

5^ο μέρος

Μεταφραστές
Λογισμικό Συστήματος
Μοντέλα Δεδομένων

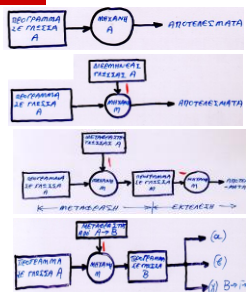
Translators and System Software

Η ΕΚΤΕΛΕΣΗ ΑΛΓΟΡΙΘΜΩΝ

- Επεξεργαστές Γλωσσών
 - Διερμηνευση - Interpretation
 - Μετάφραση - Compilation - Μεταγλώττιση
- ΛΟΓΙΣΜΙΚΟ ΣΥΣΤΗΜΑΤΟΣ SOFTWARE
 - Αναγκαιότητα Λογισμικού Συστήματος
 - Λ.Σ. = { Λειτουργικό Σύστημα + Μεταφραστές Γλωσσών } U { Εκδότες + Φορτωτές + Λογικό Επικοινωνίας }

ΤΡΟΠΟΙ ΕΠΕΞΕΡΓΑΣΙΑΣ
ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

- 1) ΑΠ' ΕΥΘΕΙΑΣ ΕΚΤΕΛΕΣΗ:
- 2) ΔΙΕΡΜΗΝΕΥΣΗ:
- 3) ΜΕΤΑΦΡΑΣΗ/
ΜΕΤΑΓΛΩΤΤΙΣΗ
- 4) ΓΕΝΙΚΗ ΠΕΡΙΠΤΩΣΗ:



ΠΡΟΒΛΗΜΑ

- Εάν $C_{L \rightarrow M}^{wM}$ ορίζει ένα μεταγλωττιστή που είναι γραμμένος σε γλώσσα μηχανής M (wM) και μεταγλωττίζει προγράμματα από την γλώσσα L στη γλώσσα M ($L \rightarrow M$), να δείξετε με ποια διαδικασία από τους μεταγλωττιστές

$$C_{L \rightarrow A}^{wA} \text{ και } C_{L \rightarrow B}^{wL}$$

και διαθέσιμο μόνο τον υπολογιστή με γλώσσα μηχανής A , πως μπορεί να παραχθεί ο μεταγλωττιστής $C_{L \rightarrow B}^{wB}$ για ένα νέο υπολογιστή με γλώσσα μηχανής B ; Ποια είναι η πρακτική αξία αυτής της διαδικασίας;

Κλάσεις Γλωσσών

- Κλάση 0: Γλώσσες χωρίς περιορισμούς (unrestricted)
 - Αποδεκτές από Μηχανές Turing
 - Κανόνες: $uAv \rightarrow uwv$, $A = \text{μη-τερματικό}$,
- Κλάση 1: Γλώσσες ευαίσθητες συμφραζομένων (context sensitive)
 - Αποδεκτές από Μηχανές Turing με γραμμικά περιορισμένη ταινία (η κεφαλή δεν μπορεί να κινηθεί δεξιάτερα)
 - Κανόνες: $uAv \rightarrow uwv$ με $A = \text{μη-τερματικό}$ και $w \in \epsilon$ (ή $|uAv| \leq |uwv|$)
- Κλάση 2: Γλώσσες ελεύθερες συμφραζομένων (context free)
 - Αποδεκτές από αυτόματα στοιβας
 - Κανόνες: $A \rightarrow u$, $A = \text{μη-τερματικό}$,
 - Εδώ είναι οι Γλώσσες Προγραμματισμού
- Κλάση 3: Γλώσσες Κανονικές (regular) (π.χ., $L((a \cup b)^* a)$)
 - Αποδεκτές από πεπερασμένα αυτόματα

Ορισμός Συντακτικού
Γλωσσών προγραμματισμού

- BNF συμβολισμός
 - τερματικά σύμβολα
 - μη τερματικά σύμβολα ή συντακτικές κατηγορίες
 - παραγωγές (τρόπος σύνταξης συντακτικών κατηγοριών, →
- π.χ. καταχωρήσεις τηλεφωνικού καταλόγου


```
entry → person_name address number
person_name → surname forename
surname → name
forename → proper_forename | initial
proper_forename → name
initial → letter .
address → number street_name
street_name → name
name → letter { @ name }
number → digit { @ digit }
letter → A | B | C | ... | Z
digit → 0 | 1 | 2 | ... | 9
```

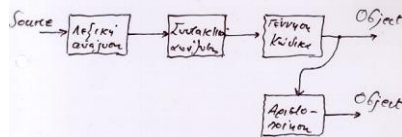
{ @ = επανάληψη }
| = OR

Ορισμός Συντακτικού Γλωσσών Προγραμματισμού

π.χ. Απλή γλώσσα προγραμματισμού

- program → statement {statement}
 - statement → conditional | loop | assignment
 - conditional → if condition then statement
 - loop → while condition do statement
 - assignment → set name to expression
 - expression → name operator name
 - operator → + | -
 - condition → name relation number
 - relation → = | ≠
 - name → letter{@ name}
 - number → digit {@ digit}
 - letter → A | B | C | ... | Z
 - digit → 0 | 1 | 2 | ... | 9
- {@ = επανάληψη }
| = OR

Μεταφραστές Compilers – Μεταγλωττιστές

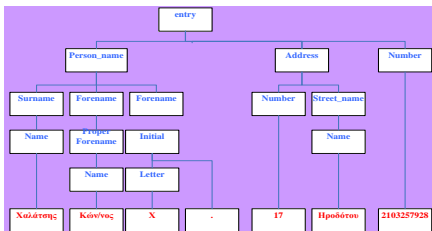


Λεξική ανάλυση tokens -κουπόνια
 if $x \neq 1$ then set Y to $X + Y$
types if name ≠ number then set name to name+name
values (X) (1) (X) (X) (Y)
 token = (type, value)

Συντακτική ανάλυση (parsing)

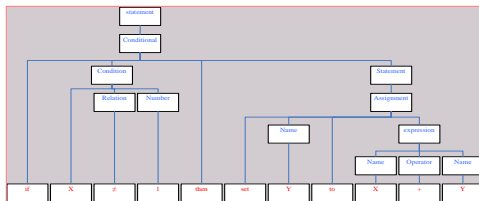
□ συντακτικό δένδρο (parse tree)

