

22Υ103 Εισαγωγή Υπολογιστές

Ενότητα #9.1

Λειτουργικά Συστήματα

22Υ103 Εισαγωγή Υπολογιστές - Ν. Αβούρης
(Διάλεξη 2014-2015)

Το Λειτουργικό Σύστημα...

είναι ένα σύνολο από προγράμματα που δρουν ως ενδιάμεσο μεταξύ των εφαρμογών ενός υπολογιστικού συστήματος και του υλικού του υπολογιστή.

Παραδείγματα από Λειτουργικά Συστήματα

Linux

HPUX

Windows

BSDI

FreeBSD

IRIX

Android

AIX

Be OS

Solaris

MacOS

OS/2

UnixWare

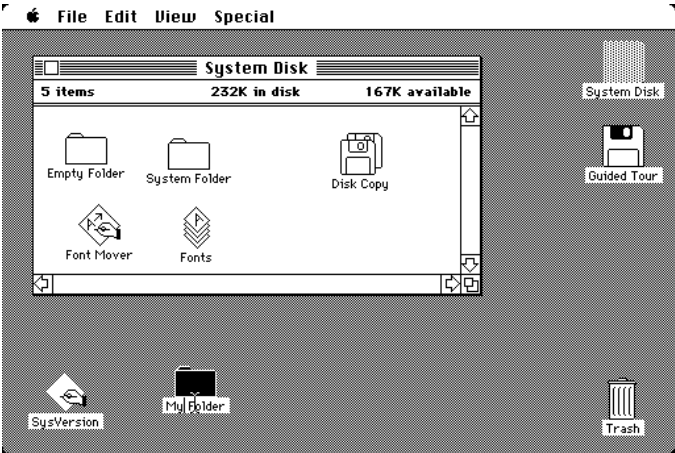
OpenServer

«Ένας ελέφαντας είναι ένα ποντίκι με λειτουργικό σύστημα»
—Knuth

```
Current date is Tue 1-01-1980
Enter new date:
Current time is 7:48:27.13
Enter new time:

The IBM Personal Computer DOS
Version 1.10 (C)Copyright IBM Corp 1981, 1982

A>dir/w
COMMAND COM FORMAT COM CHKDSK COM SYS COM DISKCOPY COM
DISKCOMP COM COMP COM EXE2BIN EXE MODE COM EDLIN COM
DEBUG COM LINK EXE BASIC COM BASICA COM ART BAS
SAMPLES BAS MORTGAGE BAS COLORBAR BAS CALENDAR BAS MUSIC BAS
DONKEY BAS CIRCLE BAS PIECHART BAS SPACE BAS BALL BAS
COMM BAS
        26 File(s)
A>dir command.com
COMMAND COM 4959 5-07-82 12:00p
        1 File(s)
A>
```



Λειτουργίες...

- Το Λειτουργικό Σύστημα συντονίζει τη χρήση των **πόρων του συστήματος**, όπως της **Κεντρικής Μνήμης**, του χρόνου της **Κεντρικής Μονάδας Επεξεργασίας** και της πρόσβασης σε περιφερειακές συσκευές (**δίσκων - συσκευών εισόδου/εξόδου**).
- Το Λειτουργικό Σύστημα **προστατεύει το υλικό** από μη επιτρεπτή χρήση του από τις εφαρμογές των χρηστών.

