

**Υπολογιστές Amiga**  
**Ιούνιος 2006**  
**Μητρέγκας Γεώργιος (Α.Μ.: 3560)**  
**Μπουδαλάκης Παντελής (Α.Μ. 3825)**

**Επιβλέπων Καθηγητής: Μηνάς Δασυγένης**

*Πρόλογος*

Η Amiga είναι μια οικογένεια ηλεκτρονικών υπολογιστών οι οποίοι άρχισαν να πρωτοσχεδιάζονται το 1982. Το αρχικό hardware της Amiga και την παρουσιάστηκε το 1985 στην αγορά. Βασισμένη στην γενιά μικροεπεξεργαστών Motorola 68k, διέθετε chipset ειδικής κατασκευής με προηγμένη τεχνολογία γραφικών και ήχου και το πρωτοποριακό Multitasking software, κάτι για το οποίο ξεχωρίζει και το λειτουργικό σύστημα της Amiga, το Amiga OS. Ενώ ο M68k ήταν ένας 32-μπιτος επεξεργαστής, η αρχική έκδοση που χρησιμοποιούσε η Amiga διέθετε μία 16-μπιτη αρτηρία δεδομένων. Τα επόμενα μοντέλα όμως χρησιμοποίησαν πλήρη 32-μπιτο σχεδιασμό. Η Amiga απέκτησε μεγάλη φήμη στην Ευρώπη κυρίως επειδή αποτελούσε φθηνότερη πρόταση από τους Apple Macintosh και IBM-PC. Επίσης βρήκε εφαρμογές και σε διάφορες επιχειρήσεις που είχαν να κάνουν με παραγωγή video (π.χ. τηλεοπτικούς σταθμούς). Εμπορικά η Amiga γνώρισε επιτυχία κυρίως ως οικιακός υπολογιστής.

## Hardware

Ως πυρήνα η Amiga χρησιμοποίησε απλά σχεδιασμένους συνεπεξεργαστές, χρήσιμους για οπτικοακουστική κωδικοποίηση και animation. Αυτό έδινε την ελευθερία στην κεντρική μονάδα επεξεργασίας (CPU) να ασχολείται με άλλες διεργασίες, προσφέροντας στην Amiga πλεονέκτημα σε αρκετούς τομείς, σε σχέση με τους ανταγωνιστές της.

Η πλατφόρμα εισήγαγε επίσης και άλλες καινοτομίες. Η Amiga CDTV, για παράδειγμα, ήταν ο πρώτος υπολογιστής που περιελάμβανε drive για CD-ROM σαν στάνταρ, καθώς και ένας και ένας από τους πρώτους που δεν περιελάμβαναν drive δισκέτας. Οι Amiga ήταν επίσης οι πρώτοι υπολογιστές όπου ήταν δυνατή η φθηνή δειγματοληψία ήχου και η ψηφιοποίηση video.

Από το 2000 περίπου το hardware έχει φτάσει σε σημείο να προσομοιώνεται από προσομοιωτές Amiga σε διάφορες πλατφόρμες, ικανούς να αναπαράγουν τις λειτουργίες του hardware της Amiga στο software.

*Κεντρική μονάδα επεξεργασίας*

Όλα τα μοντέλα Commodore Amiga κάνουν χρήση CPU της Motorola, βασισμένων στην αρχιτεκτονική M68k. Στους Desktop style Amiga η CPU προσαρμολόγεται σε μία θυγατρική κάρτα (daughterboard). Επίσης κυκλοφόρησαν

Amiga χαμηλού κόστους με τη CPU ενσωματωμένη στη motherboard. Σε όλα τα μοντέλα της Amiga η CPU μπορούσε να αναβαθμιστεί μέσω μίας κάρτας επέκτασης ή με απευθείας αντικατάσταση. Οι κάρτες CPU συνοδεύονταν συχνά από διασυνδέσεις για σκληρούς δίσκους και θέσεις μνήμης ενσωματωμένες στη μητρική, διευκολύνοντας έτσι τη βασική Amiga από αυτές τις διεργασίες. Αργότερα η Amiga One χρησιμοποίησε Power PC CPU αντί για 68k CPU.

### ***Chipset***

Το Amiga chipset αποτελείται από ένα αριθμό τσιπ. Τρεις είναι οι γενιές των chipset που χρησιμοποιήθηκαν σε διάφορα μοντέλα της Amiga. Το πρώτο ήταν το OCS που ακολουθήθηκε από το από το ECS και τελικά το AGA. Τα τρία αυτά chipset έχουν κοινό το ό,τι διαχειρίζονται γραφικά, ψηφιακό ήχο και διασυνδέουν διάφορα περιφερειακά (π.χ. CPU, μνήμη και δισκέτες).

### ***Γραφικά***

Όλοι οι Amiga μπορούν να εμφανίσουν πλήρη οθόνη και γραφικά με χρώματα. Όλοι οι Amiga μπορούν να εμφανίσουν γραφικά με 32, 64 ή 4096 χρώματα. Μοντέλα με AGA chipset (A1200 και A4000) έχουν καταστάσεις με 128, 256 και 262,000 χρώματα και μια παλέτα που επεκτείνεται από τα 4096 στα 16.8 εκατομμύρια χρώματα.

Το chipset της Amiga μπορούσε να κάνει gunlock – να προσαρμόσει τον χρόνο ανανέωσης της οθόνης έτσι ώστε να ταιριάζει με ένα σήμα video NTSC ή PAL. Αυτό επέτρεπε στην Amiga να επικαλύπτει μια πηγή βίντεο με γραφικά. Αυτή η δυνατότητα έκανε την Amiga διάσημη σε πολλές εφαρμογές, όπως τα εφέ CGI.

### ***Ήχος***

Το τσιπ ήχου που ονομάστηκε Paula, υποστηρίζει τέσσερα κανάλια ήχου (δύο για το αριστερό ηχείο και δύο για το δεξί) με ανάλυση 8 bits για κάθε κανάλι και 6-μπιτο έλεγχο έντασης. Η αναλογική έξοδος συνδέεται σε ένα χαμηλοπερατό φίλτρο, το οποίο αποκόπτει τις υψηλές συχνότητες όταν η Amiga χρησιμοποιεί χαμηλότερο ρυθμό δειγματοληψίας. Η φωτεινότητα του Power LED της Amiga υποδεικνύει την κατάσταση του φίλτρου. Το Paula μπορεί να διαβάσει απευθείας από την RAM του τσιπ, χρησιμοποιώντας άμεση προσπέλαση μνήμης (DMA). Ο ήχος δηλαδή δημιουργείται χωρίς παρέμβαση της CPU.

Η ποιότητα ήχου της Amiga και το ό,τι το hardware είναι ευρέως διαδεδομένο και εύκολα διευθυνσιοδοτούμενο από το software, ήταν τα ιδιαίτερα χαρακτηριστικά της Amiga που δεν υπήρχαν σε άλλες υπολογιστικές πλατφόρμες. Υπάρχουν κάρτες ήχου τρίτου μέρους που προωθούν λειτουργίες DSP, απευθείας εγγραφή σε δίσκο, πολλαπλά κανάλια ήχου και 16-μπιτες και άνω αναλύσεις.

### ***ROM***

Το κλασσικό λειτουργικό σύστημα της Amiga αποτελούταν από το Intuition και το Workbench. Στο μοντέλο Amiga 1000, πρώτα φορτωνόταν το Intuition με μία

δισκέτα και μετά ακολουθούσε το Workbench. Τα μεταγενέστερα μοντέλα κρατούσαν το Intuition σε μία ROM, βελτιώνοντας τους χρόνους έναρξης. Τα μοντέλα μπορούσαν να αναβαθμιστούν αλλάζοντας τη ROM.

### ***Hardware τρίτου μέρους***

Πολλές κάρτες επέκτασης παρήχθησαν από την Amiga με σκοπό να βελτιώσουν την απόδοση του hardware, όπως επεκτάσεις μνήμης, ελεγκτές SCSI, κάρτες CPU και γραφικών. Άλλες αναβαθμίσεις περιλαμβάνουν κάρτες Ethernet, modems, κάρτες ήχου, USB, επιπλέον σειριακές θύρες και ελεγκτές IDE.

## **Λειτουργικό σύστημα AmigaOS**

Το AmigaOS είναι το εγγενές λειτουργικό σύστημα προεπιλογής του προσωπικού υπολογιστή Amiga. Αναπτύχθηκε πρώτα από την Commodore International, και πρωτοεμφανίστηκε το 1985 με τον Amiga 1000. Έτρεχε στην σειρά 32μπιτων μικροεπεξεργαστών της Motorola, 68k, εκτός από AmigaOS 4 που τρέχει στους μικροεπεξεργαστές PowerPC. Στην κορυφή ενός πυρήνα πολλαπλών καθηκόντων (multitasking) αποκαλούμενου Exec, περιλαμβάνει μια αφαίρεση του μοναδικού υλικού της Amiga, ένα λειτουργικό σύστημα δίσκων αποκαλούμενο AmigaDOS, ένα σύστημα windowing αποκαλούμενο Intuition και μία διασύνδεση γραφικών, το Workbench. Μια διασύνδεση γραμμών εντολής είναι επίσης διαθέσιμη.