π.χ. 1 : καταχώρηση τηλεφωνικού καταλόγου
 Χαλάτσος Κων/νος Χ. 17 Ηρώδτου 2103257928



Συντακτική ανάλυση (parsing)

π.χ. 2 : if $x \neq 1$ then set Y to $X + Y$



Μέθοδες Συντακτικής ανάλυσης

- Από-κάτω-προς- τα-πάνω (bottom-up)
- Από-πάνω-προς-τα-κάτω (top-down)
 - > Πλεονεκτεί διότι επιτρέπει την αυτόματη παραγωγή συντακτικού αναλυτή
- Ανάγκη οπισθοδρόμησης (backtracking)
- top-down συντακτικός αναλυτής
 - module parse-statement
 - if the type of the next token is "if" then parse-conditional
 - else if the type of the next token is "while" then parse-loop
 - else if the type of the next token is "set" then parse-assignment
 - else report failure

Συντακτική ανάλυση assignment

- statement → conditional/loop/assignment
- module parse-assignment
 - check that the type of the next token is "set"
 - check that the type of the next token is "name"
 - check that the type of the next token is "to"
 - parse-expression

Γέννηση κώδικα

- **Αφορά:**
 - > εκχώρηση μνήμης στα δεδομένα
 - > γέννηση κώδικα – εντολές μηχανής
- Η εκχώρηση μνήμης στα δεδομένα
 - > γίνεται με τη βοήθεια του Πίνακα Συμβόλων (**SYMBOL TABLE**)
 - > είναι συνάρτηση του ΤΥΠΟΥ των δεδομένων
 - > ο ΤΥΠΟΣ είτε δηλώνεται ή υπονοείται
- Η γέννηση κώδικα
 - > είναι συνάρτηση της συντακτικής δομής
- Π.χ., `module generate-statement (T) {T=parse Tree} if statement conditional then generate-conditional (subtree of T) else if statement loop then generate-loop (subtree of T) else generate-assignment (subtree of T)`

Παράδειγμα γέννησης κώδικα

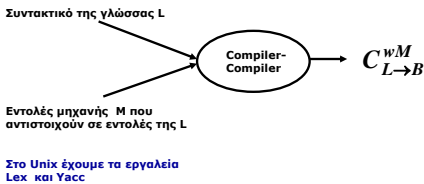
- Έστω η εντολή: `if X ≠ 1 then set Y to X + Y`
- Ο πίνακας συμβόλων

όνομα	ΤΥΠΟΣ	address
X	integer	1456
Y	integer	1457
1	integer	100
- Οι κλήσεις και οι αντίστοιχες εντολές μηχανής που γεννιούνται:

Address	Instruction
1000	LOAD 1456
1001	SUB 100
1002	JUMPZ 1006
1003	LOAD 1456
1004	ADD 1457
1005	STORE 1457
1006	

Εργαλεία δημιουργίας/κατασκευής compilers

Compiler-Compiler



Συμβολομεταφραστής – Assembler

- Μετατρέπει τη Συμβολική Γλώσσα Μηχανής (Assembly) σε γλώσσα μηχανής (binary)


```
Assembly → Assembler → Binary
```
- Η γλώσσα Assembly περιέχει:
 - > Συμβολική γραφή των εντολών μηχανής
 - > Ψευδοεντολές (οδηγίες συμβολομετάφρασης)
 - > Δηλώσεις δεδομένων (τύπος, τιμή)
 - > Δήλωση MACRO εντολών
 - > Διαφορά μεταξύ Υπορουτίνας και MACRO

Φορτωτές - Loaders

- **Απόλυτος Δυαδικός Φορτωτής (Absolute binary loader)**
 - > Φορτώνει στη μνήμη ένα πρόγραμμα σε απόλυτη δυαδική μορφή (όπως αυτό έχει γεννηθεί)
- **Σχετικός Φορτωτής (Relocating Loader)**
 - > Μεταθετό πρόγραμμα (relocatable program)
 - > Έχει παραχθεί με αρχή τη μηδέν διεύθυνση μνήμης
 - > Οι διευθύνσεις αλλάζουν με βάση την διεύθυνση αρχής από την οποία αρχίζει να φορτώνεται το πρόγραμμα στη μνήμη

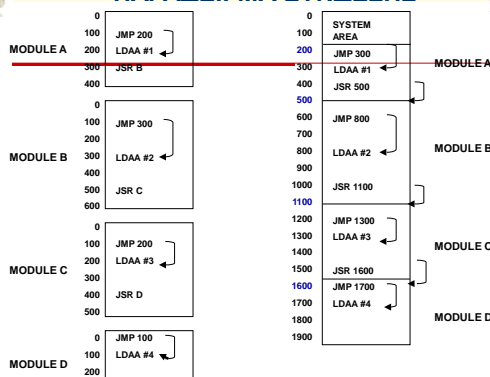
ΜΕΤΑΘΕΣΗ - RELOCATION

ORIGIN 0		ORIGIN \$2000
	PIAD EQU \$4000	
0000 B6 1000	LOAD LDAA ALPHA	2000 B6 3000
0003 B7 1001	STAA BETA	2003 B7 3001
0006 7E 0100	JMP READ	2006 7E 3100
...
0100 F6 4000	READ LDAB PIAD	2100 F6 4000
0103 7E 0000	JMP LOAD	2103 7E 2000
...
1000 00	ALPHA RMB 1	3000 00
1001 00	BETA RMB 2	3001 00

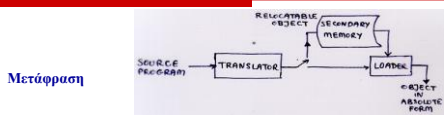
Σύνδεση - Linking

- Επιτρέπει τη σύνδεση στοιχείων προγράμματος σε ένα μόνο πρόγραμμα
- Η σύνδεση μπορεί να λάβει χώρα σε επτά διαφορετικές χρονικές στιγμές:
 - > Κατά την ώρα συγγραφής του πηγαίου προγράμματος
 - > Αυτό σημαίνει σύνδεση με το χέρι
 - > Μετά την κωδικοποίηση αλλά πριν την μετάφραση
 - > Υπονοεί ξανα-κωδικοποίηση και σύνδεση με το χέρι
 - > Κατά την μετάφραση
 - > Υπονοεί ξανα-μετάφραση
 - > Μετά την μετάφραση αλλά πριν την φόρτωση
 - > Συνεπάγεται ένα Συνδότη (Linker)
 - > κατά την ώρα φόρτωσης
 - > Συνεπάγεται ένα Συνδότη-Φορτωτή (Linking Loader)
 - > Μετά την φόρτωση αλλά πριν την εκτέλεση
 - > Απαιτεί να έχουμε στη μνήμη ένα μεγάλο αριθμό στοιχείων
 - > Κατά την ώρα εκτέλεσης
 - > Συνεπάγεται Δυναμική Σύνδεση (Dynamic Linking)

