

Λειτουργικά Συστήματα

Ενότητα 2: Βασικές Έννοιες ΛΣ. POSIX. Κλήσεις Συστήματος.
Δομές ΛΣ. Kernel/Device Drivers. Bootstrap.

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Εργαστήριο Ρομποτικής, Ενσωματωμένων και Ολοκληρωμένων
Συστημάτων

<http://arch.ece.uowm.gr/mdasyg>



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Σκοπός Ενότητας

- Η κατανόηση των κλήσεων συστήματος.
- Η κατανόηση των δομών λειτουργικών συστημάτων.
- Η παρουσίαση των αρχιτεκτονικών δημοφιλών ΛΣ.



Τα ΛΣ συχνά πρέπει να ισορροπήσουν σε αντικρουόμενες απαιτήσεις

Μεγέθη που χαρακτηρίζουν την απόδοση ενός Λ.Σ. :

- Μέση χρησιμοποίηση των πόρων.
- Ρυθμοαπόδοση (throughput).
- Χρόνος απόκρισης.



Βασικές έννοιες ΛΣ

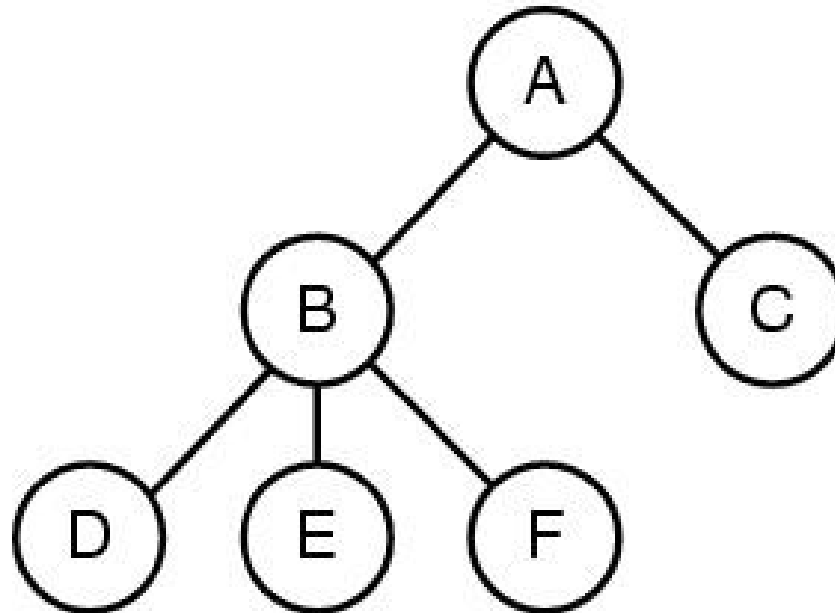
- Processes.
- Address spaces.
- Files.
- Input / Output.
- Protection.
- The shell.



Ιεραρχία Διεργασιών

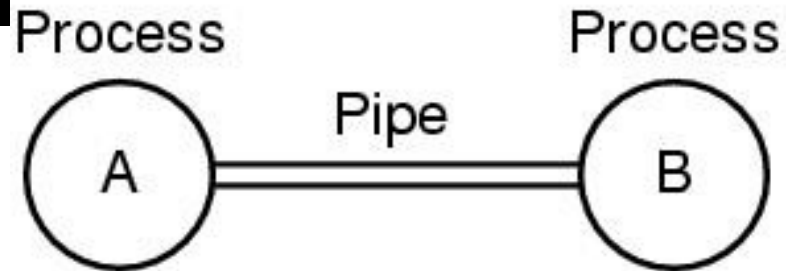
Ένα δέντρο διεργασιών:

Η διεργασία A δημιουργήσε δύο διεργασίες παιδιά, την B και την C.
Η διεργασία B δημιουργήσε τρεις διεργασίες παιδιά, την D, E και F.



Επικοινωνία διεργασιών με διασωλήνωση

```
bigb5# ps
  PID  TT  STAT      TIME COMMAND
12949  u0  I<s+    0:00.03 login
12072  v0  Is+    0:00.00 /usr/libexec/getty Pc ttyv0
12387  v1  Is+    0:00.00 /usr/libexec/getty Pc ttyv1
12681  v2  Is+    0:00.00 /usr/libexec/getty Pc ttyv2
65838   7  Is+    0:00.07 tcsh (csh)
14041   4  I      0:00.00 /usr/local/bin/sudo su - mdasyg
14107   4  I      0:00.01 su - mdasyg
70615   6  Is     0:00.03 login [pam] (login)
33229   1  R+     0:00.00 ps
63659   1  I      0:00.00 /usr/local/bin/sudo su -
63719   1  I      0:00.00 su -
63890   1  S      0:00.19 -su (csh)
65623   1  T      0:02.07 mc
bigb5# ps | grep mc
33695   1  RL+    0:00.00 grep mc
65623   1  T      0:02.07 mc
bigb5#
```



Κατηγορίες Κλήσεων Συστήματος

- Έλεγχος διεργασιών (Process control).
- Διαχείριση αρχείων (File management).
- Διαχείριση συσκευών (Device management).
- Διατήρηση πληροφοριών (Information maintenance).
- Επικοινωνίες (Communications).