Το AmigaOS μπορεί να διαιρεθεί σε δύο μέρη: το Kickstart (ROM) και τους δίσκους Workbench. Συνηθιζόταν οι εκδόσεις Kickstart και Workbench να κυκλοφορούν μαζί, για να κάνει χρήση η μία με της άλλης. Από το Workbench 3.5 και μετά, η AmigaOS έχει γίνει λογισμικό-μόνο, τυποποιώντας στην έκδοση 3.1 το Kickstart στη ROM.

### ***Kickstart***

Kickstart είναι το όνομα που δίνεται στη ROM εκκίνησης. Στο πρώτο μοντέλο της Amiga, το A1000, αυτό φορτωνόταν από μία δισκέτα σε ένα πρόσθετο τμήμα της μνήμης αποκαλούμενο εγγράψιμο κέντρο ελέγχου (WCS), αν και τελικά το Kickstart ενσωματώθηκε σε ένα τσιπ ROM μέσα στον υπολογιστή. Το Amiga 1000 μπορούσε να τροποποιηθεί για να πάρει αυτά τα τσιπ, και όλα τα επόμενα μοντέλα της Amiga χρησιμοποιούσαν τσιπ ROM.

Το Kickstart περιέχει τον κώδικα που απαιτείται για να εκκινήσει το τυποποιημένο hardware της Amiga. Το Kickstart περιείχε επίσης πολλά μέρη του λειτουργικού συστήματος της Amiga, όπως το Exec, το Intuition και τον πυρήνα AmigaDOS. Αυτό σήμαινε ότι ένα απλά ανοιχτό Amiga είχε ήδη πολλά ουσιαστικά μέρη του λειτουργικού συστήματος διαθέσιμα. Οι πιο πρόσφατες εκδόσεις Kickstart περιείχαν τους οδηγούς για τους ελεγκτές IDE και SCSI, τις θύρες καρτών PC και για διάφορα άλλα μέρη του υλικού της Amiga. Μπορεί να συγκριθεί με το BIOS συν τον

κύριο πυρήνα των Windows στα PC της IBM, εντούτοις έχει πολύ περισσότερη λειτουργικότητα διαθέσιμη στο χρόνο εκκίνησης.

Με το λογισμικό τρίτου μέρους, είναι δυνατό να υπάρξει ένα διαφορετικό Kickstart φορτωμένο στη RAM για να χρησιμοποιηθεί αντί αυτού της ROM - παραδείγματος χάριν το Kickstart 1.3 μπορεί να φορτωθεί προκειμένου να τρέξει τα παλαιά παιχνίδια που είναι ασυμβίβαστα με το Kickstart 2.0 και άνω. Αυτά τα προγράμματα καλούνται softkickers. Υπάρχουν επίσης διακόπτες Kickstart που επιτρέπουν στους χρήστες να έχουν περισσότερα από ένα Kickstart τσιπ μέσα στον υπολογιστή και να διαλέγουν ποιο θέλουν να χρησιμοποιήσουν.

## ***Workbench***

Workbench είναι το όνομα που δίνεται για το λογισμικό πυρήνα του λειτουργικού συστήματος που δεν αποθηκεύεται στο Kickstart ROM και επίσης για το εγγενές “κέλυφος” (shell) γραφικών της Amiga. Το περιβάλλον Workbench δεν είναι απαραίτητο να φορτωθεί για να τρέξει το λογισμικό.

Όπως προτείνει το όνομα, χρησιμοποιείται η μεταφορά ενός πάγκου εργασίας, παρά μία επιφάνεια εργασίας, οι κατάλογοι απεικονίζονται ως συρτάρια, τα εκτελέσιμα αρχεία είναι εργαλεία και τα αρχεία δεδομένων είναι projects. Σε πολλές άλλες πτυχές η διασύνδεση μοιάζει με το MAC OS, με τον κύριο desktop να παρουσιάζει τις εικόνες των εισερχομένων δισκετών και των partitions του σκληρού, και μία ενιαία μπάρα επιλογών στην κορυφή κάθε οθόνης. Αντίθετα από το Macintosh, το τυποποιημένο ποντίκι της Amiga έχει δύο κουμπιά.

Ένα μοναδικό χαρακτηριστικό γνώρισμα του Workbench είναι οι πολλαπλές οθόνες. Αυτές είναι εννοιολογικά παρόμοιες με τα εικονικά desktop ή τους χώρους εργασίας του συστήματος Windows X, αλλά παράγονται δυναμικά από τα προγράμματα εφαρμογής ανάλογα με τις ανάγκες. Κάθε οθόνη μπορεί να έχει διαφορετική ανάλυση και βάθος χρώματος. Οι οθόνες μπορούν επίσης να συρθούν πάνω-κάτω από τις μπάρες τίτλου τους. Στις παλαιότερες Amiga αυτή η λειτουργία παρείχετο από τα συνήθη chipsets που ήταν σχεδιασμένα ειδικά για την πλατφόρμα. Το “σύρε κι άφησε” μεταξύ διαφορετικών οθονών είναι δυνατό επίσης,

Κάτω από το Workbench είναι το σύστημα windowing, Intuition. Αυτό ελέγχει και σχηματίζει τις οθόνες, τα παράθυρα και τα κουμπιά, και χειρίζεται τα δεδομένα από το πληκτρολόγιο και το ποντίκι, περνώντας μηνύματα στα προγράμματα.

## ***Workbench 2.0 (Βελτιώσεις στη διασύνδεση χρήστη)***

Μέχρι τη δημιουργία του Workbench 2.0, δεν υπήρξε κανένα ενιαίο πρότυπο σχεδιασμού - οι υπεύθυνοι για την ανάπτυξη εφαρμογής έπρεπε να γράψουν τα widgets τους (και κουμπιά και επιλογές), με το Intuition που παρείχε την ελάχιστη υποστήριξη. Με το Workbench 2.0 έκανε την εμφάνισή του το gadtools.library, που παρείχε σύνολα τυποποιημένων widget, και ο Οδηγός Ύφους Διασύνδεσης Χρήστη της Amiga, τα οποίοι εξηγούσαν πώς έπρεπε να σχεδιαστούν με συνέπεια οι εφαρμογές.

Το Workbench 2.0 προσέθεσε επίσης υποστήριξη για τις δημόσιες οθόνες. Αντί της οθόνης Workbench που ήταν η μόνη διαμοιράσιμη οθόνη, οι εφαρμογές θα μπορούσαν να δημιουργήσουν τις δικές τους ονομαστικές οθόνες, τις οποίες θα μοιράζονταν με άλλες εφαρμογές.

Το Workbench 2.0 εισήγαγε παράλληλα το AmigaGuide, μια μηχανή αναζήτησης και σήμανσης υπερκειμένων, για την παροχή διαδικτυακής βοήθειας μέσα στις εφαρμογές. Εισήγαγε επίσης τον Installer, ένα τυποποιημένο πρόγραμμα εγκαταστάσεων λογισμικού, οδηγούμενο από μια γλώσσα προγραμματισμού όμοια της LISP.

### ***ARexx***

Το Amiga OS παρείχε υποστήριξη για τη γλώσσα Rexx. Κλήθηκε ARexx (από το «Amiga Rexx») και ήταν μια γλώσσα χειρογράφων που επέτρεπε τον πλήρη προγραμματισμό του λειτουργικού συστήματος, καθώς επίσης και την διαπρογραμματική επικοινωνία. Η κατοχή μιας ενιαίας γλώσσας προγραμματισμού για οποιαδήποτε εφαρμογή στο λειτουργικό σύστημα ήταν πιο συμφέρουσα και αποδοτική, από το να πρέπει να μαθευτεί μια νέα γλώσσα για κάθε εφαρμογή.

### ***Βιβλιοθήκες***

Η κύρια τεχνική ρύθμισης στο AmigaOS είναι βασισμένη στις δυναμικά-φορτωμένες κοινές βιβλιοθήκες που αποθηκεύονται είτε ως αρχείο στο δίσκο με μια επέκταση ονομάτων αρχείου «.library» είτε στο Kickstart ROM.

## **Σύστημα Αρχείων στο AmigaOS**

Στην ιστορία της Amiga διακρίνουμε τέσσερις εκδόσεις συστήματος αρχείων. Αυτές είναι οι:

- I. OFS (Old File System)
- II. FFS (Fast File System)
- III. PFS (Professional File System)
- IV. SFS (Smart File System)

Αρχικά όμως θα αναφερθούμε στο λειτουργικό σύστημα δίσκων AmigaDOS και τους «χειριστές» (handlers).

### ***AmigaDOS***

Το AmigaDOS παρέχει το λειτουργικό σύστημα δίσκων του AmigaOS. Αυτό περιλαμβάνει τα συστήματα αρχείων, το χειρισμό καταλόγων και αρχείων, τη διασύνδεση γραμμών εντολής, επαναπροσανατολισμό αρχείων, παράθυρα κονσολών κ.α. Στο AmigaOS 1.x, το AmigaDOS βασίστηκε σε μία θύρα TRIPOS από τη MetaComCo, που γράφτηκε σε BCPL. Μεγάλα ποσοστά λειτουργικότητας ήταν μόνο διαθέσιμα στα προγράμματα που γράφτηκαν σε BCPL και όχι σε όσους χρησιμοποιούσαν άλλες γλώσσες. Το πρόγραμμα τρίτου μέρους AmigaDOS Resource Project παρείχε ισοδύναμη λειτουργικότητα στους προγραμματιστές C.