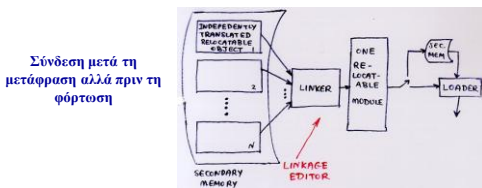
ΠΑΡΑΔΕΙΓΜΑ ΣΥΝΔΕΣΗΣ



Συνήθη σχήματα Μετάφρασης και Σύνδεσης

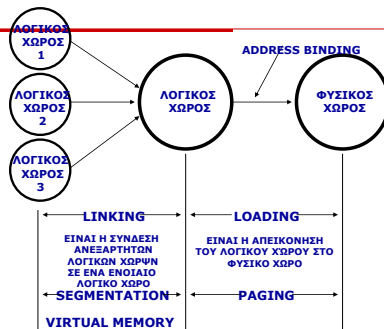


Μετάφραση



Σύνδεση μετά τη μετάφραση αλλά πριν τη φόρτωση

Μια άλλη άποψη Σύνδεσης και Φόρτωσης



Λειτουργικό Σύστημα (Operating System)

Ορισμός Ένα σύνολο διεργασιών που καθοδηγούν και ελέγχουν τη λειτουργία και τη χρήση ενός υπολογιστικού συστήματος

Αντικειμενικός Σκοπός Η μεγιστοποίηση της απόδοσης ενός συστήματος με ταυτόχρονη δημιουργία φιλικού περιβάλλοντος επεξεργασίας των προγραμμάτων των χρηστών

Βασικές Έννοιες Διεργασία (Process)
 Κατάσταση διεργασίας
 Εργασία = σύνολο διεργασιών
 Λειτουργικές καταστάσεις μηχανής
 Διακοπές
 Προνομιούχες Εντολές
 Προτεραιότητα
 Προστασία

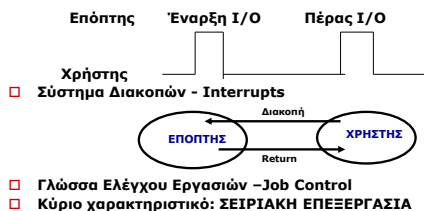
Λογισμικό Συστήματος 1ης γενιάς (1949-1956)

- Αυτοφορτωτής (Bootstrap loader)
- Απόλυτος φορτωτής (Absolute loader)
- Assembly - Assembler / Συμβολομεταφραστής
- OS Μαζικής Επεξεργασίας (Batch processing)
- Γλώσσες Υψηλού Επιπέδου - Μεταγλωττιστές (Compilers)
- Μεταθετός Κώδικας (relocatable code)
- Σχετικός φορτωτής (relocating loader)
- Συνδότης (linker)
- Συνδότης Φορτωτής (linking loader)

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Λογισμικό Συστήματος 2ης γενιάς (1956-1962)

- Αυτόνομο σύστημα I/O
- Αρχή υποκλοπής κύκλων – cycle stealing
- Κατάσταση ΕΠΟΠΤΗ (SVR) – κατάσταση Χρήστη (User)



Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 25

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Λογισμικό Συστήματος 3ης γενιάς (1962-1970)

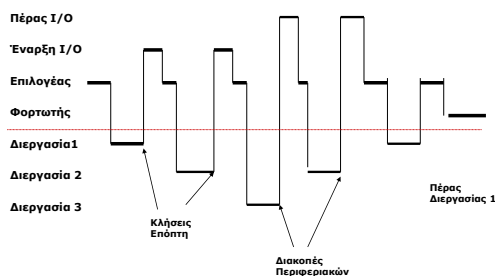
- ΠΟΛΥΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ
 - Σελιδοποίηση μνήμης – Paging
 - Εικονική Μνήμη – Virtual Memory
 - Αποσύνδεση I/O – off-line spooling
 - Διεργασία – Process
- Λειτουργικό Σύστημα – Operating System



Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 26

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Πολυπρογραμματισμός



Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 27

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Υλοποίηση Πολυπρογραμματισμού

- Εναλλαγή της CPU μεταξύ διεργασιών
- Χειρισμός διακοπών / interrupt handling
- Εκχώρηση πόρων / resource allocation: (μνήμη, I/O...)
- Προστασία μεταξύ των διεργασιών
- Προστασία πόρων / resource protection
- Χρονοπρογραμματισμός μεταξύ διεργασιών

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 28

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Είδη Λειτουργικών Συστημάτων Σήμερα

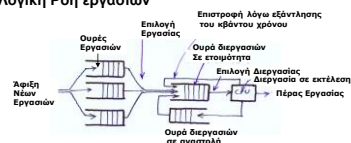
- Μαζικής Επεξεργασίας (Batch processing)
- Καταμερισμού Χρόνου (Time Sharing)
 - Συστήματα Πελάτη-Υπηρετή (Client-Server)
- Κατανεμημένης Επεξεργασίας (distributed processing)
 - Π.χ., Υπολογιστική Πλέγματος (Grid computing)
 - Ευφυής Σκόνη
- Παράλληλης Επεξεργασίας (parallel computing)
 - Συμμετρική πολυεπεξεργασία (symmetric multiprocessing)
 - Πολυεπεξεργασία με Πέρασμα Μηνυμάτων (Message passing)
- Πραγματικού Χρόνου, (real-time computing) για
 - Έλεγχο Φυσικών Διεργασιών
 - Κρίσιμος χρόνος, Ανάδραση, Ασφαλής Αποτυχία
 - Συστήματα Πληροφορικής, π.χ., Μηχανές Αναζήτησης
 - Συστήματα Συναλλαγών, π.χ., Τραπεζικές συναλλαγές
 - Κρατήσεις Θέσεων