Το ΛΣ : σχέση με εικονική μηχανή



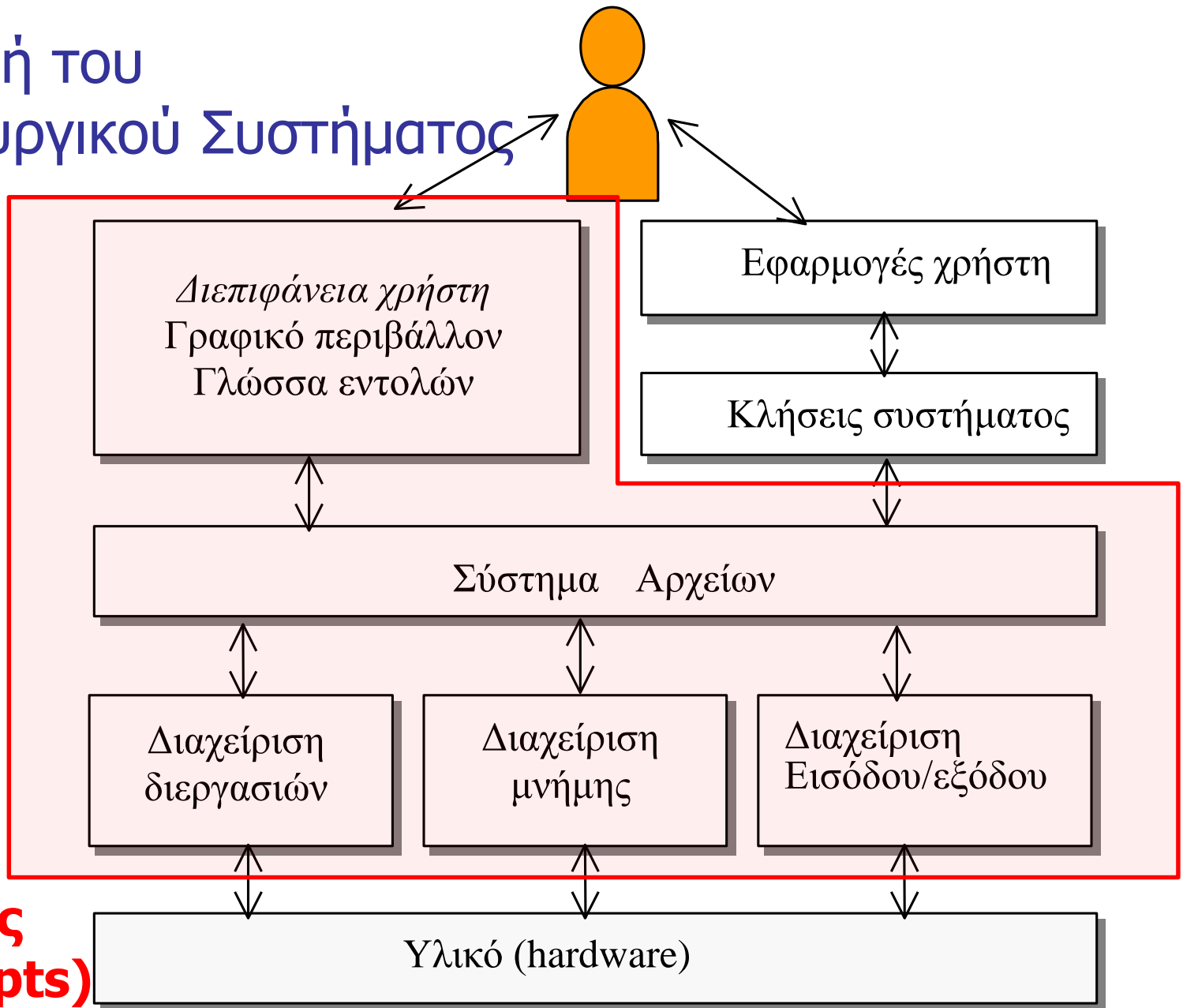
Ιστορικά στοιχεία

- 1η γενιά (δεκαετία 40-50) υπολογιστών : **συστήματα ομαδικής επεξεργασίας** (batch processing).
- 2η γενιά (δεκαετία του 60) : **Λειτουργικά Συστήματα πολύ-προγραμματισμού** (multiprogramming). αν η διεργασία είναι ανενεργή, σε αναμονή μιας περιφερειακής συσκευής, αποβάλλεται προσωρινά από την ΚΜΕ.
- 3η γενιά (από τη δεκαετία 70..) : δυνατότητα **καταμερισμού του χρόνου** μεταξύ των διεργασιών (**time sharing**). Κατάλληλα για υπολογιστές πολλών χρηστών.

Συστήματα καταμερισμού χρόνου

- Οι **απαιτήσεις σε μνήμη** οδήγησαν σε **συστήματα εικονικής μνήμης** (virtual memory), δηλαδή στη χρήση τμήματος περιφερειακής μνήμης σαν βοηθητικού χώρου της κεντρικής μνήμης.
- Απαιτήσεις για **διαχείριση συστημάτων αρχείων**
- εξελιγμένη **διαχείριση των διεργασιών** και του χρόνου της ΚΜΕ.
- μηχανισμοί **αποφυγής αδιεξόδων** μεταξύ των διεργασιών δηλαδή καταστάσεων κατά τις οποίες δύο διεργασίες βρίσκονται επ-άπειρο η μια εν αναμονή της άλλης.

Η δομή του Λειτουργικού Συστήματος



**Διακοπές
(interrupts)**

Ερώτηση

- Ποια η διαφορά μεταξύ κλήσεων συστήματος και διακοπών;

Η δομή του Λειτουργικού Συστήματος

- Το υποσύστημα **διαχείρισης διεργασιών,**
- το υποσύστημα **διαχείρισης της κεντρικής μνήμης,**
- Το υποσύστημα **διαχείρισης συστήματος αρχείων,**
- το υποσύστημα **διαχείρισης συσκευών εισόδου / εξόδου,**
- το υποσύστημα **διαχείρισης συσκευών δευτερεύουσας αποθήκευσης,**
- το υποσύστημα **δικτύωσης,**
- το υποσύστημα **προστασίας και ασφάλειας**
- το υποσύστημα **κελύφους εντολών και περιβάλλοντος χρήστη.**

Διακοπές (interrupts): επικοινωνία λειτουργικού με το υλικό

- **Διακοπές (interrupts)** είναι **σήματα** που στέλνονται από τις συσκευές στην ΚΜΕ. Η ΚΜΕ διακόπτει τη λειτουργία της να χειριστεί το σήμα διακοπής.
- **Interrupt handler:** το τμήμα του λειτουργικού συστήματος που χειρίζεται τις διακοπές.
- **Interrupt vector** οι διευθύνσεις των ρουτινών που απαντούν στα σήματα διακοπών
- Χειρισμός προτεραιότητας διακοπών, μηχανισμός **επιβολής μάσκας (masking)**

#1 Διαχείριση διεργασιών

Διεργασία (process) ορίζεται ως ένα πρόγραμμα που εκτελείται στην ΚΜΕ ενός υπολογιστή.

Για να εκτελεστεί η διεργασία απαιτούνται πόροι του συστήματος, όπως χρόνος της ΚΜΕ, χώρος στη μνήμη, χώρος στο σύστημα αρχείων και τις περιφερειακές συσκευές, συσκευές εισόδου/εξόδου κλπ.

Διεργασίες: windows

Task Manager					
File Options View					
Processes Performance App history Startup Users Details Services					
Name	Status	CPU	Memory	Disk	Network
PyCharm Community Edition (3...		0%	125,9 MB	0,1 MB/s	0 Mbps
Thunderbird (32 bit)		0%	121,0 MB	0 MB/s	0 Mbps
Skype (32 bit)		0%	95,2 MB	0 MB/s	0 Mbps
Windows Explorer (4)		1,2%	73,0 MB	0,1 MB/s	0 Mbps
Microsoft PowerPoint (32 bit)		0%	67,9 MB	0 MB/s	0 Mbps
Antimalware Service Executable		0%	57,6 MB	0,1 MB/s	0 Mbps
Service Host: Local System (13)		0%	40,1 MB	0 MB/s	0 Mbps
Service Host: Local System (Net...		0%	39,5 MB	0 MB/s	0 Mbps
Microsoft Windows Search Inde...		0%	24,9 MB	0 MB/s	0 Mbps
Desktop Window Manager		2,2%	17,9 MB	0,1 MB/s	0 Mbps
Service Host: Local Service (No ...		0%	16,0 MB	0 MB/s	0 Mbps
ExpressCache Service		0%	15,9 MB	0 MB/s	0 Mbps
Service Host: Local Service (Net...		0%	15,2 MB	0 MB/s	0 Mbps
Service Host: Network Service (5)		0%	12,5 MB	0 MB/s	0 Mbps
Intel(R) Management and Secur...		0%	11,7 MB	0 MB/s	0 Mbps
Service Host: Local Service (8)		0%	11,3 MB	0,1 MB/s	0 Mbps
Task Manager		2,5%	11,1 MB	0 MB/s	0 Mbps
Intel® SBA WindowsService (32 ...		0%	10,8 MB	0 MB/s	0 Mbps
WMI Provider Host		0%	9,8 MB	0 MB/s	0 Mbps
Location Task Manager (32 bit)		0%	8,0 MB	0 MB/s	0 Mbps
Spooler SubSystem App		0%	7,9 MB	0 MB/s	0 Mbps