Από το AmigaOS 2.x και μετά, το AmigaDOS ξαναγράφηκε σε C και σε Assembler, διατηρώντας την πλήρη συμβατότητα με το πρόγραμμα BCPL, και ενσωματώνοντας το μεγαλύτερο μέρος του ARP στο λειτουργικό σύστημα.

Τα partitions και τα φυσικά drives αναφέρονται χαρακτηριστικά ως DF0: (drive δισκέτας 0), DH0: (σκληρός δίσκος 0) κ.λ.π. Εντούτοις, αντίθετα από πολλά λειτουργικά συστήματα, έξω από τις ενσωματωμένες συσκευές όπως η DF0: αυτά τα ονόματα είναι συνολικά αυθαίρετα. Παραδείγματος χάριν, ένα partition σκληρών δίσκων θα μπορούσε να ονομασθεί HARDDISK: ή A: ή HD0: όταν χωρίστηκε. Τα ονόματα των τομέων (volumes) έχουν την ίδια δομή με τα ονόματα συσκευών. Έτσι ένα partition δίσκου στη συσκευή DH0: μπορεί να έχει το όνομα τομέα BOOT:. Επιπλέον, τα εικονικά ονόματα τομέα θα μπορούσαν να τεθούν με την εντολή προσδιορισμού (assign) σε οποιοδήποτε κατάλογο ή συσκευή. Παραδείγματος χάριν, τα προγράμματα όριζαν συχνά ένα εικονικό όνομα τομέα στον κατάλογο εγκαταστάσεων τους. Αυτό επιτρέπει τον εύκολο επανεντοπισμό των εγκατεστημένων προγραμμάτων.

## *Χειριστές*

Τα υψηλότερα επίπεδα της διαχείρισης των συσκευών και των πόρων ελέγχονται από τους χειριστές, οι οποίοι δεν είναι βιβλιοθήκες, αλλά εντολές (tasks) και επικοινωνούν με διαβίβαση μηνυμάτων. Ένας σημαντικός τύπος χειριστών είναι οι χειριστές του συστήματος αρχείων. Το AmigaOS μπορεί να χρησιμοποιήσει οποιοδήποτε σύστημα για το οποίο έχει γραφτεί ένας χειριστής, μια δυνατότητα που έχει χρησιμοποιηθεί από προγράμματα όπως το CrossDOS και από μερικά «εναλλακτικά» συστήματα αρχείων των OFS και FFS. Αυτά τα συστήματα αρχείων επιτρέπουν σε κάποιον να προσθέσει νέα χαρακτηριστικά γνωρίσματα τα οποία δεν βρίσκονται στο τυποποιημένο λειτουργικό σύστημα.

Οι χειριστές εκθέτουν τα χαρακτηριστικά ονόματα των συσκευών στο DOS, το οποίο μπορεί να χρησιμοποιηθεί για να υπάρξει πρόσβαση στο περιφερειακό που συνδέεται με τον χειριστή.

Τα ονόματα συσκευών είναι αλληλουχίες χαρακτήρων, όπου δεν διακρίνουμε κεφαλαία και μικρά. Αυτές ακολουθούνται από μια άνω και κάτω τελεία. Μετά από την άνω και κάτω τελεία ένας "προσδιοριστής" μπορεί να προστεθεί, ο οποίος δίνει στο χειριστή πρόσθετες πληροφορίες για το τι προσεγγίζεται και πώς. Στην περίπτωση του συστήματος αρχείων, ο προσδιοριστής αποτελείται συνήθως από μια διαδρομή προς ένα αρχείο. Για άλλους χειριστές, οι προσδιοριστές θέτουν συνήθως τα χαρακτηριστικά του επιθυμητού καναλιού εισόδου-εξόδου. Για τον SER (οδηγός σειριακής θύρας), παραδείγματος χάριν, ο προσδιοριστής περιέχει τον ρυθμό δεδομένων, τα bits έναρξης και λήξης κ.λπ.).

Το σύστημα αρχείων εκθέτει τα ονόματα των drives ως ονόματα των συσκευών. Για παράδειγμα, το DF0: εξ ορισμού αναφέρεται στο πρώτο drive δισκέτας στο σύστημα. Σε πολλά συστήματα το DH0: χρησιμοποιείται για να αναφερθεί στον πρώτο σκληρό δίσκο.

Το σύστημα αρχείων εκθέτει επίσης τα ονόματα των τομέων, με την ίδια σύνταξη με τα ονόματα συσκευών. Τα ονόματα τομέων προσδιορίζουν το συγκεκριμένο μέσο στο διαχειριζόμενο από το σύστημα αρχείων drive. Εάν το DF0: περιέχει έναν δίσκο που ονομάζεται «Workbench», τότε το Workbench θα είναι ένα όνομα τομέα που μπορεί να χρησιμοποιηθεί για να έχουμε πρόσβαση στα αρχεία του DF0:.

Εάν κάποιος ήθελε να έχει πρόσβαση σε ένα αρχείο που ονομάστηκε «Amp» και βρίσκεται στον κατάλογο «Win» του δίσκου με το όνομα «Work» στο drive DF0:, θα μπορούσε να γράψει:

DF0:Win/Amp

ή

Work:Win/Amp

Εντούτοις, αυτά δεν είναι απολύτως ισοδύναμα, δεδομένου ότι όταν χρησιμοποιείται η τελευταία μορφή, το σύστημα ξέρει ότι ο επιθυμητός τομέας είναι ο «Work» και όχι οποιοσδήποτε τομέας στον DF0:. Επομένως, όποτε ένα ζητούμενο αρχείο του «Work» προσεγγίζεται χωρίς το όνομα του τομέα να αναφέρεται, θα εμφανιστεί το μήνυμα:

«Παρακαλώ εισάγετε τον τομέα Work σε κάποιο drive»

Τα προγράμματα πρέπει συχνά να έχουν πρόσβαση σε αρχεία χωρίς γνώση της φυσικής θέσης τους (είτε drive είτε τομέα). Ξέρουν μόνο τη «λογική διαδρομή» του αρχείου, δηλ. εάν το αρχείο είναι μια βιβλιοθήκη, ένα αρχείο τεκμηρίωσης, μια μετάφραση των μηνυμάτων του προγράμματος κ.λ.π. Αυτό λύνεται στο AmigaOS με την χρήση "οριστών" (assigns). Ένας οριστής ακολουθεί, πάλι, την ίδια σύνταξη με ένα όνομα συσκευών. Παρόλα αυτά, στοχεύει σε έναν κατάλογο μέσα στο σύστημα αρχείων. Η θέση την οποία δείχνει ένας οριστής μπορεί οποιαδήποτε στιγμή να αλλάξει από τον χρήστη. Τα πρότυπα των οριστών που χρησιμοποιεί γενικά AmigaOS είναι:

\* SYS: Δείχνει τον κύριο κατάλογο του drive εκκίνησης. αυτός είναι ο μόνος οριστής που δημιουργείται αυτόματα από το Kickstart.

\* LIBS: Δείχνει έναν κατάλογο που περιέχει τις βιβλιοθήκες του συστήματος, συνήθως SYS: Libs/

\* DEVS: Δείχνει έναν κατάλογο που περιέχει τις συσκευές του συστήματος, συνήθως SYS: Devs/

\* C: Δείχνει έναν κατάλογο που περιέχει τις εντολές κελύφους, συνήθως SYS: C

\* PROGDIR: Αυτό δεν είναι κανονικά προσιτό σε έναν χρήστη, αλλά όλα τα προγράμματα το ορίζουν έτσι ώστε να μην χρειάζεται να ξέρουν που έχουν αποθηκευτεί σε δίσκο.

## ***I. Old File System (OFS)***

Αρχικά ονομαζόταν Amiga File System. Έλαβε όμως τον χαρακτηρισμό Old όταν εκδόθηκε το Fast File System. Παρά το ότι χρησιμοποίησε μπλοκ των 512 bytes, διατήρησε το πρώτο μικρό τμήμα κάθε μπλοκ για τα metadata, αφήνοντας μια πολύ ακανόνιστη χωρητικότητα δεδομένων μόνο 488 bytes ανά μπλοκ. Ήταν κατάλληλο μόνο για δισκέτες, και αντικαταστάθηκε σύντομα.

Το Amiga χρησιμοποιεί κωδικοποίηση συχνότητας MFM όταν διαχειρίζεται δισκέτες. Ο κανόνας κωδικοποίησης είναι ο εξής:

Δεδομένα χρήστη	MFM bits
1	01
0	10 εάν ακολουθείται από bit 0
0	00 εάν ακολουθείται από bit 1

Υπάρχουν 80 κύλινδροι σε μια δισκέτα Amiga. Κάθε κύλινδρος έχει 2 MFM τροχιές, μια σε κάθε πλευρά της δισκέτας. Οι διπλής πυκνότητας δισκέτες (DD) έχουν 11

τομείς ανά τροχιά MFM, ενώ οι δισκέτες υψηλής πυκνότητας (HD) έχουν 22 τομείς. Η γεωμετρία μιας δισκέτας Amiga είναι η ακόλουθη:

- DD δισκέτες: 512 bytes/τομέα, 11 τομείς/τροχιά, 2 τροχιές/κύλινδρο, 80 κύλινδροι/δισκέτα
- HD δισκέτες: 512 bytes/τομέα, 22 τομείς/διαδρομή, 2 τροχιές/κύλινδρο, 80 κύλινδροι/δισκέτα.