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 29

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Εκτέλεση εργασιών

- Εργασία: Σύνολο διεργασιών
- Τυπικά βήματα εκτέλεσεως μιας εργασίας
 1. Εισαγωγή πηγαίου προγράμματος
 2. Μετάφραση αυτού του και παραγωγή αντικείμενου κώδικα
 3. Φόρτωση αντικείμενου προγράμματος στη μνήμη
 4. Εισαγωγή δεδομένων εισόδου
 5. Εκτέλεση προγράμματος
 6. Εξαγωγή αποτελεσμάτων
- Λογική Ροή εργασιών



Έλεγχος Εργασιών – Job Control

- > Γλώσσα ελέγχου εργασιών (απλή μορφή)
 - > Με αυτή ο χρήστης δίνει στο Λ.Σ.
 - > την αναγνώριση / ταυτότητα του: username - password
 - > λειτουργική πληροφορία (τι θέλει, π.χ., load, compile,...)
 - > I/O πληροφορία (I/O συσκευές, I/O files)
 - > πληροφορία εκτιμώμενων αναγκαιούτων πόρων για (μνήμη, χρόνο εκτέλεσης κλπ)
 - > το επείγον (προτεραιότητα) της εργασίας
- > Αυτά τα χρειάζεται ο χρονοπρογραμματιστής εργασιών για την επιλογή της εργασίας στο υψηλό επίπεδο

Χρονοπρογραμματισμός & εκχώρηση πόρων

- > Χρονοπρογραμματισμός εργασιών:
 - > Επιλογή λαμβάνοντας υπόψη κυρίως:
 - απαιτούμενους πόρους (μνήμη, χρήση I/O)
 - διαθέσιμους πόρους
 - προτεραιότητα
 - χρονική διάρκεια αναμονής
- > Είδος επιλογής:
 - > Στατική : όλοι οι πόροι πρέπει να ικανοποιούνται
 - > Δυναμική: οι πόροι ικανοποιούνται σταδιακά, ίσως οδηγήσει σε **αδιέξοδο / deadlock**

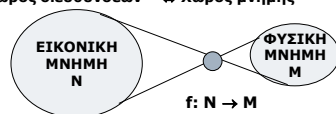
Καταστάσεις μιας Διεργασίας

- > Τρεις κύριες καταστάσεις:
 - > Σε ετοιμότητα (ready)
 - > Τρέχουσα/ Εκτελούμενη (running)
 - > Σε αναστολή – Μπλοκαρισμένη (blocked)
- > Μεταπτώσεις κατάστασης



Διαχείριση Κυρίας Μνήμης (memory management)

- Στόχοι:
 - > Μεταθετότητα (relocation)
 - > Προστασία (protection)
 - > Λογική Οργάνωση (logical organization)
 - > Φυσική Οργάνωση (physical Organization)
- Εικονική Μνήμη (Virtual Memory)
 - > Διευθύνσεις προγράμματος ↔ θέσεις μνήμης
 - > Χώρος διευθύνσεων ↔ Χώρος μνήμης



Οργάνωση Μνήμης

Εικονική Μνήμη

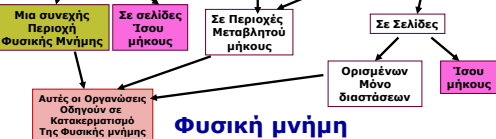
Οργανώνεται

- Γραμμικά
- Με περιγραφείς
- Σε τμήματα

Εκχωρείται

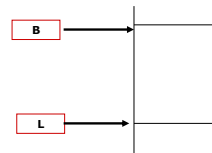
εκχωρείται

εκχωρείται



Διαχείριση Μνήμης Βάση-Όριο (Datum-Limit)

- Στη διεργασία δίνεται μια συνεχής περιοχή Φυσικής Μνήμης
- Η περιοχή αυτή ορίζεται με δύο καταχωρητές:
 - > Βάση B (basis)
 - > Όριο L (limit)



- Μια λογική διεύθυνση a αντιστοιχεί στη φυσική διεύθυνση $f(a) = [B] + a$
- Προστασία μνήμης: $[B] \leq f(a) = [B] + a \leq [L]$

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Παράδειγμα Βάσης-Οριού

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 37

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Κατακερματισμός μνήμης

Ανάγκη μετακίνησης

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 38

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Σελιδοποίηση - Paging (σελίδες ίδιου μεγέθους)

- Ο λογικός χώρος της διεργασίας χωρίζεται σε σελίδες
- Αντίστοιχα, ο φυσικός χώρος χωρίζεται σε πλαίσια σελίδων ίδιου μεγέθους
- Η σελίδα P της διεργασίας αποθηκεύεται σε αντίστοιχο κενό πλαίσιο σελίδας P' της φυσικής μνήμης
- Μια λογική διεύθυνση χωρίζεται στα δύο:

p	w
---	---

 - P = αριθμός σελίδας διεργασίας
 - W = αριθμός θέσης στη σελίδα P
- Ένας μηχανισμός αντιστοίχιας μετατρέπει μια λογική διεύθυνση της διεργασίας στην αντίστοιχη φυσική

P'	w
----	---
- Μηχανισμός μετατροπής/αντιστοίχιας
 - Πίνακας σελίδων (page table)
 - Συνευριστική Μνήμη (associative memory)

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 39

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Σελιδοποίηση - Paging

- Πλεονεκτήματα σελιδοποίησης**
 - Δεν οδηγεί σε κατακερματισμό της μνήμης
 - Δεν είναι απαραίτητο να είναι ταυτόχρονα στη μνήμη όλες οι σελίδες της διεργασίας (Οι άλλες είναι στο δίσκο)
 - Άρα, μπορεί ο λογικός χώρος μιας διεργασίας να είναι μεγαλύτερος του φυσικού χώρου
- Μειονεκτήματα**
 - Αυξημένο κόστος Υλικού του επεξεργαστή
 - Αυξημένη πολυπλοκότητα του λειτουργικού συστήματος
 - Ανάγκη διαχείρισης «λαθών σελίδας»

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 40

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Υλοποίηση Σελιδοποίησης Με πίνακα σελίδων (page table)

Μέγεθος σελίδας = 2^n

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 41

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Υλοποίηση Σελιδοποίησης Με συνευριστική μνήμη (associative memory)

> Στη συνευριστική μνήμη φυλάσσεται η αντιστοίχια λογικής και φυσικής σελίδας για ένα μικρό αριθμό σελίδων
 > Απαιτείται ενημέρωση της συνευριστικής μνήμης ώστε να μειωθεί ο ρυθμός αποτυχίας αντιστοίχιας.