- Base Priority
- CPU Time
- CPU Usage
- GDI Objects
- Handle Count
- Image Name
- I/O Other
- I/O Other Bytes
- I/O Reads
- I/O Read Bytes
- I/O Writes
- I/O Write Bytes
- Memory Usage
- Memory Usage Delta
- Non-paged Pool
- Page Faults
- Page Faults Delta
- Paged Pool
- Peak Memory Usage
- PID (Process Identifier)
- Session ID
- Thread Count
- User Name
- USER Objects
- Virtual Memory Size

Διεργασίες: UNIX

```
ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	COMMAND
a95-3625	29806	29471	0	09:42:36	tty09	0:18	vi sem1994
daemon	25912	103	0	13:24:38	?	0:00	sendmail -postman.esx.ac.uk
daemon	25993	25912	0	13:31:18	?	0:00	sendmail -AA25993 poman.essex.ac.uk
root	7758	7749	0	Dec 1	ttyp3	0:00	hpterm -C -ls -name Console -Tconsole
a95-3613	212	29810	0	10:15:46	tty11	0:05	vi sem
daemon	1035	103	0	11:02:38	?	0:00	sendmail -vergina.en.auth.gr
root	878	88	0	10:55:21	ttyt3	0:00	telnetd
c94-3133	468	444	0	10:39:26	ttyt1	0:00	elm
a95-3653	27284	1	0	14:49:26	tty02	0:00	-sh
user1	226	225	0	10:17:50	ttyt0	0:01	-csh
a95-3644	27359	1	0	14:51:49	tty10	0:00	-sh
a95-3612	29221	1	0	09:05:28	tty04	0:01	-sh
a95-3644	402	27359	0	10:32:40	tty10	0:00	vi sem1994
root	443	88	0	10:38:50	ttyt1	0:00	telnetd
c94-3212	984	29558	0	10:58:53	tty12	0:00	telnet
a95-3589	430	97	0	10:37:10	tty14	0:02	vi sem1994
root	958	88	0	10:58:15	ttyt2	0:00	telnetd

Διεργασίες: python module psutil

psutil.get_process_list()

