sont "quantisés", ce qui signifie qu'on les divise par des valeurs pour supprimer quelques bits superflus. La quantization peut changer pour chaque macrobloc (c'est à dire les 16x16 de Y et les 8x8 correspondants dans chaque U et V). Le résultat de tout cela, qui inclus le coefficient DCT, les vecteurs de mouvement, et les paramètres de quantization, est codé en Huffman utilisant des tables fixes. Les coefficients DCT ont une table Huffman spéciale qui est bi-dimensionnelle.

Il y a en fait 3 types de trames codées :

I ou Intra frames : Il s'agit de trames codées comme une image fixe, n'utilisant aucune trame passée.

P ou Predicted frames : Elles proviennent de la prévision de la plus récente trames I ou P reconstruite.

B ou Bidirectional frames : Elles résultent des deux trames I ou P les plus proches, une passée et une à venir. Il s'agit en fait d'une interpolation.

Le codage du son joue sur un effet de masque, effaçant les parties du spectre sonore inaudibles ou "cachées" par d'autres sons.

En fait, que ce soit pour le son ou pour les images, le format MPEG joue sur les limites de la perception de nos sens. Tout comme le JPEG.

Mais de par la nature même de son principe, le MPEG occasionne une perte de qualité, certes infime, mais pourtant perceptible. Ainsi, le Video-CD (12 cm) contient 74 mn de film dans une qualité proche ou supérieure à celle d'une cassette vidéo. Cependant, le Compact Disk Video (30 cm et 1 heure par face), avec son enregistrement analogique de l'image, garde un net avantage au niveau de la qualité d'image et du son, surtout dans les scènes mouvementées, où le MPEG a bien du mal à prendre ses repères.

1.13 DCS

Il s'agit d'une variante de l'EPS contenant l'ensemble des paramètres de séparation quadrichromique. Un fichier dans ce format est constitué de 5 sous-fichiers EPS dont un est utilisé pour la visualisation de l'image et les quatre autres pour définir les données des couleurs.

1.14 EPS

L'Encapsuled PostScript est un format de fichier permettant d'inclure une image bitmap dans une description en langage PostScript. Un fichier EPS est très souvent utilisé pour décrire une illustration vectorielle comme les clip-arts.

1.15 TIFF

Le Tagged Image File Format définit une image bitmap en mode RVB ou CMJN (ce dernier servant en PAO pour la séparation des couleurs).

Ce format est couramment utilisé dans le monde Macintosh. La plupart des scanners fournissent aussi des fichiers dans ce format.

1.16 Exec

Au coeur du système d'exploitation se trouve Exec. Il s'occupe de la gestion des interruptions et gère donc le multitâche. Exec est beaucoup plus petit que le Kernel UNIX, car de nombreuses fonctions comme les drivers se trouvent dans une autre partie du système.

A l'inverse des systèmes "multitâches coopératifs" trouvés sur d'autres machines, Exec est un système multitâche préemptif à part entière.

1.17 Inter-Process Communication

Il y a deux formes d'IPC :

- Le mécanisme interne de transfert de message (message-passing)
- ARexx

Les processus de l'Amiga communiquent par message-passing. Un processus s'inscrit auprès du système d'exploitation pour recevoir certaines classes de messages, et les messages lui sont envoyés quand ils apparaissent.

De nombreuses routines de messages comme le changement de taille des fenêtres peuvent être gérées par le système plutôt que par l'application elle-même.

ARexx est un langage de programmation interprété pour les IPC. L'utilisateur écrit des scripts ARexx qui permettent à des applications de communiquer entre elles.

1.18 Intuition

Intuition est une couche logicielle qui gère certaines tâches automatiquement, libérant les programmes d'application de quelques responsabilités courantes. Les applications n'ont pas besoin de savoir si les fenêtres sont changées de taille, rafraichies, déplacées, bougées d'avant en arrière, activées ou

désactivées, iconifiées et ainsi de suite. Evidemment, les applications peuvent toujours choisir de gérer elles-mêmes ces événements. Ceci est en complète opposition avec ce qui existe sur le système du Macintosh dans lequel l'application doit gérer chaque événement.