- HT/MISS RATE
- ΑΝΤΙΚΑΤΑΣΤΑΣΗ: ΚΥΚΛΙΚΗ
- K σελίδες 8, 16
- οι υπολοίποι θέσεις του πίνακα είναι στην μνήμη
- MRR: Most Recently Referenced

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 42

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Αντικατάσταση σελίδων

page swap

- Προκύπτει η ανάγκη αντικατάστασης σελίδας όταν μια σελίδα που θέλει η διεργασία δεν είναι στη μνήμη και η μνήμη δεν έχει κενό πλαίσιο σελίδας για να την φέρει από τη δευτερεύουσα μνήμη
- Στη περίπτωση αυτή λέμε ότι έχουμε ένα **λάθος σελίδας (page fault)** που δημιουργεί μια **διακοπή** στη συνέχιση της εκτέλεσης της διεργασίας
- Το σύστημα διαχείρισης του Λ.Σ. Αναλαμβάνει να κάνει τη σχετική αντικατάσταση
- Πρόβλημα: ποια σελίδα θα αντικατασταθεί;
- Στόχοι της πολιτικής αντικατάστασης σελίδων:
 - Αντικατάσταση της σελίδας που δεν θα χρειαστεί για τον περισσότερο χρόνο στο μέλλον
 - Πρόβλεψη της μελλοντικής συμπεριφοράς από τη συμπεριφορά του παρελθόντος

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 43

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Πολιτικές αντικατάστασης σελίδων

- Τυχαία
 - Με γεννήτρια τυχαίων αριθμών
- Λιγότερο πρόσφατα χρησιμοποιημένη (LRU, least recently used)
 - Με καταχωρητή γήρανσης της σελίδας
- Λιγότερο συχνά χρησιμοποιημένη (LFU, least frequently used)
 - Με απεριθμητή χρήσης της σελίδας
- Περισσότερο χρόνο στη μνήμη, ισοδύναμα, πρώτη-μέσα-πρώτη-έξω, (longest resident, FIFO: first-in-first-out)
 - Με ουρά FIFO
- Βελτιώσεις:
 - Δεύτερη ευκαιρία (FIFO + used bit)
 - Έχει τροποποιηθεί η υπό αντικατάσταση σελίδα; Αυτό σημαίνει ότι σχετικό **written-into bit** μας δίνει αυτή την πληροφορία.

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 44

Neapolis University Pafos Introduction to Computer Science and networks

Χειρισμός ΕΙΣΟΔΟΥ/ΕΞΟΔΟΥ

(Input/Output, I/O)

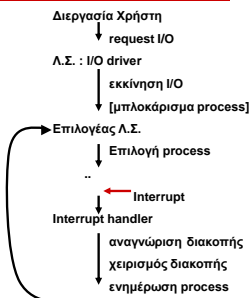
- Το λειτουργικό Σύστημα χειρίζεται όλα τα I/O μηχανήματα
- αποφεύγεται έτσι να ξέρει λεπτομέρειες ο προγραμματιστής
- Το I/O το κάνει το OS (εκκίνηση I/O, τερματισμός I/O) μέσω interrupts
- Το OS χειρίζεται διαφορετικά μοιραζόμενες ή μη (sharable & unsharable) I/O συσκευές
- Τεχνικές βελτίωσης του συστήματος I/O
 - Μνήμες απομόνωσης (buffers)
 - Κουβάρισμα - Spooling
 - Αποσύνδεση διεργασίας από μη διαμοιραζόμενες μονάδες I/O
 - Χρήση ενδιάμεσου μέσου (συνήθως Δίσκος)

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 45

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Χειρισμός ΕΙΣΟΔΟΥ/ΕΞΟΔΟΥ

(Input/Output, I/O)



Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 46

Neapolis University Pafos Introduction to Computer Science and networks

Διαχείριση αρχείων

(file management)

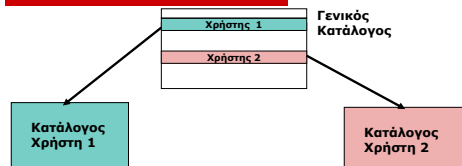
- Αφορά:
 - δημιουργία & σβήσιμο αρχείων
 - πρόσβαση αυτών (για διάβασμα, ενημέρωση, σβήσιμο)
 - διαχείριση δευτερεύουσας μνήμης
 - προστασία αρχείων από μη εξουσιοδοτημένη πρόσβαση
 - προστασία αρχείων από απώλεια ή διαφθορά λόγω H/W και/ή S/W βλαβών ή λαθών
 - Διαμοιρασμός αρχείων μεταξύ χρηστών
 - Αλλά και προστασία από μη εξουσιοδοτημένη πρόσβαση
- Κατάλογος / directory αρχείων
 - master directory - user directories
 - multi-level directory δομή
- Οργάνωση αρχείων: Λογική οργάνωση και Φυσική οργάνωση
- Back-up αρχείων
 - περιοδική - όλο το αρχείο
 - αυθημερόν - μόνο το μέρος που αλλάζει

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 47

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Δομή Καταλόγου Αρχείων

master directory - user directories



- Για κάθε χρήση ο Γενικός κατάλογος περιέχει: το όνομα του χρήστη και τη διεύθυνση του δικού του καταλόγου
- Ο κατάλογος του χρήστη περιέχει τα εξής για κάθε δικό του αρχείο:
 - Όνομα αρχείου
 - Διεύθυνση αρχείου στη δευτερεύουσα μνήμη
 - Μέγεθος αρχείου
 - Επιτρεπτή πρόσβαση (R, W, ...)
 - Διαχειριστική πληροφορία (π.χ., ημερομηνία δημιουργίας, χρόνος τελευταίας αλλαγής)

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 48

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Πολυεπίπεδος κατάλογος αρχείων

• Δεοντοεπής δομή
• Επιβάρυνση: Παλλακτές προσπάθειες στο δίσκο για να εντοπιστεί ο κατάλληλος κατάλογος
• Λύση: χρήση «τρέχοντος» καταλόγου