Μία DD δισκέτα έχει  $11 * 2 * 80 = 1760$  (0 έως 1759) μπλοκς, ενώ μία HD έχει  $22 * 2 * 80 = 3520$  μπλοκς. Το Amiga αποθηκεύει 880 KB στις DD και 1760 KB στις HD δισκέτες. Το τυποποιημένο σύστημα αρχείων του Amiga έχει μια εσωτερική 32-μπιτη ευρεία παράμετρο όφσετ. Αυτή ορίζει από πού να αρχίσει η λειτουργία ανάγνωσης-γραφής. Το μεγαλύτερο μέγεθος για έναν δίσκο Amiga είναι επομένως  $2^{32} = 4$  GB. Αυτό το όριο άλλαξε με τα Amiga OS 3.5 και 3.9 και είναι τώρα  $2^{64} = 4$  Exabyte. Το μπλοκ δεδομένων του OFS μπορεί να αποθηκεύσει μέχρι 488 bytes από τα 512 χρησιμοποιούμενα. Το κύριο μπλοκ (Rootblock) βρίσκεται στο μπλοκ 880 για τις DD δισκέτες και στο μπλοκ 1760 για τις HD. Ο ακριβής υπολογισμός για το πού είναι αποθηκευμένο γίνεται ως εξής:

```
numCyls = highCyl - lowCyl + 1
```

```
highKey = numCyls * numSurfaces * numBlocksPerTrack - 1
```

```
rootKey = INT (numReserved + highKey) / 2
```

Το κύριο μπλοκ περιέχει τις πληροφορίες για την δισκέτα: το όνομά της, την ημερομηνία μορφοποίησής της, κ.λ.π. Περιέχει επίσης τις πληροφορίες για την πρόσβαση στα αρχεία/καταλόγους/συνδέσεις που βρίσκονται στον κύριο κατάλογο. Οι χαρακτήρες «/» και «:» είναι απαγορευμένοι στα ονόματα αρχείων και τομέων, αλλά τα `*!@#%|^+&_ () = \ - [] {} «», <>.;` και τα τονισμένα γράμματα όπως `â` επιτρέπονται.

Τα πεδία ημερομηνίας στο κύριο μπλοκ (και άλλα μπλοκς) είναι δομημένα υπό μορφή ΗΜΕΡΩΝ, ΛΕΠΤΩΝ και TICKS. Το πεδίο ΗΜΕΡΕΣ περιέχει τον αριθμό ημερών από την 1η Ιανουαρίου. 1978. Τα ΛΕΠΤΑ είναι ο αριθμός των λεπτών που έχουν περάσει από τα μεσάνυχτα και τα TICKS εκφράζονται σαν το 1/50s ενός δευτερολέπτου. Μια τιμή ημέρας μηδέν θεωρείται άκυρη από τα περισσότερα προγράμματα.

Για να βρει ένα αρχείο, έναν κατάλογο ή μια σύνδεση στα αφαιρούμενα μέσα, το Amiga πρέπει να υπολογίσει την χαρακτηριστική τιμή "hash" με έναν αλγόριθμο αποκαλούμενο συχνά συνάρτηση Hash<sup>1</sup>. Αφού έχει υπολογιστεί η τιμή hash, χρησιμοποιείται έπειτα για πρόσβαση στον πίνακα HashTable. Το HashTable[HashValue] περιέχει τον αριθμό του πρώτου μπλοκ του αντικειμένου.

Υπάρχει περίπτωση διαφορετικά ονόματα να δίνουν την ίδια τιμή hash. Εάν περισσότερα από ένα ονόματα έχουν την ίδια hash, τα αντίστοιχα μπλοκς (για τα αρχεία και τον κατάλογο μόνο) αποθηκεύονται σε μία συνδεδεμένη λίστα. Παραδείγματος χάριν: τα αρχεία `file_1a`, `file_24` και `file_5u` έχουν την ίδια τιμή hash.

Τα αρχεία αποτελούνται από ένα μπλοκ επικεφαλίδας αρχείων, που περιέχει πληροφορίες σχετικά με το αρχείο (μέγεθος, τελευταία φορά πρόσβασης, δείκτες μπλοκ δεδομένων κ.λ.π.), και από τα μπλοκ δεδομένων, τα οποία περιέχουν τα πραγματικά δεδομένα. Αν ένα αρχείο είναι μεγαλύτερο από όσο πρέπει, τότε δημιουργούνται μπλοκ επέκτασης αρχείων. Αυτά οργανώνονται σε μία συνδεδεμένη λίστα, η οποία αρχίζει από το μπλοκ επικεφαλίδας αρχείου.



---

<sup>1</sup> Η συνάρτηση Hash είναι μια μέθοδος μετατροπής δεδομένων σε έναν μικρό αριθμό που μπορεί να χρησιμεύσει ως ένα ψηφιακό «δακτυλικό αποτύπωμα» των συγκεκριμένων δεδομένων.

## ***II. Fast File System (FFS)***

Το παλαιό σύστημα αρχείων Amiga (OFS) ήταν πάρα πολύ αργό για να συμβαδίσει με τους σκληρούς δίσκους. Το FFS διαφέρει κυρίως στη διαγραφή των περιττών πληροφοριών. Τα μπλοκ δεδομένων δεν περιέχουν παρά μόνο δεδομένα, επιτρέποντας στο σύστημα αρχείων να διαχειρίζεται τη μεταφορά μεγάλων τμημάτων δεδομένων άμεσα από τον host adapter στον τελικό προορισμό. Επίσης στις παλαιές εκδόσεις AmigaDOS υπήρξε ένα πρόβλημα κατά την εφαρμογή της συνάρτησης Hash σε διεθνείς χαρακτήρες ASCII. Το πρόβλημα αυτό λύθηκε στο FFS. Τα χαρακτηριστικά υπόλοιπα χαρακτηριστικά του FFS δεν διαφέρουν από τον OFS και αναλύονται στην προηγούμενη παράγραφο.

Πιο πρόσφατα, το MorphOS και το AmigaOS 4 έχουν εισαγάγει την υποστήριξη για το ελαφρώς ενημερωμένο Fast File System 2 ή FFS2. Όπως το κανονικό FastFileSystem, το FFS2 είναι ξεπερασμένο έναντι των πιο σύγχρονων συστημάτων αρχείων. Στερείται την αξιοπιστία και τα προηγμένα χαρακτηριστικά γνωρίσματα των πιο σύγχρονων προσφορών. Όλες οι εκδόσεις FFS αποθηκεύουν έναν «χάρτη» του συστήματος αρχείων σε ένα bitmap που είναι επιρρεπές σε διαγραφή ή φθορά εάν το OS επαναρυθμιστεί ή καταστραφεί κατά τη διάρκεια εγγραφής, με συνέπεια την απώλεια στοιχείων ή την ακύρωση του συστήματος αρχείων. Το FFS2 είναι μια δευτερεύουσα αναπροσαρμογή του FastFileSystem για τα AmigaOS 4 και MorphOS. Εφαρμόζει τη μακροχρόνια υποστήριξη ονομάτων αρχείου διατηρώντας την συμβατότητα με τα παλαιότερα FastFileSystem και το OldFileSystem. Η μόνη σημαντική προσθήκη του είναι η ικανότητα του χειρισμού των πιο μακρών ονομάτων αρχείου, ενώ "πάσχει" ακόμα από τα αρνητικά σημεία του FFS.

## ***III. Professional File System (PFS)***

Το Professional File System είναι ένα σύστημα αρχείων του Amiga που αναπτύχθηκε αρχικά για εμπορικούς σκοπούς. Τείνει να έχει πολύ καλή απόδοση, λόγω της απλότητας του σχεδιασμού του.

Η συσκευή είναι χωρισμένη σε δύο κύριες περιοχές. Στην αρχή της συσκευής είναι το τμήμα των μεταδεδομένων (metadata), που αποτελούνται από ένα κύριο μπλοκ, και μια γενική σειρά από μπλοκς που μπορούν να κατανέμονται ώστε να αποθηκεύουν μεταδεδομένα. Το υπόλοιπο της συσκευής είναι μια άλλη παρακείμενη σειρά από μπλοκς που μπορεί να κατανεμηθεί για την αποθήκευση των δεδομένων. Το τμήμα των μεταδεδομένων χρησιμοποιεί συνήθως ένα ποσοστό της συσκευής, ανάλογα με το μέγεθός της.

Τα μεταδεδομένα αποθηκεύονται ως δέντρο των ενιαίων μπλοκς στο τμήμα μεταδεδομένων. Ολόκληρη η δομή καταλόγου καταγράφεται στα μεταδεδομένα, έτσι ώστε το τμήμα δεδομένων να περιέχει αποκλειστικά τα δεδομένα των αρχείων. Τα

μεταδεδομένα περιγράφουν τη θέση των δεδομένων στα αρχεία, κάτι το οποίο τα καθιστά αρκετά συμπαγή.