```
>>>
%CPU  %MEM  VMS  RSS  NAME,
0.0000  2.72  139190272  84897792  AcroRd32.exe
0.0000  0.21  9285632  6504448  AcroRd32.exe
0.0000  0.11  3067904  3444736  AdobeCollabSync.exe
0.0000  0.60  21860352  18632704  EXCEL.EXE
0.0000  0.04  1036288  1335296  HP1005MC.EXE
0.0000  0.02  864256  548864  ONENOTEM.EXE
0.0000  0.13  29814784  4038656  POWERPNT.EXE
0.0000  0.14  1626112  4440064  SearchFilterHost.exe
0.0000  0.94  58400768  29274112  SearchIndexer.exe
0.0000  0.21  2682880  6545408  SearchProtocolHost.exe
0.0000  2.90  109166592  90267648  Skype.exe
0.0000  0.11  3059712  3510272  SnippingTool.exe
0.0000  0.10  315392  2981888  System
100.0000  0.00  0  24576  System Idle Process
0.0000  0.21  4808704  6590464  TabTip.exe
0.0000  0.06  1445888  1839104  VCDDaemon.exe
0.0000  0.01  847872  311296  armsvc.exe
0.0000  0.45  15405056  14094336  audiodg.exe
0.0000  0.01  22102016  380928  avgcsrvc.exe
0.0000  0.07  2351104  2174976  avgemcx.exe
0.0000  0.45  14823424  13967360  avgidsagent.exe
0.0000  0.04  11177984  1130496  avgnsx.exe
.....
```


Διασύνδεση Python με λειτουργικό σύστημα: το module os

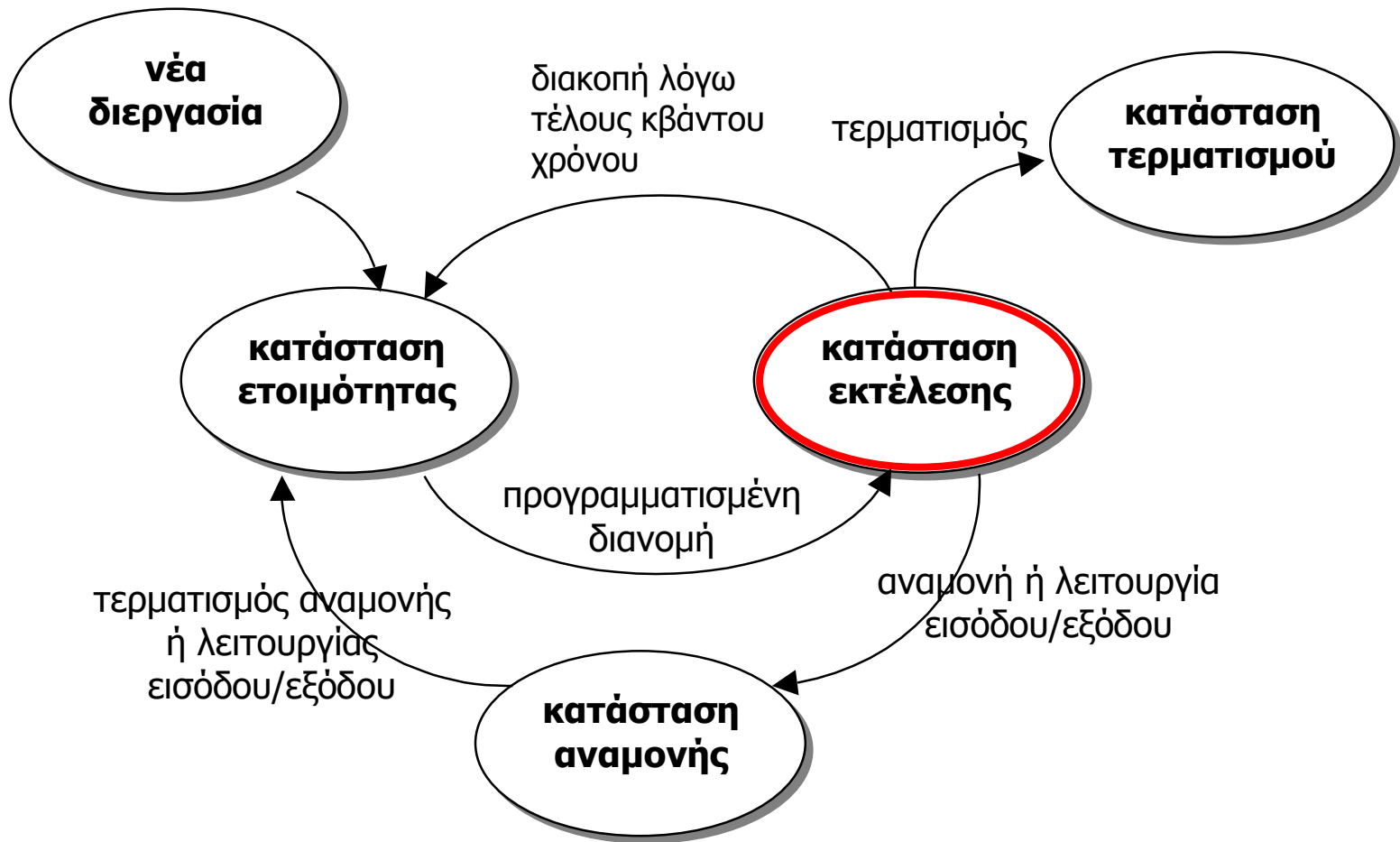
```
import os
os.getcwd()
os.chdir('/server/accesslogs')
os.system('mkdir today')
os.listdir(d)
os.rename( "test1.txt", "test2.txt" )
os.remove(file_name)
os.mkdir("newdir")
os.rmdir('dirname')
```

Ερώτηση:

Ποια η διαφορά μεταξύ των εντολών:

- `os.mkdir('today')`
- `os.system('mkdir today')`

Διάγραμμα καταστάσεων διεργασίας



ερώτηση

- Πώς το διάγραμμα καταστάσεων διεργασιών τροποποιείται για τις διάφορες γενιές λειτουργικών συστημάτων;

Κλήσεις συστήματος

Μια διεργασία επικοινωνεί με το ΛΣ μέσω των **κλήσεων συστήματος (system calls)**.

Οι κλήσεις αυτές είναι **εντολές προς το ΛΣ** στις οποίες παρέχεται πρόσβαση στον προγραμματιστή μέσω **βιβλιοθηκών συναρτήσεων ή υπορουτινών** γλωσσών ανωτέρου επιπέδου.

Π.χ. open, read, write, close, wait, exec, fork, exit, kill

Κλήσεις συστήματος: **έλεγχος διεργασιών**

- τέλος - καταστροφή διεργασίας
- φόρτωμα και εκτέλεση διεργασίας
- δημιουργία διεργασίας
- ανάκτηση και τροποποίηση χαρακτηριστικών παραμέτρων διεργασίας
- θέσε την διεργασία σε κατάσταση αναμονής για χρόνο t
- διάθεση μνήμης σε διεργασία

Κλήσεις συστήματος: **Διαχείριση αρχείων**

- δημιουργία και διαγραφή αρχείων
- άνοιγμα και κλείσιμο αρχείων
- ανάγνωση και εγγραφή πληροφορίας σε αρχείο
- ανάκτηση και τροποποίηση χαρακτηριστικών αρχείου

Κλήσεις συστήματος: **Διαχείριση συσκευών**

- αίτημα για συσκευή και ελευθέρωση συσκευής
- ανάγνωση /εγγραφή σε συσκευή
- ανάκτηση και τροποποίηση χαρακτηριστικών συσκευής
- λογική συσχέτιση ή αποδέσμευση συσκευών

Κλήσεις συστήματος: **Διαχείριση πληροφορίας**

- ανάκτηση ή διόρθωση χρόνου ημερομηνίας
- ανάκτηση ή διόρθωση δεδομένων συστήματος