Πλήρες όνομα: root/zeta/theta/alpha
Σχετικό όνομα: alpha

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 49

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Τρόποι αποθήκευσης αρχείων

- ❑ Διασύνδεση τμημάτων (block linkage)
- ❑ Χάρτης απεικόνισης αρχείου (file map)
- ❑ Δείκτης τμημάτων (index block)

Κατάλογος

Κατάλληλος τρόπος για σειριακή πρόσβαση

Κατάλογος

Κοινός χάρτης για όλη τη Δευτερεύουσα μνήμη
Επιβάρυνση: η τμηματική προσκόμιση του χάρτη στη μνήμη
Κατάλληλος για σειριακή πρόσβαση

Κατάλογος

Ένα index block ανά αρχείο
Κατάλληλος για Τυχεία Πρόσβαση

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 50

Neapolis University Pafos Introduction to Computer Science and networks

ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ

Data Processing (DP)

- Μεγάλος όγκος στοιχείων που δεν χωρά στη μνήμη
- I/O μεταξύ μνήμης & δευτερεύουσας μνήμης - αριστοποίηση
- κύριος χώρος εφαρμογής DP:
 - Εμπορικός,
 - Δημόσια Διοίκηση
 - & Επιστημονικός τύρα
- Δραστηριότητες:
 - εισαγωγή στοιχείων
 - επαλήθευση στοιχείων
 - επεξεργασία στοιχείων
 - έκδοση αποτελεσμάτων
- Ο χειρισμός τους απαιτεί ΔΟΜΗΣΗ των στοιχείων

αξιοπιστία & ασφάλεια

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 51

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ΜΟΝΤΕΛΑ ΔΕΔΟΜΕΝΩΝ

- ❑ Οντότητες / αντικείμενα (entities / objects)
- ❑ Ιδιότητες / συνιστώσες / χαρακτηριστικά (properties / attributes)
- ❑ Σύνολα οντοτήτων, κατηγορίες οντοτήτων, κλάσεις οντοτήτων (entity set, entity category, entity class): **ιδιες ιδιότητες**
- ❑ Τιμές ιδιοτήτων (values)
- ❑ Κλειδί οντότητας (key) : ιδιότητα ή συνδυασμός ιδιοτήτων με τιμή / τιμές που μονοσήμαντα ξεχωρίζει μια οντότητα από το σύνολο της
- ❑ Σχέσεις μεταξύ οντοτήτων - relationships
- ❑ Μοντέλα δεδομένων: ανεξάρτητα της εφαρμογής
 - Δομή structure
 - Χειρισμός manipulation – updates
 - Ακεραιότητες integrity constraints

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 52

Neapolis University Pafos Introduction to Computer Science and networks

Τυπικά παραδείγματα οντοτήτων

Εφαρμογή	Σύνολο οντοτήτων	Ιδιότητες	Κλειδί	Σχέση
Τράπεζα	Πελάτες	Όνομα Διεύθυνση αρ. πελάτη	αρ. πελάτη	Καποχή (Πελάτης, Λογισμός)
	Λογαριασμοί	αρ. λογαριασμού υπόλοιπο	αρ. λογαριασμού	
Κρατήσεις θέσεων	Πτήσεις	αρ. Πτήσεως αφετηρία προορισμός ώρα αναχώρησης	αρ. Πτήσης	Ανέτηση (Αεροπλάνο, πτήση)
	Αεροπλάνα	Κατασκευαστής Μοντέλο αρ. σειράς	αρ. σειράς	
Αποθήκη	Επιβάτες	Όνομα Διεύθυνση	Όνομα + Δί/ση	
	Εξαρτήματα αυτοκινήτων	αρ. εξαρτήτος Κόστος Παράσταση	αρ. εξ'τος	Προμήθεια (Κατασκευαστής, εξάρτημα)
	Κατασκευαστές	Όνομα εταιρίας Διεύθυνση Όριο Πιστώσεως	ονομ. εταιρίας	

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 53

Neapolis University Pafos Introduction to Computer Science and networks

ΤΟ ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

(Codd 1970)

ΔΟΜΗ

- οντότητα = **πλειάδα** (tuple) = {τιμές **ιδιοτήτων**}
- σχέση (relation) = σύνολο οντοτήτων {attribute values} = πίνακας: **γραμμή = πλειάδα**
στήλη = ιδιότητα

σχέση Αεροπλάνο

κατασκευαστής	Model	αρ.	Σειράς
Boeing	747	82	
Boeing	737	46	
British Aerospace	concorde		14

σχέση Πτήσεις

αρ. Πτήσης	Αφε'ρια	Προορι/μος	Ώρα αναχ.
BA014	London	Sydney	21:40
BA837	NY	London	09:30
BA152	London	Paris	10:20

σχέση Ανάθεση

αρ. Πτήσης	Αρ. σειράς
BA014	82
BA837	14
BA152	46

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 54

ΧΕΙΡΙΣΜΟΣ ΣΧΕΣΕΩΝ

□ Πράξεις σχεσιακής άλγεβρας :SQL γλώσσα

➤ Επιλογή Selection

- επιλέγει γραμμές
- αποτέλεσμα = νέα σχέση
- Π.χ., `Select Πτήσεις where Αφετηρία = London`
`Select Αεροπλάνο where Κατασκευαστής = Boeing`

□ Προβολή Projection

- επιλέγει στήλες
- αποτέλεσμα = νέα σχέση
- Π.χ., `Project Πτήσεις on Προορισμός`
`Project Αεροπλάνο on Κατασκευαστής, Model`
`Project Αεροπλάνο on Κατασκευαστής`

ΧΕΙΡΙΣΜΟΣ ΣΧΕΣΕΩΝ

➤ Συνένωση Join:

- συνδυάζει δύο σχέσεις
- Φυσική συνένωση (natural join) : μια τουλάχιστον κοινή στήλη
- Π.χ., $A \text{ natural join } B \rightarrow R$