Όταν μια αναπροσαρμογή (update) μεταδεδομένων εμφανίζεται, το σύστημα εξετάζει το μπλοκ που περιέχει τα μεταδεδομένα που αλλάζονται, και το αντιγράφει σε ένα πρόσφατα καταναμημένο μπλοκ από το τμήμα μεταδεδομένων. Κατόπιν αλλάζει κατ' επανάληψη τα μεταδεδομένα στο μπλοκ που δείχνει στο προαναφερθέν μπλοκ με τον ίδιο τρόπο. Με αυτόν τον τρόπο, τελικά το κύριο μπλοκ πρέπει να αλλάξει, προκαλώντας την ατομική αναπροσαρμογή των μεταδεδομένων. Το συγκεκριμένο σύστημα αρχείων, τέλος, είναι εύλογα καλό στο να αποτρέπει τον κατακερματισμό των αρχείων.

#### ***IV. Smart File System (SFS)***

Το Smart File System είναι άλλο ένα σύστημα αρχείων που χρησιμοποιείται στους υπολογιστές Amiga. Έχει σχεδιαστεί με σκοπό καλύτερη απόδοση, εξελιξιμότητα και ακεραιότητα. Χρησιμοποιεί μεγέθη μπλοκ που κυμαίνονται από 512 ( $2^9$ ) ως 32768 ( $2^{15}$ ) bytes με ένα μέγιστο μέγεθος 128 GB.

Η καλή απόδοση του συστήματος αρχείων πραγματοποιείται με την ομαδοποίηση των πολλαπλών καταχωρήσεων καταλόγου μέσα σε ένα ενιαίο μπλοκ και με το να συγκεντρώνει τα μπλοκ των μεταδεδομένων σε συστάδες (clusters). Ένα bitmap χρησιμοποιείται για να διατηρεί τροχιές ελεύθερου χώρου, ενώ τα δεδομένα αρχείων τακτοποιούνται σε μια δομή B+ δέντρου.

Η ακεραιότητα διατηρείται με την ενημέρωση ενός αρχείου "επιστροφής" όλων των αλλαγών που γίνονται στα μεταδεδομένα κατά τη διάρκεια μιας ορισμένης χρονικής περιόδου. Το αρχείο γράφεται στο δίσκο πρώτα σε ελεύθερο χώρο και έπειτα τα μπλοκ μεταδεδομένων το επικαλύπτουν άμεσα. Αν το σύστημα υποστεί crash, την επόμενη φορά που θα λειτουργήσει το σύστημα αρχείων, θα παρατηρήσει την ασυνέχεια και θα γυρίσει πίσω στην τελευταία γνωστή σταθερή κατάσταση. Για λόγους απόδοσης, μόνο η ακεραιότητα των μεταδεδομένων εξασφαλίζεται. Τα πραγματικά δεδομένα στα αρχεία μπορούν ακόμα να αλλοιωθούν εάν μια λειτουργία εγγραφής διακοπεί στη μέση.

Ένα ιδιαίτερα ενδιαφέρον χαρακτηριστικό γνώρισμα του SFS είναι η δυνατότητά του να ανασυγκροτείται (defragment) ενώ το σύστημα αρχείων είναι σε λειτουργία. Η διαδικασία ανασυγκρότησης μπορεί να σταματήσει και να ξαναρχίσει αμέσως. Κατά τη διάρκεια της ανασυγκρότησης εξασφαλίζεται η ακεραιότητα των μεταδεδομένων καθώς και των κανονικών δεδομένων.

#### ***Αναφορές***

**[1] Wikipedia:** <http://en.wikipedia.org/wiki/Amiga>

**[2] Amiga Operating System:** [http://www.operating-system.org/betriebssystem/\\_english/bs-amigaos.htm](http://www.operating-system.org/betriebssystem/_english/bs-amigaos.htm)

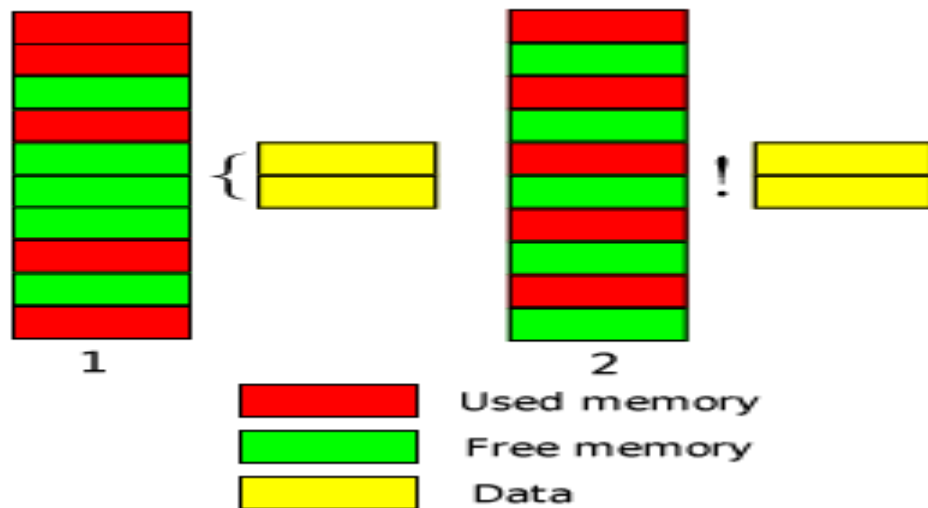
**[3] Amiga – Recognized Formats:**  
<http://disktype.sourceforge.net/doc/ch03s01.html>

**[4] The Amiga Disk File Format:**  
[http://lclevy.free.fr/adflib/adf\\_info.html](http://lclevy.free.fr/adflib/adf_info.html)

## Κατανομή μνήμης στο AmigaOS 4.0

Παλαιότερα, όταν χρησιμοποιούνταν το αρχικό AmigaOS, το σύστημα που χρησιμοποιήθηκε για να διαθέσει τους τομείς της αχρησιμοποίητης μνήμης στα νέα tasks ήταν αρκετά απλό. Το μέγεθος και η θέση των blocks της ελεύθερης μνήμης κρατούνταν σε έναν κατάλογο, και όταν υπήρχε απαίτηση μνήμης, το σύστημα θα διάβαινε αυτόν τον κατάλογο έως ότου βρει ένα block που ήταν αρκετά μεγάλο. Αυτό το block έπειτα χωριζόταν, σε ένα μέρος που επιστράφηκε από τον διαθέτη (*allocator*) για χρήση, ενώ το άλλο μέρος (όλο το κομμάτι του block που δεν χρειαζόνταν) αφήνονταν στον ελεύθερο κατάλογο. Αυτή η μέθοδος ήταν αρκετά αποτελεσματική για τον σκοπό που είχε φτιαχτεί, αλλά με τις αυξανόμενες απαιτήσεις του σύγχρονου υπολογισμού - και φυσικά η επιθυμία να παρουσιαστεί αυτή η νέα έκδοση του λειτουργικού συστήματος - το AmigaOS4.0 έχει εισαγάγει έναν καλύτερο τρόπο για την κατανομή της μνήμης.

### Memory fragmentation (the old way)



Στο σχήμα, βλέπουμε τον τεμαχισμό της μνήμης στο παλιό σύστημα: Στο παράδειγμα 1, το νέο στοιχείο διατίθεται επιτυχώς πρώτο διάστημα δύο-byte που είναι διαθέσιμο στον κατάλογο. Στο παράδειγμα 2, αν και υπάρχει εξίσου ελεύθερο διάστημα στοιχείων είναι πάρα πολύ τεμαχισμένο για να επιτρέψει σε ένα παρακείμενο block 2 byte των δεδομένων να διατεθεί οπουδήποτε στον κατάλογο

Ας εξετάσουμε αρχικά τους περιορισμούς της παλαιάς μεθόδου. Φανταστείτε ότι έχουμε έναν τομέα της ελεύθερης μνήμης του 1 MB στο μέγεθος, και το διαθέτουμε όλο σε blocks ενός byte. Κατόπιν ελευθερώνουμε κάθε δεύτερο block. Θεωρητικά, ο κατάλογος προσφέρει τώρα μισό MB του ελεύθερου τμήματος, αλλά δεν μπορούμε ακόμη και να ικανοποιήσουμε ένα αίτημα για δύο byte, επειδή πουθενά στον κατάλογο δεν υπάρχει ένα block ελεύθερης μνήμης που να είναι τόσο μεγάλο. Αυτό είναι φυσικά ένα πολύ ακραίο παράδειγμα, αλλά στην πράξη μπορούν να προκύψουν παρόμοιες καταστάσεις. Αυτή η επίδραση καλείται «(εξωτερικός) τεμαχισμός» (*external fragmentation*) - ο κατάλογος μνήμης τεμαχίζεται κατά τέτοιο τρόπο ώστε τα αιτήματα κατανομής να μην μπορούν να ικανοποιηθούν.