** Η υλοποίηση των κλήσεων συστήματος διαφέρει στα λειτουργικά συστήματα ***



Λειτουργία Κλήσεων Συστήματος

Οι κλήσεις συστήματος παρέχουν τη διασύνδεση (*interface*) μεταξύ ενός προγράμματος που εκτελείται και του ΛΣ:

- Γενικώς, διαθέσιμες ως ομάδες εντολές σε γλώσσα assembly.
- Οι γλώσσες που έχουν οριστεί να αντικαταστήσουν την assembly στον προγραμματισμό συστήματος (*system programming*) επιτρέπουν την απευθείας επίκληση κλήσεων συστήματος (π.χ., C, C++) μέσω αντίστοιχων βιβλιοθηκών.

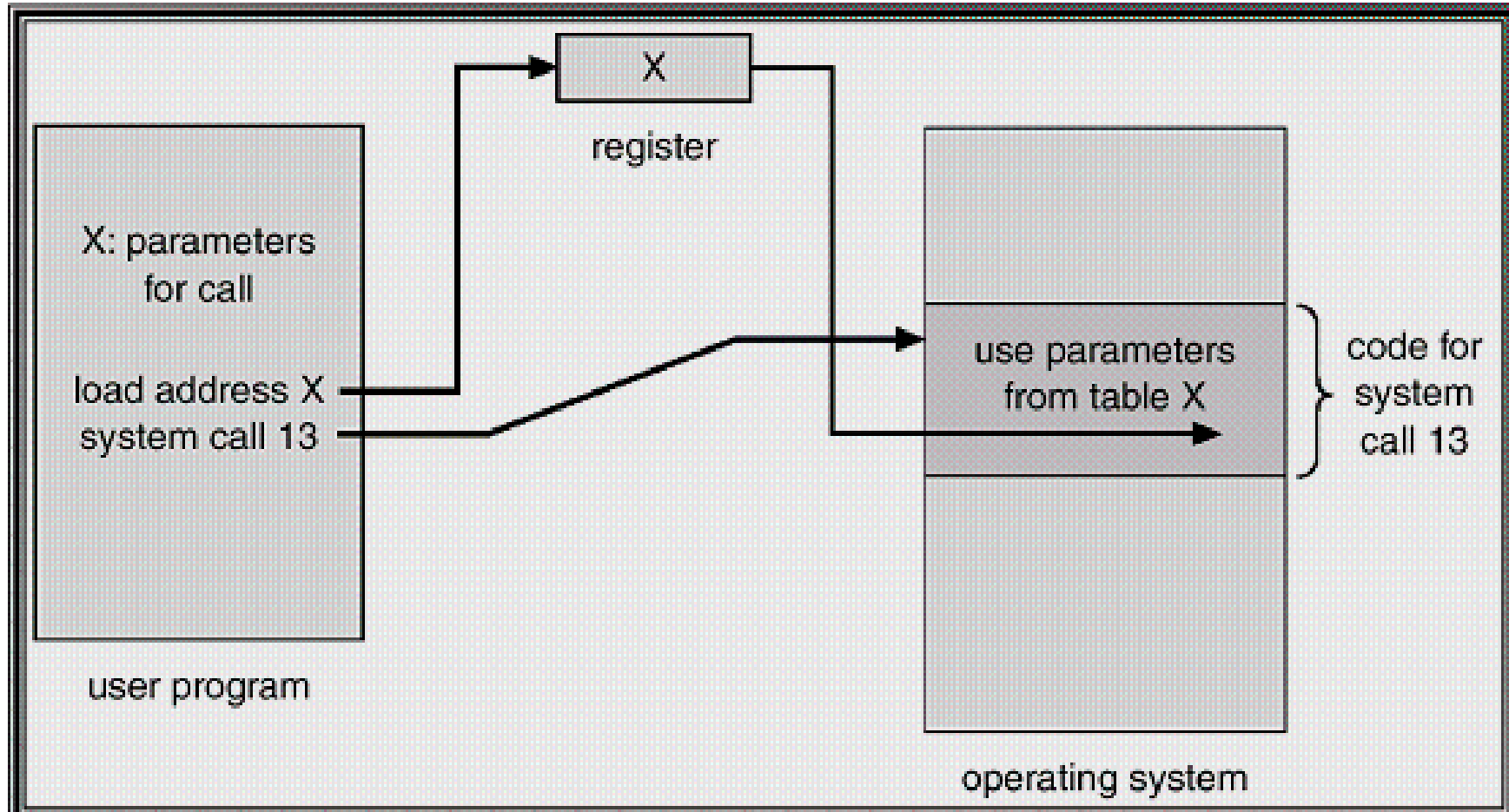


3 είναι οι μέθοδοι για πέρασμα παραμέτρων σε κλήσεις συστήματος

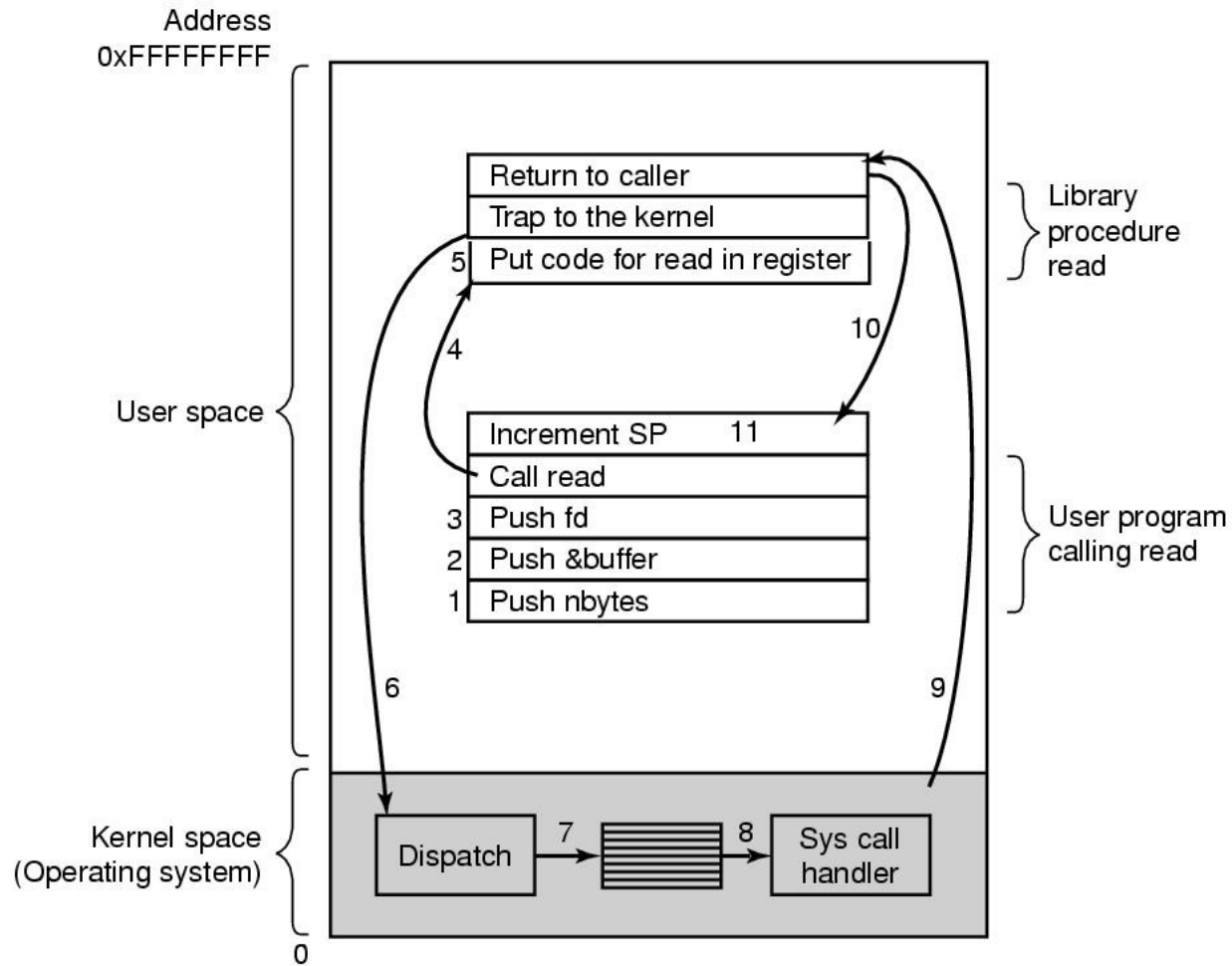
- Πέρασμα παραμέτρων σε καταχωρητές (*registers*).
- Αποθήκευση των παραμέτρων σε έναν πίνακα στη μνήμη, και πέρασμα της διεύθυνσης του πίνακα ως παράμετρο σε καταχωρητή.
- Σπρώξιμο (**Push**) (δηλαδή αποθήκευση) από το πρόγραμμα των παραμέτρων στη στοίβα (**stack**), και ανάκτηση (**pop**) από το ΛΣ των παραμέτρων από τη στοίβα.