- ανάκτηση ή τροποποίηση χαρακτηριστικών αρχείου ή συσκευής

Κλήσεις συστήματος: **ΕΠΙΚΟΙΝΩΝΙΕΣ**

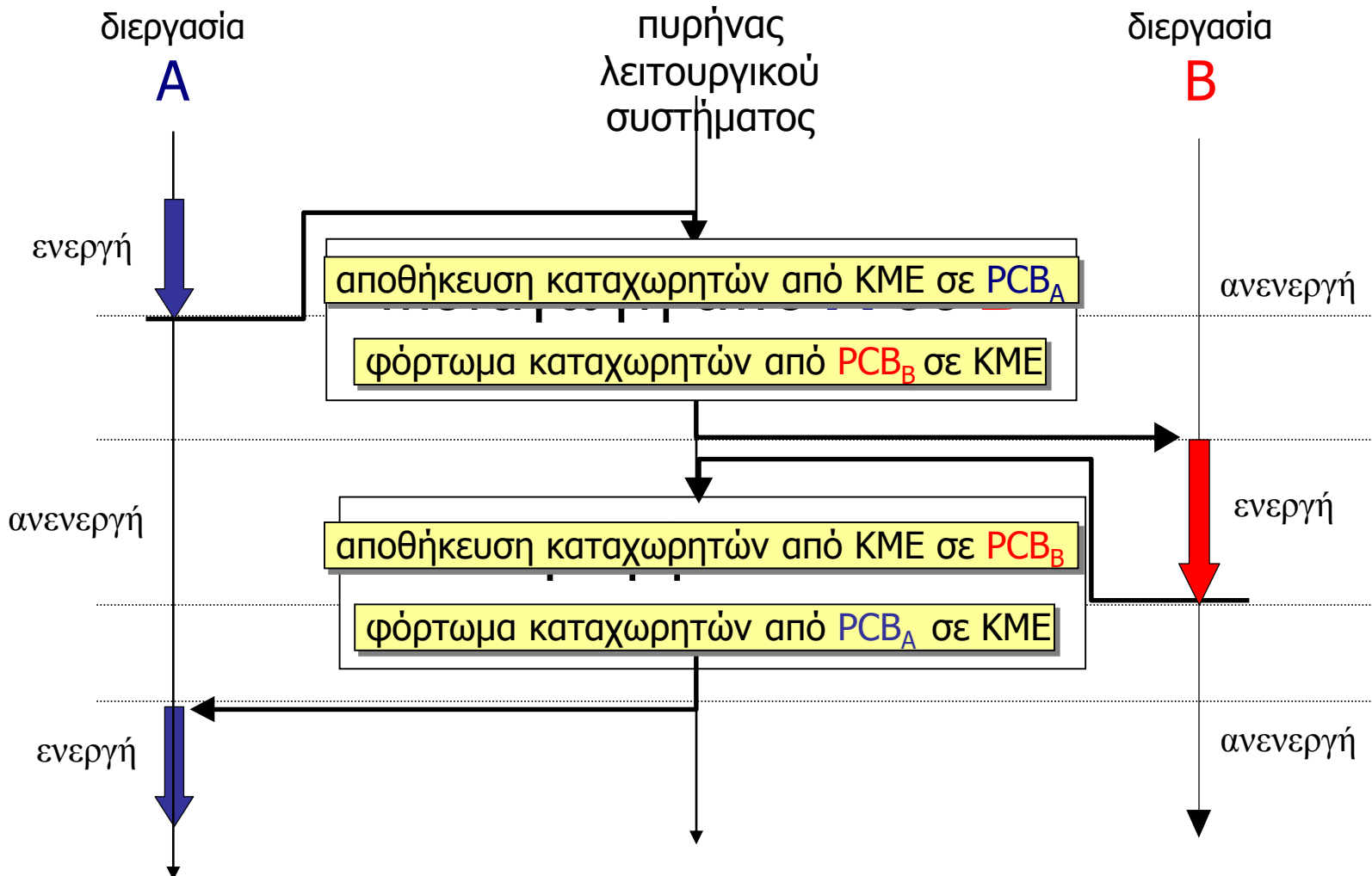
- δημιουργία ή διαγραφή σύνδεσης επικοινωνίας
- αποστολή και παραλαβή μηνυμάτων
- μεταφορά πληροφορίας κατάστασης
- σύνδεση και αποσύνδεση απομακρυσμένης συσκευής

Process Control Block

Κάθε διεργασία αναπαρίσταται στο ΛΣ από μία δομή ελέγχου διεργασίας (process control block, PCB).

← δείκτης	κατάσταση διεργασίας
ταυτότητα διεργασίας	
απαριθμητής προγράμματος	
καταχωρητές	
όρια μνήμης	
λίστα ανοικτών αρχείων	
κατάσταση εισόδου/εξόδου	
· · ·	

Μεταγωγή περιβάλλοντος



Νήματα (threads) ή ελαφρές διεργασίες (light process)

Κάθε **διεργασία-νήμα** διαθέτει ένα απαριθμητή προγράμματος, καταχωρητές και ένα σωρό βοηθητικών πληροφοριών, όμως δεν διαθέτει ξεχωριστό χώρο στη μνήμη για τον κώδικα και τα δεδομένα, καθώς επίσης μοιράζεται τους πόρους του συστήματος, όπως ανοικτά αρχεία, σήματα από το ΛΣ με άλλα νήματα.

Μια ομάδα νημάτων που μοιράζονται πόρους απαρτίζουν μια **εργασία (task)**.

Χρονοπρογραμματισμός διεργασιών

- **μη διακοπτοί αλγόριθμοι (non preemptive)**
επιτρέπουν σε μια διεργασία που έχει ήδη την ΚΜΕ υπό τον έλεγχο της, να τον διατηρήσει μέχρι αυτή να ολοκληρωθεί ή να απαιτήσει η ίδια μια άλλη λειτουργία, (πχ λειτουργία εισόδου/εξόδου).
- **διακοπτοί αλγόριθμοι (preemptive)**
επιτρέπουν τη διακοπή μιας διεργασίας που βρίσκεται σε εκτέλεση μετά παρέλευση ενός ορισμένου χρόνου (κβάντο χρόνου).

Αλγόριθμοι χρονοπρογραμματισμού

μη διακοπτοί (non preemptive)

- **FCFS** (first come first served)
- **SJF** (shortest job first)
- **Priority scheduling**

διακοπτοί (preemptive)

- **RR** (round robin)
- **SRT** (shortest remaining time)
- **Priority scheduling**
- **Multilevel feedback queues**

Μη διακοπτοί Αλγόριθμοι...

- **First Come First Serve (FCFS)**

Το κριτήριο επιλογής μιας διεργασίας είναι η σειρά άφιξης. Οι διεργασίες όταν εισέρχονται σε κατάσταση ετοιμότητας εισάγονται σε μια ουρά τύπου FIFO.

- **Shortest Job First (SJF)**

Το κριτήριο επιλογής είναι η διάρκεια της διεργασίας.

- **Εξυπηρέτηση με βάση την προτεραιότητα (Priority scheduling).**

Διακοπτοί αλγόριθμοι

Αλγόριθμος κυκλικής επαναφοράς (RR round robin). κβαντο χρόνου (quantum) σε κάθε διεργασία με τη σειρά (αντίστοιχος του μη διακοπτού αλγόριθμου FCFS.)

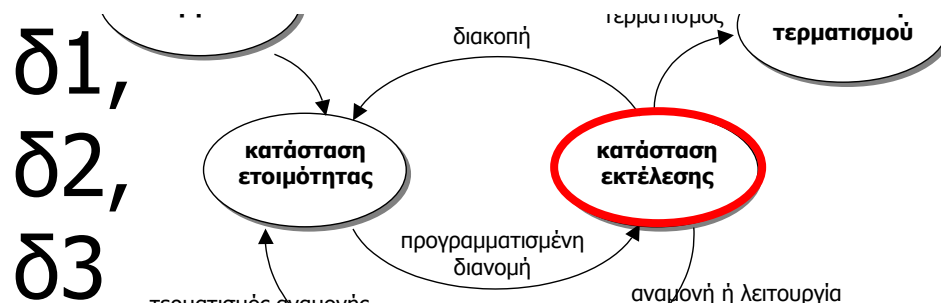
Αλγόριθμος ελάχιστου υπολειπόμενου χρόνου (SRT Shortest Remaining Time). Αντίστοιχος με μη διακοπτό αλγόριθμο Shortest Job First. διαφορά: Όταν μια καινούργια διεργασία εισέρχεται η διεργασία που εκτελείται διακόπτεται.

Εξυπηρέτηση εκ περιτροπής από ουρές προτεραιότητας (priority queues)

Αλγόριθμος πολλαπλών ουρών με ανατροφοδότηση (multilevel feedback queue).

Παράδειγμα