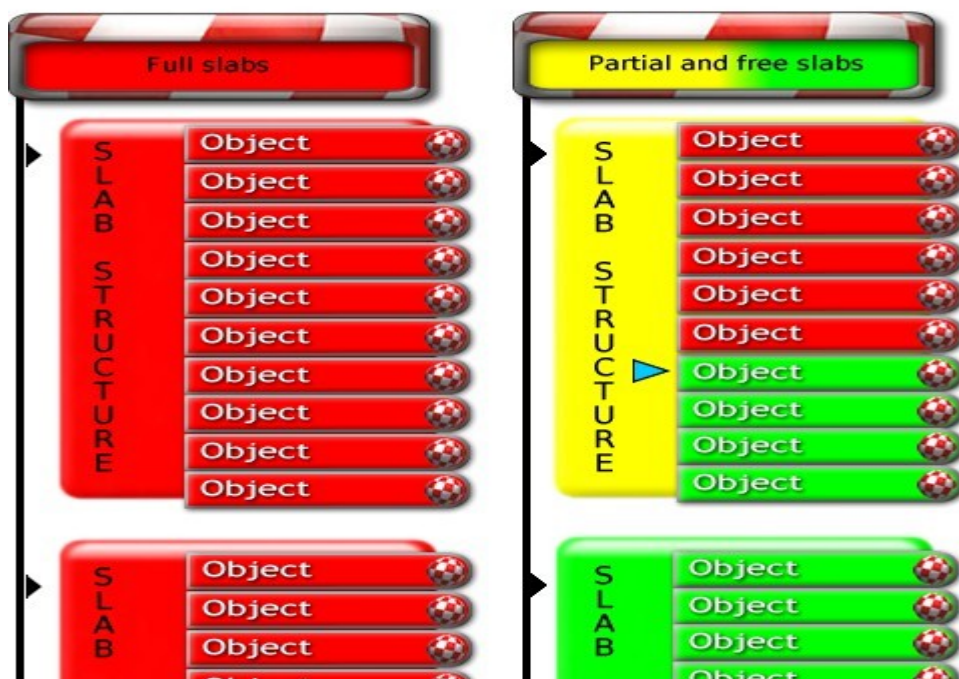
Το δεύτερο μειονέκτημα είναι ότι ο κατάλογος που περιγράφει το μέγεθος και τη θέση των τομέων της ελεύθερης μνήμης μπορεί να γίνει πολύ μεγάλος. Σε ένα σύστημα που δουλεύει καιρό, ο κατάλογος της μνήμης τεμαχίζει όλο και περισσότερο, κάτι που έχει σαν αποτέλεσμα την δημιουργία ενός πολύ μεγάλου καταλόγου χιλιάδων κόμβων. Για κάθε λειτουργία σε αυτόν τον κατάλογο (κατανομή και απελευθέρωση ενός block της μνήμης), πρέπει να περαστεί. Δεδομένου ότι δεν υπάρχει κανένα στοιχείο για το πόσο μεγάλος μπορεί να γίνει ο κατάλογος, δεν υπάρχει επίσης κανένα ανώτερο όριο για το πόσο χρόνο μπορεί να πάρει αυτή η λειτουργία. Κάτι τέτοιο, καθιστά το σύστημα ακατάλληλο για τα συστήματα που λειτουργούν σε πραγματικό χρόνο ή τουλάχιστον κοντά σε τιμές κοντά σε πραγματικό χρόνο.

### ***Κατανομή πλακών (slab allocation)***

Το AmigaOS4.0 εισάγει μια απολύτως νέα αρχιτεκτονική μνήμης που εξετάζει αυτά τα προβλήματα με ένα προηγμένο, ισχυρό και σύγχρονο σύστημα. Το σύστημα όχι μόνο εξετάζει τα παραδοσιακά προβλήματα του τεμαχισμού και τις απαιτήσεις σε πραγματικό χρόνο, αλλά επιτρέπει και άλλες βελτιστοποιήσεις.

Η αρχιτεκτονική μνήμης του AmigaOS4.0 βασίζεται στο σύστημα του *slab allocation* ή διαφορετικά του *object cache* (δείτε αναφορά [1]). Στην ουσία, ο διαθέτης πλακών διαθέτει μόνο τα αντικείμενα ενός ενιαίου μεγέθους, διαθέτοντας αυτά σε μεγαλύτερα batches («πλάκες») από τον χαμηλού επιπέδου διαθέτη σελίδων. Αυτές οι πλάκες χωρίζονται έπειτα σε απομονωτές με το απαραίτητο μέγεθος, και κρατούνται σε έναν κατάλογο στον slab allocator.

Memory lists in the AmigaOS4.0 Slab Allocation system



Η διάθεση ενός αντικειμένου με το διαθέτη πλακών γίνεται μια διαδικασία της απλής αφαίρεσης κόμβων: ο πρώτος κόμβος στην πρώτη πλάκα που περιέχει τους ελεύθερους κόμβους αφαιρείται και επιστρέφεται για χρήση. Δεδομένου ότι ο διαθέτης πλακών κρατά τις ελεύθερες πλάκες ή τις μερικώς ελεύθερες πλάκες σε έναν χωριστό κατάλογο από τις πλήρεις πλάκες, αυτή η λειτουργία μπορεί να διενεργηθεί σε έναν σταθερό χρόνο. Η απελευθέρωση της μνήμης ολοκληρώνεται με την επιστροφή του απομονωτή στο cache του, και την προσθήκη του στον ελεύθερο κατάλογο της αρχικής πλάκας του. Οι πλάκες που είναι απολύτως ελεύθερες μπορούν να επιστραφούν στο *page pool* του συστήματος (αυτή η λειτουργία χρησιμοποιείται *on demand* ( με απαίτηση), και χρησιμοποιούνται *timestamps* για να αποφύγουν την περιττή φόρτωση των στοιχείων, ή όπως είναι γνωστό το «*thrashing*»). Ο εξωτερικός τεμαχισμός είναι ελάχιστος, και ο εσωτερικός τεμαχισμός ελέγχεται και εγγυάται για να μην ξεπεράσει ένα ορισμένο ποσό.

### ***Εναποθήκευση αντικειμένου (object caching)***

Ο διαθέτης πλακών μπορεί επίσης να χρησιμοποιηθεί για να εναποθηκεύσει αντικείμενα. Στον πραγματικό κόσμο πολλές διαδικασίες κατανομής μνήμης θα χρησιμοποιηθούν για να διαθέσουν το ίδιο αντικείμενο. Το σύστημα έχει διάφορες δομές δεδομένων που διατίθενται συχνά (σηματοφόροι, ports μηνυμάτων και άλλα). Κάθε φορά που διατίθενται τέτοιες δομές, πρέπει να τους δοθούν αρχικές τιμές, και όταν διαγράφονται πάλι, πρέπει να «καθαριστούν». Είναι πιθανό, εντούτοις, ότι μια τέτοια δομή θα χρειαστεί πάλι στο μέλλον, έτσι ώστε μπορεί να κρατηθεί στο αρχικοποιημένη κατάσταση της και να επαναχρησιμοποιηθεί αργότερα. Κάτι τέτοιο περαιτέρω μειώνει το φορτίο στις ρουτίνες διαθετών, και βελτιώνει έτσι την απόδοση συστημάτων.

### ***Περισσότερα πλεονεκτήματα***

Ένα άλλο πλεονέκτημα είναι η δυνατότητα να βελτιωθεί η χρήση της cache της CPU. Συνήθως, τα περισσότερα αντικείμενα έχουν «*hot spots*» (κρίσημα σημεία), δηλ. έχουν μερικούς τομείς που χρησιμοποιούνται συχνά. Δεδομένου ότι τις περισσότερες φορές μια μικρή μνήμη αφήνεται αχρησιμοποίητη σε μια πλάκα (το μέγεθος του αντικειμένου να μην είναι ένα πολλαπλάσιο του μεγέθους πλακών), αυτή η πρόσθετη μνήμη μπορεί να χρησιμοποιηθεί «στη μετατόπιση» των κρίσημων σημείων για μερικά bytes για να βελτιστοποιήσει τη δομή μνήμης, που οδηγεί στην καλύτερη χρήση της cache.

Τέλος, το σύστημα μπορεί να επεκταθεί στις πολλαπλές CPU. Σε ένα σύστημα με multi - CPUs, αυτοί οι επεκτεταμένοι slab allocators μεταβάλλονται σχεδόν γραμμικά με τον αριθμό των CPU που υιοθετούνται, κάτι που κάνει αυτήν την επιλογή ιδανική για τέτοια συστήματα (δείτε αναφορά [2]).

Ο συνδυασμός του object caching και του μηχανισμού της κράτησης caches για διαφορετικά block μνήμης (για την εξομοίωση AllocVec/FreeVec) καθιστά τη διαχείριση μνήμης τέμνων-ακρών (*cutting - edge*) του AmigaOS4.0 πολύ αποδοτικότερη, γρηγορότερη, και γενικά με περισσότερες μελλοντικές προεκτάσεις από το παλαιότερο σύστημα. Αυτό είναι μόνο ένα ακόμη παράδειγμα των τεχνικών καινοτομιών που επινοούνται από το AmigaOS4.0, για να μετατραπεί από το καλύτερο λειτουργικό σύστημα του παρελθόντος στο καλύτερο λειτουργικό σύστημα του μέλλοντος.