A	<table style="border-collapse: collapse; text-align: center;"> <tr><td>X</td><td>Y</td></tr> <tr><td>X1</td><td>Y1</td></tr> <tr><td>X2</td><td>Y2</td></tr> <tr><td>X3</td><td>Y1</td></tr> <tr><td>⋮</td><td></td></tr> <tr><td></td><td></td></tr> </table>	X	Y	X1	Y1	X2	Y2	X3	Y1	⋮				B	<table style="border-collapse: collapse; text-align: center;"> <tr><td>Y</td><td>Z</td></tr> <tr><td>Y1</td><td>Z1</td></tr> <tr><td>Y1</td><td>Z2</td></tr> <tr><td>Y2</td><td>Z2</td></tr> <tr><td>Y3</td><td>Z3</td></tr> <tr><td>⋮</td><td></td></tr> <tr><td></td><td></td></tr> </table>	Y	Z	Y1	Z1	Y1	Z2	Y2	Z2	Y3	Z3	⋮				⇒	R	<table style="border-collapse: collapse; text-align: center;"> <tr><td>X</td><td>Y</td><td>Z</td></tr> <tr><td>X1</td><td>Y1</td><td>Z1</td></tr> <tr><td>X1</td><td>Y1</td><td>Z2</td></tr> <tr><td>X2</td><td>Y2</td><td>Z2</td></tr> <tr><td>X3</td><td>Y1</td><td>Z1</td></tr> <tr><td>X3</td><td>Y1</td><td>Z2</td></tr> <tr><td>⋮</td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>	X	Y	Z	X1	Y1	Z1	X1	Y1	Z2	X2	Y2	Z2	X3	Y1	Z1	X3	Y1	Z2	⋮					
X	Y																																																							
X1	Y1																																																							
X2	Y2																																																							
X3	Y1																																																							
⋮																																																								
Y	Z																																																							
Y1	Z1																																																							
Y1	Z2																																																							
Y2	Z2																																																							
Y3	Z3																																																							
⋮																																																								
X	Y	Z																																																						
X1	Y1	Z1																																																						
X1	Y1	Z2																																																						
X2	Y2	Z2																																																						
X3	Y1	Z1																																																						
X3	Y1	Z2																																																						
⋮																																																								

- Πτήσεις join Ανάθεση join Αεροπλάνο = ?
- Πτήσεις join Αεροπλάνο = ?

Παράδειγμα Join

- Πτήσεις join Ανάθεση join Αεροπλάνο = ?

join: Πτήσεις join Ανάθεση:

αρ. Πτήσης	Αφετηρία	Προορ/μός	Ωρα Αναχ.	Αρ. Σειράς
BA014	London	Sydney	21:40	82
BA837	NY	London	09:50	14
BA152	Londo	Paris	10:20	46

και στη συνέχεια: join αεροπλάνο:

κατασ/στής	model
Boeing	747
British Ar.	Concord
Boeing	737

Σχεσιακές εκφράσεις (Παραδείγματα)

- Π.χ. 1: Επέλεξε όλες τις πλειάδες (Αρ. πτήσης, κατασκευαστής, Model) με αρ. πτήσης = BA014

`Select (Project (Πτήσεις join Ανάθεση join Αεροπλάνο) on Αρ. Πτήσης, Κατασκευαστής, Model) where αρ. πτήσης = BA014`

Αποτέλεσμα: BA014 Boeing 747

- Π.χ. 2 Βρέξ όλους τους αρ. πτήσεων που αναχωρούν από το Λονδίνο

`Project (Select Πτήσεις where Αφετηρία = Λονδίνο) on Αρ. Πτήσης`

Αποτέλεσμα: BA014
BA152

Κανόνες ακεραιότητας

- Κάθε σχέση πρέπει να έχει ένα ΚΛΕΙΔΙ και κάθε πλειάδα στην σχέση πρέπει να έχει μια τιμή ΚΛΕΙΔΙΟΥ που να είναι μοναδική και όχι κενή
- Δες π.χ., σχέση Ανάθεση (Αρ. Πτήσης, Αρ. σειράς)
- Κάθε οντότητα στην οποία κάνει αναφορά μια σχέση πρέπει επίσης να εμφανίζεται στην σχέση η οποία αντιπροσωπεύει το σύνολο οντοτήτων στο οποίο η οντότητα ανήκει
- Π.χ., Κάθε αεροπλάνο που υπάρχει στη σχέση Ανάθεση πρέπει να υπάρχει και στη σχέση Αεροπλάνο
- Δηλαδή, όχι "ελεύθερα / χαλαρά άκρα"

ΕΓΓΡΑΦΕΣ, ΑΡΧΕΙΑ και ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ (records, files, databases)

Μοντέλο Δεδομένων - σχήμα ΣΔΒΔ (schema DBMS)

Δομές Δεδομένων - Λ.Σ. Οργάνωση αρχείων

Φυσικά μέσα αποθήκευσης - Λ.Σ. Drivers

- **Εγγραφή (record):** περιγράφει μια οντότητα
Είναι ένα σύνολο πεδίων (fields)
Ένα πεδίο είναι το κλειδί της εγγραφής
- **Αρχείο (file)** = σύνολο σχετικών εγγραφών
- **Βάση δεδομένων (Database, DB)** = σύνολο αρχείων
- **Σύστημα Διαχείρισης βάσης δεδομένων (Data Base Management System)**
- **Πράξεις επί των εγγραφών :** συναλλαγή (transaction)
read, write, add, delete, update

Υλοποίηση του σχεσιακού μοντέλου

- **σχέση** = αρχείο
- **πλειάδα** = εγγραφή στο επίπεδο οργάνωσης αρχείου
- **ιδιότητα** = πεδίο
- **Διάφορες τεχνικές στο επίπεδο φυσικού μέσου** (ξεφεύγει των ορίων του μαθήματος)

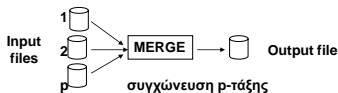
ΟΡΓΑΝΩΣΗ ΕΓΓΡΑΦΩΝ

Οργάνωση εγγραφών μέσα σ' ένα αρχείο: είναι συνάρτηση του τρόπου πρόσβασης

- **αρχείο τυχαίας πρόσβασης**
 - κατάλληλο για εφαρμογές πραγματικού χρόνου
 - κρατήσεις θέσεων
 - τράπεζες
- **Σειριακό αρχείο**: κλειδί διάταξης (order key) (όχι αναγκασία μοναδικό)
 - π.χ. αρχείο προσωπικού για μισθοδοσία
 - Βασικές πράξεις σε σειριακά αρχεία
 - Συγχώνευση
 - Ταξινόμηση