- Έστω τρεις διεργασίες $\delta_1, \delta_2, \delta_3$ που εισέρχονται με αυτή τη σειρά σε κατάσταση ετοιμότητας κατά τη χρονική στιγμή 0 και αναμένεται να έχουν χρόνους εκτέλεσης στην ΚΜΕ (CPU burst) 21 msec, 3 msec, 7 msec. Να περιγραφεί η εκτέλεση τους σύμφωνα με τους αλγόριθμους FCFS και SJF



Παράδειγμα 2

- Έστω τρεις διεργασίες $\delta_1, \delta_2, \delta_3$ που εισέρχονται με αυτή τη σειρά σε κατάσταση ετοιμότητας κατά τη χρονική στιγμή 0 και αναμένεται να έχουν χρόνους εκτέλεσης στην ΚΜΕ (CPU burst) 21 msec, 3 msec, 7 msec. Να περιγραφεί η εκτέλεση τους σύμφωνα με τον αλγόριθμο RR αν υποθέσουμε κβάντο χρόνου 4 msec.
- Τι γίνεται αν υποθέσουμε χρόνο μεταγωγής περιβάλλοντος 1 msec;

Παράδειγμα 3

Αν υποθέσουμε τις ακόλουθες διεργασίες, οι οποίες εισέρχονται σε κατάσταση ετοιμότητας τις χρονικές στιγμές που φαίνονται στον πίνακα με τους εξής αναμενόμενους χρόνους εκτέλεσης:

Διεργασία	άφιξη	χρόνος ΚΜΕ
■ δ1	0	8
■ δ2	1	4
■ δ3	5	9
■ δ4	10	5

Να βρεθεί ο μέσος χρόνος αναμονής αν εκτελεστούν οι διεργασίες με τον αλγόριθμο RR και κβάντο χρόνου 4 χρ. στ. Υποθέστε χρόνο μεταγωγής περιβάλλοντος = 1 χρ. στιγμή.

Να συγκρίνετε με την εκτέλεση με τον αλγόριθμο SRT

Φροντιστηριακή άσκηση

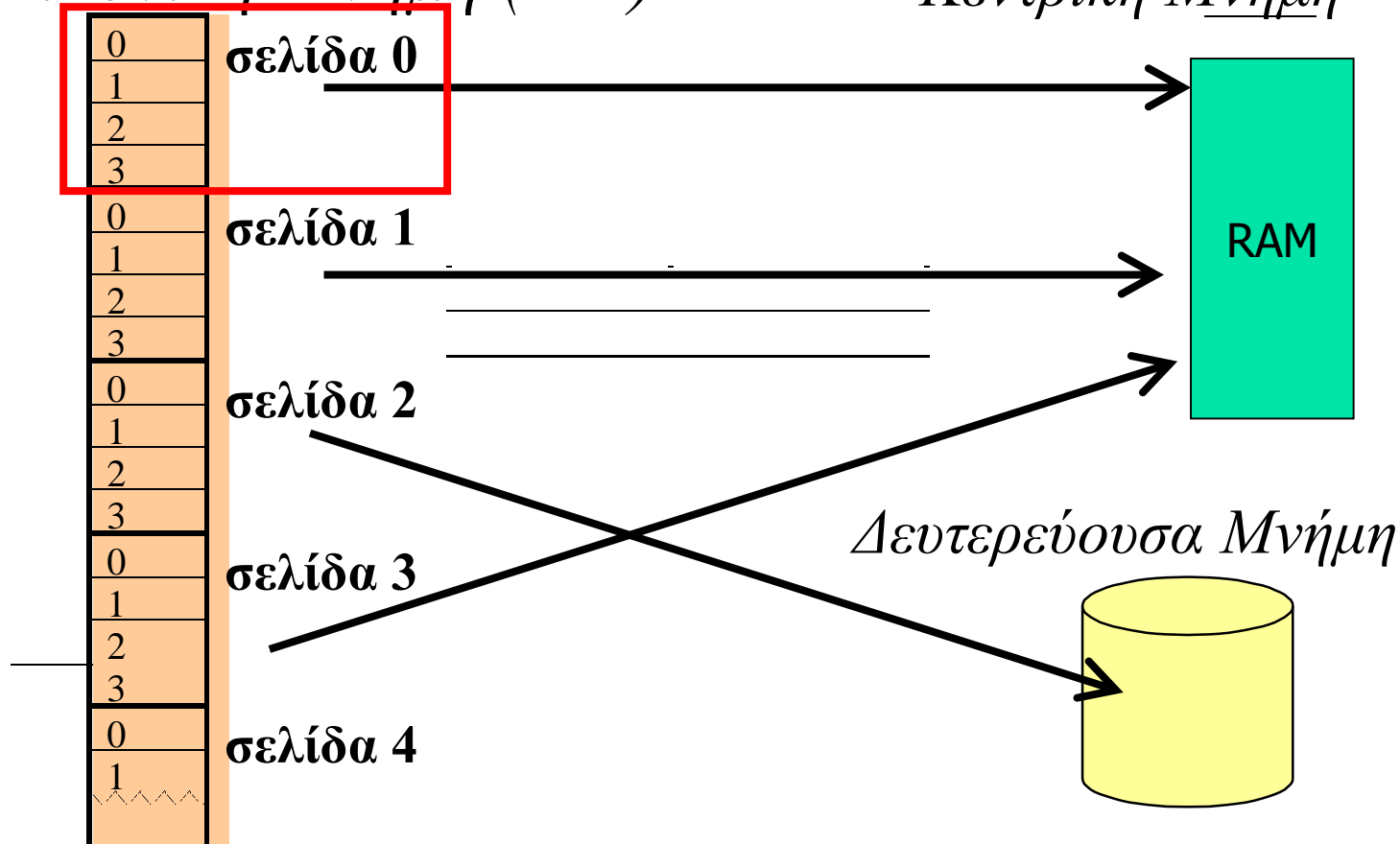
- Έστω τρεις διεργασίες $\delta_1, \delta_2, \delta_3$ που εισέρχονται σε κατάσταση ετοιμότητας κατά τις χρονικές στιγμές 0, 8, 12 αντίστοιχα. Η προσδοκώμενη διάρκεια εκτέλεσης τους στην ΚΜΕ (CPU burst) είναι αντίστοιχα 18 msec, 8 msec, 11 msec. (α) Να περιγραφεί διαγραμματικά η εκτέλεση τους σύμφωνα με τον αλγόριθμο RR αν υποθέσουμε κβάντο χρόνου 5 msec και χρόνο μεταγωγής περιβάλλοντος 1 msec. (β) το ίδιο για κβάντο χρόνου 10 msec. (γ) Στη συνέχεια να περιγραφεί διαγραμματικά η διαδικασία εκτέλεσης τους με βάση τον αλγόριθμο SRT με προεκκένωση.
- Να υπολογίσετε τη μέση τιμή του χρόνου απόκρισης του συστήματος για τις διαφορετικές αυτές περιπτώσεις αλγορίθμων χρονοπρογραμματισμού διεργασιών.

Διαχείριση Κεντρικής Μνήμης

Διαχείριση Κεντρικής Μνήμης: Εικονική Μνήμη (Virtual Memory)

Εικονική Μνήμη (EM)

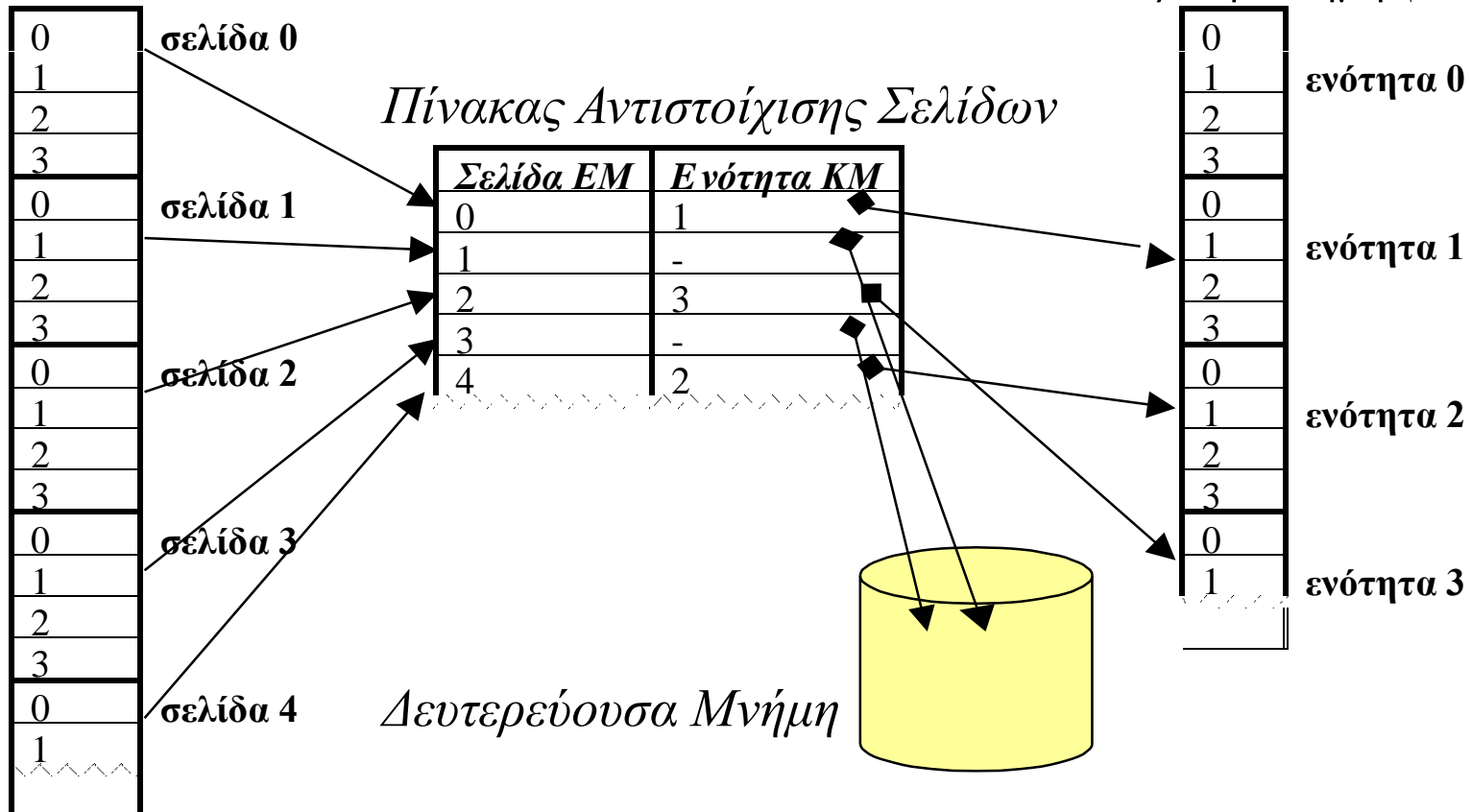
Κεντρική Μνήμη



Εικονική Μνήμη: Πίνακας Αντιστοίχισης σελίδων Εικονικής Μνήμης – Ενοτήτων Κεντρικής Μνήμης

Εικονική Μνήμη (EM)

Κεντρική Μνήμη (KM)



Διαχείριση Εικονικής Μνήμης

- Ανάκληση σελίδων:
 - κατά απαίτηση της διεργασίας (demand paging)
 - Προ-ανάκληση σελίδων (prefetching).

Στρατηγικές αντικατάστασης σελίδων Εικονικής Μνήμης

- με βάση τον χρόνο παραμονής στην ΚΜ (first in first out - **FIFO**- strategy).
- στρατηγική ελάχιστης πρόσφατης χρήσης (least recently used **LRU** strategy)
- στρατηγική σπανιότερης χρήσης (least frequently used **LFU**)
- στρατηγική της μη πρόσφατης χρήσης (not recently used **NRU** strategy)
- στρατηγική μη τροποποιημένης σελίδας (non modified page **NMP** strategy). – χρήση dirty bit