Περισσότερες πληροφορίες για τα συστήματα διαθετών πλακών καθώς επίσης και την ανάλυση χρόνου εκτέλεσης μπορούν να βρεθούν μέσα στις αναφορές [1] και [2].

## **Αναφορές**

[1] [The Slab Allocator: An Object-Caching Kernel Memory Allocator \(1994\)](#)

Jeff Bonwick, Sun Microsystems

<http://citeseer.ist.psu.edu/bonwick94slab.html>

[2] [Magazines and Vmem: Extending the Slab Allocator to Many CPUs and Arbitraty resources \(2001\)](#)

Jeff Bonwick, Sun Microsystems

Jonathan Adams, California Institute of Technology

<http://citeseer.ist.psu.edu/bonwick01magazines.html>

## **AmigaOS 4.0 - νέο σύστημα μνήμης**

*Σε ένα προηγούμενο στοιχείο, περιγράψαμε πώς το σύστημα μνήμης AmigaOS4.0 λειτουργεί τις κατανομές μνήμης από άποψη διαχείρισης από την κορυφή. Οι caches αντικειμένου λειτουργούν φυσικά στη μνήμη που έχει χαρτογραφηθεί ήδη στο διάστημα εικονικής μνήμης. Αλλά και η εικονική διεύθυνση κυμαίνεται, καθώς επίσης ενώ η φυσική μνήμη πρέπει να προέλθει από κάποια πηγή, επίσης.*

### ***Εικονική κατανομή διαστήματος διευθύνσεων***

Το AmigaOS4.0 χρησιμοποιεί τώρα έναν διαθέτη χαρτών των πόρων για τη διάθεση του εικονικού διαστήματος διευθύνσεων. Βασικά αυτό είναι ένας τρόπος για την διαχείριση ενός συνόλου πόρων (όχι απαραίτητως μνήμη). Για λόγους απόδοσης, χρησιμοποιεί διάφορες τεχνικές βελτιστοποίησης.

Αρχικά, όλα τα ελεύθερα blocks των πόρων κρατιούνται στους διαχωρισμένους σε κενά καταλόγους, δηλ. υπάρχει ένας κατάλογος για κάθε ομάδα της δύναμης δύο πόρων. Αυτό καθιστά τις κατανομές πολύ γρηγορότερες παρέχοντας

ένα μεγαλύτερο και ένα μικρότερο bound για αναζήτηση. Παραδείγματος χάριν, εάν θέλετε να διαθέσετε ένα block  $2^{10}$  bytes, μπορείτε βασικά να αποφύγετε να ψάξετε οποιοδήποτε block μεγέθους κάτω από  $2^{10}$  bytes μόνο και μόνο επειδή δεν θα χωρέσει. Ομοίως, δεν πρέπει να ψάξετε για τα blocks που είναι μεγαλύτερα από, για παράδειγμα, δύο φορές το μέγεθος του block, δεδομένου ότι εκεί μπορεί να υπάρχουν blocks ενός μεγέθους πιο κοντά σε αυτό που χρειαζόμαστε. Οι διαχωρισμένοι κατά μέγεθος ελεύθεροι κατάλογοι βοηθούν στο να στενέψουν την αναζήτηση, τεχνική που κάνει την ίδια την αναζήτηση γρηγορότερη και φέρει καλύτερο αποτέλεσμα στο επίπεδο του κερματισμού.

Επιπλέον, οι χάρτες των πόρων χρησιμοποιούν τις κανονικές caches αντικειμένου για την επιτάχυνση των «μικρών» κατανομών. Οι περισσότερες κατανομές είναι κάτω από ένα ορισμένο μέγεθος. Παραδείγματος χάριν, οι εικονικές διευθύνσεις διατίθενται πάντα στα χοντρά κομμάτια τουλάχιστον μιας «σελίδας» στη μνήμη (4KB). Έτσι είναι λογικό να διατεθούν τα blocks μιας, δύο, τεσσάρων, ή οκτώ σελίδων. Οι caches αντικειμένου παρέχουν μια εύκολη μέθοδο για αυτά τα κοινά μεγέθη, κάνοντας κάθε κατανομή αυτών των μεγεθών να χωράει, μειώνοντας έτσι περαιτέρω τον κατακερματισμό.

### **Φυσική κατανομή σελίδων**

Μια κοινή λειτουργία στην κατανομή μνήμης είναι η ανάθεση των εικονικών διευθύνσεων στις φυσικές θέσεις μνήμης. Η κατανομή της φυσικής μνήμης γίνεται συνήθως διαφορετικά από τις εικονικές κατανομές, δεδομένου ότι είναι απαραίτητο να ελευθερωθεί μόνο μέρος της κατανομής.

Τα de facto πρότυπα στην κατανομή των φυσικών σελίδων είναι μια μέθοδος που εφευρίσκειται από τον Knuth, αποκαλούμενη ως «*buddy system*» (σύστημα φιλαράκων).

Τα *buddy systems* είναι στην ουσία ελεύθεροι κατάλογοι διαχωρισμένοι κατά μέγεθος (δείτε παραπάνω). Για να γίνει διάθεση (*allocate*), το σύστημα ψάχνει ένα ελεύθερο block τουλάχιστον του μεγέθους της κατανομής. Κατόπιν, εάν το block είναι πάρα πολύ μεγάλο, χωρίζεται σε δύο blocks ίσου μεγέθους. Αυτά τα blocks λέγονται buddies. Ένα block επιστρέφει στον κατάλληλο ελεύθερο κατάλογο, και το άλλο χρησιμοποιείται περαιτέρω, ίσως χωρίζοντας το περαιτέρω έως ότου το μέγεθός του να ταιριάζει με αυτό που έχουμε.

Σε ένα σύστημα φιλαράκων, είναι εύκολο να καθορίσουμε εάν ο «φιλαράκος» είναι ελεύθερος ή όχι, επειδή μπορούμε να διαλέξουμε την διεύθυνσή του, ανάλογα με την διεύθυνση του block που είναι να ελευθερωθεί.



## ***Pools μνήμης (memory pools)***

Ένα κατασκεύασμα που μεταφέρεται από το παλαιό exec είναι το *page pool*. Πρόσφατα, οι διαχειριζόμενες από κώδικα *page pools* έχουν ξαναγραφεί εντελώς, χρησιμοποιώντας έναν αλγόριθμο βασισμένο στις τιμές ορίου και στους διαχειρισμένους κατά μέγεθος καταλόγους μνήμης. Το κέρδος ταχύτητας ήταν τεράστιο: ακόμη και για τη «dump» περίπτωση του να δεσμευτούν 100000 blocks και να ξαναελευθερωθούν, η ταχύτητα είναι δέκα φορές γρηγορότερη από πριν. Λόγω διαχωριζόμενων κατά μέγεθος ελεύθερων καταλόγων και της εύκολης συγχώνευσης λόγω των τιμών ορίου, το πραγματικό κέρδος απόδοσης είναι ακόμα υψηλότερο και θα μπορούσε να γίνει μεγαλύτερο κατά 10 και 20 φορές.

## ***Page cache***

Μια νέα δομή δεδομένων, το δέντρο βάσης, έχει αντικαταστήσει τους hash – πίνακες που χρησιμοποιούνταν πριν. Αποδείχτηκε, ότι πολύς από τον χρόνο που ξοδεύτηκε στη διάθεση της μνήμης, ξοδεύτηκε ψάχνοντας τις κατάλληλες σελίδες στη μνήμη. Ένα σύστημα μνήμης 256 MB έχει 65536 4KB σελίδες. Αυτές οι σελίδες πρέπει να αναζητούνται κάθε τόσο. Χρησιμοποιήθηκαν αρχικά, hash - πίνακες, αλλά αποδείχθηκε ότι η διανομή 65536 καταχωρήσεων σελίδων σε μερικούς hash κάδους παρήγαγε καταλόγους αρκετών χιλιάδων σελίδων που έπρεπε να διαβούν για να βρεθεί μια σελίδα. Έτσι η ιδέα ήταν να αντικατασταθεί ο hash – πίνακας με μια άλλη δομή δεδομένων, το δέντρο βάσης. Αυτά τα δέντρα είναι μάλλον ευρέα, αλλά ρηχά, κάνοντας το traversal πολύ γρήγορο. Στις συνηθισμένες περιπτώσεις, το δέντρο δεν αυξάνεται περισσότερο από 4 έως 5 επίπεδα σε βάθος, κάτι που κάνει την έρευνα μιας σελίδας θέμα μεγίστου 4 έως 5 συγκρίσεων διαδικασιών... ενώ με hash – πίνακα πήρε περισσότερο από 1000 συγκρίσεις για να βρεθεί η σωστή σελίδα. Το αποτέλεσμα ήταν ότι η κατανομή μνήμης γενικά έχει επιταχυνθεί κατά περισσότερο από 30%!

## ***Pager***

Ένα νέο χαρακτηριστικό γνώρισμα που δεν υπήρξε σε παλαιότερες εκδόσεις του AmigaOS είναι η δυνατότητα να ανταλλαχθούν μέρη της μνήμης στο δίσκο, για να ελευθερωθεί περισσότερη μνήμη για άλλες εφαρμογές. Αυτό το χαρακτηριστικό γνώρισμα επιτρέπει τις εφαρμογές να χρησιμοποιούν περισσότερη μνήμη από αυτήν που έχει εγκατασταθεί αρχικά στο σύστημα.