Συγχώνευση σειριακών αρχείων (merge)

- Θεωρούμε ότι τα p αρχεία εισόδου είναι ταξινομημένα

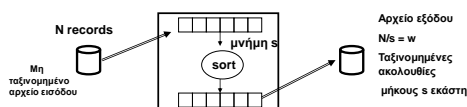


- **module merge (infile1, infile2, ..., infilen, outfile)**
Διάβασε μια εγγραφή από κάθε αρχείο εισόδου
while τουλάχιστον μια εγγραφή δεν είναι EOF **do**
Διάλεξε την εγγραφή με το μικρότερο κλειδί
Γράψε τη στο αρχείο εξόδου
Διάβασε την επόμενη εγγραφή από το ίδιο file

όπου, EOF = τέλος αρχείου

ΤΑΞΙΝΟΜΗΣΗ ΑΡΧΕΙΟΥ 1/3

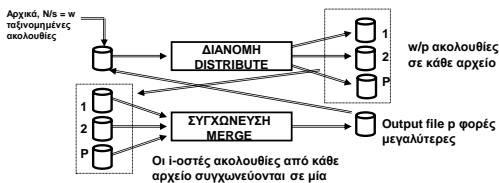
- Είναι διαφορετική για αρχεία εγγραφών σε σχέση με πίνακες όταν αυτά δεν χωρούν ολόκληρα στην μνήμη
- Στρατηγική: φθάνε όλο και μεγαλύτερες διατεταγμένες ακολουθίες εγγραφών συγχωνεύοντας μικρότερες τέτοιες ακολουθίες
- **Πολυφασική ταξινόμηση** (multiphase sort)
 - Στην πρώτη φάση (προπαρασκευαστική) διαβάζουμε από το αρχείο των N εγγραφών ομάδες των s το πλήθος εγγραφών στη μνήμη του υπολογιστή, τις ταξινομούμε και στη συνέχεια τις γράφουμε σε ένα αρχείο εξόδου.



• Το αποτέλεσμα είναι το αρχείο εξόδου να έχει $w = N/s$ ταξινομημένες ακολουθίες εγγραφών

ΤΑΞΙΝΟΜΗΣΗ ΑΡΧΕΙΟΥ 2/3

Στη συνέχεια ακολουθεί ένας αριθμός κύκλων φάσης διανομής και φάσης συγχώνευσης, όπως φαίνεται στο παρακάτω σχήμα, έως ότου προκύψει ένα πλήρως ταξινομημένο αρχείο N εγγραφών.
• Στη φάση της διανομής οι ταξινομημένες ακολουθίες διανέμονται σε p το πλήθος αρχεία εξόδου.
• Στη φάση της συγχώνευσης τα p αρχεία συγχωνεύονται σε ένα με αποτέλεσμα οι ταξινομημένες ακολουθίες που προκύπτουν να είναι p φορές μεγαλύτερες αυτών της εισόδου.



ΤΑΞΙΝΟΜΗΣΗ ΑΡΧΕΙΟΥ 3/3

• Αριθμός περασμάτων είναι $= 2 \log_p(N/s) + 1$

Πράγματι:

- έστω N = αριθμός εγγραφών
 s = μήκος των αρχικών ακολουθιών (1η φάση)
 p = αριθμός ενδιάμεσων αρχείων
- έστω k ο αριθμός κύκλων διανομής-συγχώνευσης
- Τότε θα έχουμε:
 - $s \cdot p$ = το μήκος της ταξινομημένης ακολουθίας μετά τον 1ο κύκλο
 - $s \cdot p^2$ = το μήκος της ταξινομημένης ακολουθίας μετά τον 2ο κύκλο
 - $s \cdot p^3$ = το μήκος της ταξινομημένης ακολουθίας μετά τον 3ο κύκλο
 - ...
 - $N = s \cdot p^k$ = το μήκος της ταξινομημένης ακολουθίας μετά τον k κύκλο
 - $k = \log_p(N/s)$
- Λαμβάνοντας υπόψη και την αρχική φάση διανομής έπεται ότι ο αριθμός περασμάτων $= 2k + 1 = 2 \log_p(N/s) + 1$
- το K μειώνεται αυξάνοντας τα S ή p και P
- απαιτούμενος χρόνος $N(2k+1)$
- Ασυμπτωτική χρονική πολυπλοκότητα : $O(N \log_p N)$

Neapolis University Pafos Introduction to Computer Science and networks

ΠΑΡΑΔΕΙΓΜΑ
 T=ταξινόμηση
 Δ= Διανομή
 Σ=συναγώνευση

5	Δ	3	Δ	Δ	Δ	1
80	5	5	3	3	3	2
3	80	80	5	5	4	3
90	90	90	8	8	5	4
75	Δ	Δ	Δ	Δ	7	7
2	8	17	75	75	8	8
8	12	21	80	80	12	12
12	75	35	90	90	17	17
35	4	1	4	1	20	20
17	17	19	7	11	21	21
4	21	40	17	15	35	35
21	35	65	20	19	75	75
20	7	2	21	40	80	80
7	20	8	35	65	41	81
100	81	12	81	100	90	90
81	100	75	100	100	100	100
40	1	7	Δ	Δ	Δ	65
19	19	20	11	7	11	11
65	40	81	15	17	15	15
1	65	100	19	20	19	19
15	11	11	40	21	40	40
11	15	15	65	35	65	65
102	100	100	81	100	100	100
100	102	102	100	100	102	102
T	Δ	Σ	Δ	Σ	Δ	Σ
P=2	P=2	P=2	P=2	P=2	P=2	P=2

N = 24
 s = 4
 w = N/s = 6

$k = 2\log_2 6 + 1 = 7$

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 67

Neapolis University Pafos Introduction to Computer Science and networks

ΕΝΗΜΕΡΩΣΗ ΣΕΙΡΙΑΚΟΥ ΑΡΧΕΙΟΥ

```

    graph LR
        A[Unsorted transaction file] --> B[SORT]
        B --> C[Sorted]
        C --> D[UPDATE]
        E[OLD Master File] --> D
        D --> F[NEO Master File]
    
```

Είδη συναλλαγών

- Εισαγωγή (insertion) νέας εγγραφής
- Ενημέρωση (update) εγγραφής
- Διαγραφή (deletion) εγγραφής

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 68