Παράδειγμα διαχείρισης εικονικής μνήμης

Υποθέτουμε KM μεγέθους 3 ενοτήτων. Έστω ότι οι αναφορές της ΚΜΕ σε σελίδες μιας διεργασίας γίνονται σύμφωνα με την παρακάτω ακολουθία: 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1.

Αν υποθέσουμε ότι καμιά σελίδα της διεργασίας δεν βρίσκεται στην ΚΜ αρχικά, ζητείται να περιγραφούν τα περιεχόμενα της ΚΜ κατά την εκτέλεση της διεργασίας και να μετρηθούν τα σφάλματα σελίδας, αν υποθέσουμε ότι χρησιμοποιούνται οι αλγόριθμοι αντικατάστασης σελίδας FIFO και LRU.

Λύση..

■ FIFO

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
* * * * * * * * * * * * * * * * * *

7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
	0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
		1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1

■ LRU

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
* * * * * * * * * * * * * * * * *

7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7

Άσκηση

- Έστω σύστημα διαχείρισης εικονικής μνήμης με σελιδοποίηση.
- Έστω ότι το μέγεθος κάθε σελίδας είναι 4 bytes και το μέγεθος κάθε λέξης (θέσης μνήμης) είναι 1 byte.
- Ο πίνακας αντιστοίχισης σελίδων είναι:
- Να βρεθεί η **φυσική διεύθυνση** που αντιστοιχεί στην εικονική διεύθυνση $v=5$.
- ΣΗΜΕΙΩΣΗ ΟΙ ΔΙΕΥΘΥΝΣΕΙΣ ΠΑΙΡΝΟΥΝ ΤΙΣ ΤΙΜΕΣ 0,1,2,3 ...

ΠΑΣ	
0	5
1	6
2	1
3	2

Λύση

	0	1	2	3
0				
1				
2				
3				

0	5
1	6
2	1
3	2

	0	1	2	3
0				
1				
2				
3				
4				
5				
6				
7				

$VM_{\text{address}}=5,$
 $(p,d)=(1,1)$

$(p,d)=(6,1), RAM_{\text{address}}=25$

Διαχείριση Συστήματος αρχείων

- **αρχείο (file)** είναι μια λογική ομάδα δεδομένων, στα οποία έχει δοθεί ένα όνομα και χώρος σε κάποια συσκευή αποθήκευσης.
- Η σημαντική ιδιότητα των αρχείων είναι ότι αυτά αποθηκεύονται μόνιμα σε δευτερεύουσες συσκευές αποθήκευσης όπως είναι οι μαγνητικοί δίσκοι ή οπτικά μέσα αποθήκευσης

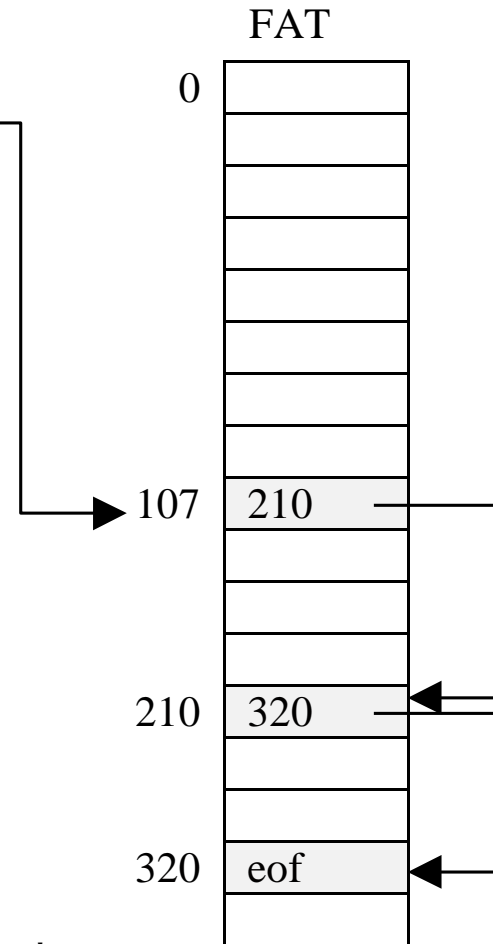
Αποθήκευση αρχείων και καταλόγων

- ακολουθιακή αποθήκευση (contiguous allocation),
- διασυνδεδεμένη αποθήκευση (linked allocation)
- αποθήκευση με χρήση ευρετηρίου (indexed allocation).

Διασυνδεδεμένη αποθήκευση

- Χρήση πίνακα FAT

filename	...	start block
file_1	...	107



Άσκηση

0	10
1	6
2	12
3	eof
4	0
5	8
6	4
7	eof
8	2
9	13
10	9
11	18
12	14
13	eof
14	11
15	eof
16	19
17	20
18	eof
19	eof
20	19

File_1 start_block = 7
File_2 start_block = 8
File_3 start_block = 1
File_4 start_block = 17

(α) Έστω ένα σύστημα αρχείων (file system) με διασυνδεδεμένη αποθήκευση με χρήση Πίνακα Διάθεσης Αρχείων. Ο πίνακας διάθεσης αρχείων φαίνεται στα αριστερά. Έστω ότι σε ένα κατάλογο περιλαμβάνονται τα τέσσερα αρχεία που φαίνονται παραπάνω. Να υπολογίσετε το συνολικό χώρο που καταλαμβάνουν τα αρχεία αυτά στο δίσκο αν υποθέσουμε μέγεθος μπλόκ 512 bytes.

Διαχείριση αρχείων στην python