Εν γένει, οι νέες βελτιστοποιημένες δομές δεδομένων επιτρέπουν στο σύστημα μνήμης για να λειτουργήσουν σε μια πολύ υψηλή ταχύτητα.

Ο χρόνος για μια κατανομή μνήμης είναι τώρα της τάξεως μερικών μικροδευτερολέπτων. Αυτό ισχύει ιδιαίτερα για τη μικρή κατανομή (κάτω από 8 KB). Για να σας δώσει μια ιδέα: Όταν το σύστημα φορτώνει στο *workbench*, έχουν πραγματοποιηθεί ήδη 40000 κατανομές στη σφαιρική *memory pool* κάτω από 2 KB.



## Επίκεντρο χαρακτηριστικών γνωρισμάτων: Ο δίσκος RAM

Ένα από τα μοναδικά χαρακτηριστικά γνωρίσματα του *workbench* του *AmigaOS* είναι ο δίσκος RAM. Είναι μια συσκευή όπως οποιοδήποτε άλλο δίσκο, και εμφανίζεται ως εικόνα τσιπ RAM στην οθόνη του *workbench* σας. Ενεργεί και αντιδρά ακριβώς όπως ένας «κανονικός» δίσκος, με μια αξιοσημείωτη εξαίρεση: τα στοιχεία που σώζονται εδώ, χάνονται όταν το μηχάνημα κλείνει ή κάνει reset.

Γιατί μπορεί αυτό να είναι χρήσιμο;

Το σύστημα χρησιμοποιεί το δίσκο RAM για το λογικό T : μία τιμή, μια θέση όπου όλα τα προσωρινά αρχεία αποθηκεύονται, και για τα CLIPS: , η λογική τιμή που περιέχει το clipboard.

Ο δίσκος RAM είναι μια καλή θέση για να αποθηκευτούν όλα εκείνα τα προσωρινά αρχεία μέσα. Παραδείγματος χάριν, μπορεί να θέλετε να εξαγάγετε κάποιο κείμενο από τον επεξεργαστή λέξεών σας σε ένα PDF για να δείτε πώς φαίνεται; μπορείτε να το σώσετε στο σκληρό σας δίσκο, οπότε σ'αυτή την περίπτωση πρέπει να το διαγράψετε αργότερα, ή μπορείτε να το σώσετε στο δίσκο RAM και να το ξεχάσετε για το παρών - δεν θα λαμβάνει πλέον τον πολύτιμο χώρο στο δίσκο μετά την επανεκκίνηση της μηχανής.

Ο δίσκος RAM πάντα μόνο θα καταναλώσει το ποσό RAM που χρειάζεται πραγματικά - καμία μνήμη δεν σπαταλιέται (που είναι και ο λόγος, του γιατί ο δίσκος RAM εμφανίζεται πάντα να είναι 100% - επαναταξινομείται όποτε χρειάζεστε περισσότερο διάστημα), και το μέγεθος περιορίζεται μόνο από το ποσό διαθέσιμης εικονικής μνήμης.

Ο δίσκος RAM είναι επίσης μια καλή θέση για να ανοίξουν τα αρχεία για την εγκατάσταση. Εάν θέλετε να εγκαταστήσετε ένα πρόγραμμα, μπορείτε εύκολα να ανοίξετε το αρχείο στο RAM, εγκαθιστάτε το πρόγραμμα, και κάνετε επανεκκίνηση. Δεν θα υπάρξουν οποιαδήποτε εναπομείναντας αρχεία να «σωριάζουν» το harddrive σας; κανένας TEMP φάκελος στο harddrive σας δεν θα συλλέξει τα εναπομείναντας αρχεία που μπορούν εύκολα να προσθέσουν μέχρι και εκατοντάδες MB του σπαταλημένου διαστήματος αποθήκευσης με την πάροδο του χρόνου.

Εκτός από το δίσκο RAM, υπάρχει και ο δίσκος RAD, ο οποίος εξυπηρετεί έναν παρόμοιο σκοπό. Εντούτοις, αντίθετα από το RAM η περιεκτικότητα σε δίσκους RAD δεν χάνεται όταν κάνετε επανεκκίνηση στη μηχανή σας! Θα βρίσκεται εκεί και μετά το reboot, και μπορεί ακόμα να χρησιμοποιηθεί κανονικά. Θα εξαφανιστεί μόνο όταν κλείσει το μηχάνημα.



## *Γλωσσάριο*

**Clipboard:** Αυτό είναι η θέση όπου αποθηκεύονται στοιχεία όπως εικόνες ή το κείμενο που αντιγράφετε για να γίνει επικόλληση αυτού αργότερα. Στο AmigaOS αυτό αντιπροσωπεύεται από τους CLIPS: λογική συσκευή, η οποία αποθηκεύεται στο δίσκο RAM.

**Logical Volume:** Αυτό είναι μια θέση αποθήκευσης που το λειτουργικό σύστημα μεταχειρίζεται ως χωριστή συσκευή αποθήκευσης, είτε είναι μια φυσική συσκευή αποθήκευσης όπως ένας σκληρός δίσκος, μια θέση σε μια τέτοια συσκευή αποθήκευσης, ή ένα εικονικό drive, όπως ο δίσκος RAM. Το όνομά του είναι η ετικέτα από την οποία το OS μπορεί να εξετάσει τις πληροφορίες που αποθηκεύονται σε εκείνο το διάστημα, οπουδήποτε μπορεί να βρίσκεται.

**Δίσκος RAD:** αυτό είναι μια έκδοση του δίσκου RAM που συντηρείται από το «ξέπλυμα» όταν κάνει reset ο υπολογιστής. Θα εξαφανιστεί όταν κλείσει ο υπολογιστής κάτω από ακριβώς κάνει κι ένας κανονικός δίσκος RAM.

## ΠΕΡΙΓΡΑΦΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Το παρακάτω πρόγραμμα γράφτηκε με την γλώσσα AmiDevCpp, η οποία δουλεύει σε περιβάλλον της C. Είναι γλώσσα προγραμματισμού για τον εξομοιωτή του Amiga, WinUAE. Είχαμε κάποια προβλήματα, γιατί έλειπε το Kickstart Rom για την εκκίνηση του εξομοιωτή, αλλά το καλό είναι ότι η AmiDevCpp τρέχει και σε περιβάλλον των Windows, βγάζοντας εκτελέσιμα αρχεία, τα οποία εκτελούνται και σε περιβάλλον Amiga ή κάποιου προσομοιωτή του. Γράφτηκε το πρόγραμμα και μετά καταφέραμε να βρούμε το Kickstart Rom καθώς και έναν workbench για τον WinUAE και στην συνέχεια τρέξαμε το exe σε αυτόν τον προσομοιωτή.

```
#include <stdio.h>
int main()
{
char c[80]; < -----δήλωση ενός χαρακτήρα σε μορφή πίνακα με μέγεθος 80 χαρακτήρων
int p,x; ----- >
printf("Please enter your name:\n"); <---Μήνυμα για πληκτρολόγηση ονόματος -->
scanf("%c",c[x]); < ----Πληκτρολόγηση ονόματος από τον χρήστη ---- >
printf("Hello %c \n",c); < ---Εκτύπωση ονόματος στην οθόνη -->
return(0);
}
```

### Παραλλαγή του προγράμματος

Εδώ, προστίθεται το στοιχείο ενός κωδικού. Ο κωδικός ορίζεται ως amiga. Ο χρήστης πληκτρολογεί το όνομά του – όπως του ζητείται από το πρόγραμμα. Στην συνέχεια του ζητείται ο κωδικός. Εάν το βρει με την πρώτη, το πρόγραμμα τον καλοσορίζει. Διαφορετικά, έχει άλλες 4 φορές να προσπαθήσει (μεταβλητή x). Αν αποτύχει, τότε το πρόγραμμα τερματίζει με ένα μήνυμα ERROR.....

```
#include <stdio.h>
int main()
{

int z;
int x;
int passwr,d;
char s[20];
passwr= amiga;
printf("Write your name: \n");
scanf("%s",s);
for(x=0;x<4;x++)
{
printf("Write the password now: \n");
scanf("%d",d);
if(d == passwr) {
x=4;
}
}
```

```
else {  
    printf("ERROR ERROR.Type the correct password or LEAVE!!!  
\n");  
    If ( x == 3) return(0);  
    }  
    }  
    printf("Welcome home %s \n" ,s);  
    }
```

Για την AmiDevCpp, τον WinUAE και το Kickstart Rom κατατρέξαμε αντίστοιχα στα παρακάτω sites:

[http://amidevcpp.amiga-world.de/amidevcpp\\_v09.php](http://amidevcpp.amiga-world.de/amidevcpp_v09.php)

<http://www.winuae.net/frames/download.html>

<http://www.fullcls.com/torrent-others-45949.html>

**Ευχαριστούμε:**

- **Μητρέγκας Γιώργος (Α.Μ.: 3560 Έτος : Πτυχίο)**
- **Μπουδαλάκης Παντελής (Α.Μ.: 3825 Έτος : Πτυχίο)**