```
fo = open("foo.txt", "wb")  
print "Name of the file: ", fo.name  
print "Closed or not : ", fo.closed  
print "Opening mode : ", fo.mode  
print "Softspace flag : ", fo.softspace
```

```
Name of the file: foo.txt  
Closed or not : False  
Opening mode : wb  
Softspace flag : 0
```

platform

```
import platform
```

```
>>> platform.system()
```

```
'Windows'
```

```
>>> platform.release()
```

```
'8'
```

```
>>>
```

os.path

```
print "using", os.name, "..."  
print "split", "=>", os.path.split(filename)  
print "splittext", "=>", os.path.splittext(filename)  
print "dirname", "=>", os.path.dirname(filename)  
print "basename", "=>", os.path.basename(filename)  
print "join", "=>",  
os.path.join(os.path.dirname(filename),  
             os.path.basename(filename))
```

Φροντιστηριακή άσκηση

- Δίδεται διεργασία που καταλαμβάνει 5 σελίδες εικονικής μνήμης (0,1,2,3,4). Η διεργασία εκτελείται σε σύστημα με 3 ενότητες κύριας μνήμης (0,1,2). Η διεργασία ζητάει κατά την εκτέλεση της τις ακόλουθες σελίδες: 3,4,1,2,4,2,0,1. Να βρείτε το χάρτη της κύριας μνήμης στο τέλος της διεργασίας αν χρησιμοποιηθεί στρατηγική αντικατάστασης σελίδων FIFO.

Φροντιστηριακή άσκηση

- Να γράψετε ένα πρόγραμμα σε Python που να υλοποιεί τους αλγόριθμους χρονοπρογραμματισμού FCFS, SJF, RR, SRT.
- Να διαβάσει για κάθε διεργασία τη χρονική στιγμή ετοιμότητας και εκτιμώμενο χρόνο εκτέλεσης (cpu burst time), να δίδεται ο χρόνος μεταγωγής περιβάλλοντος του λειτουργικού.