Αποθήκευση των παραμέτρων σε έναν πίνακα (κλήσεις συστήματος)



Κλήσεις συστήματος



POSIX

- POSIX (Portable Operating System Interface for Unix) είναι το όνομα μιας σειράς πρότυπων IEEE (IEEE Std 1003.1-1988) που καθορίζουν τις βασικές υπηρεσίες που παρέχει ένα λειτουργικό σύστημα και κυρίως το API των διαθέσιμων κλήσεων συστήματος.
- Δημιουργήθηκε με στόχο την εξασφάλιση της συμβατότητας μεταξύ των ποικίλων λειτουργικών συστημάτων UNIX.
- Υπάρχουν πολλαπλά πρότυπα POSIX.
- Τα προγράμματα που τρέχουν σε έναν υπολογιστή με UNIX να μπορούν να μεταγλωττιστούν σε οποιονδήποτε άλλον υπολογιστή με UNIX χωρίς να χρειάζεται να τροποποιηθεί ο πηγαίος κώδικας
- Υποστηρίζεται από linux, bsd, Solaris, VxWorks, Mac OS X, HPUX, AIX, Minix, QNX, RTEMS, Tru64,



Εύρεση έκδοσης POSIX σε ΛΣ

- FreeBSD

```
root@parsys:~ # uname -a
FreeBSD parsys [REDACTED] 12.1-RELEASE-p10 FreeBSD 12.1-RELEASE-p10 GENERIC amd64
root@parsys:~ # more /usr/include/unistd.h | grep POSIX2_VERSION
#define _POSIX2_VERSION          199212L
root@parsys:~ #
```

- Linux

```
root@bbeuropos:~# uname -a
Linux bbeuropos.[REDACTED] 5.4.51+ #1333 Mon Aug 10 16:38:02 BST 2020 armv6l GNU/Linux
root@bbeuropos:~# more /usr/include/unistd.h | grep POSIX2_VERSION
#define _POSIX2_VERSION __POSIX2_THIS_VERSION
root@bbeuropos:~# more /usr/include/unistd.h | grep POSIX2_THIS_VERSION
# define __POSIX2_THIS_VERSION 200809L
# define __POSIX2_THIS_VERSION 200112L
# define __POSIX2_THIS_VERSION 199506L
# define __POSIX2_THIS_VERSION 199209L
#define _POSIX2_VERSION __POSIX2_THIS_VERSION
#define _POSIX2_C_VERSION __POSIX2_THIS_VERSION
#define _POSIX2_C_BIND __POSIX2_THIS_VERSION
#define _POSIX2_C_DEV __POSIX2_THIS_VERSION
#define _POSIX2_SW_DEV __POSIX2_THIS_VERSION
#define _POSIX2_LOCALEDEF __POSIX2_THIS_VERSION
root@bbeuropos:~#
```

POSIX σε Windows

<http://support.microsoft.com/kb/308259>

- Partial POSIX Support.
- Windows Services for Unix (extra package).

Cygwin (<http://www.cygwin.com>)

- Posix compatible environment.
- A collection of tools which provide a Linux look and feel environment for Windows.
- A DLL (cygwin1.dll) which acts as a Linux API layer providing substantial Linux API functionality.



Κυριότερες κλήσεις συστήματος για διαχείριση διεργασιών (1/2)

Process management

Call	Description
<code>pid = fork()</code>	Create a child process identical to the parent
<code>pid = waitpid(pid, &statloc, options)</code>	Wait for a child to terminate
<code>s = execve(name, argv, environp)</code>	Replace a process' core image
<code>exit(status)</code>	Terminate process execution and return status



Κυριότερες κλήσεις συστήματος για διαχείριση αρχείων (2/2)

File management

Call	Description
<code>fd = open(file, how, ...)</code>	Open a file for reading, writing, or both
<code>s = close(fd)</code>	Close an open file
<code>n = read(fd, buffer, nbytes)</code>	Read data from a file into a buffer
<code>n = write(fd, buffer, nbytes)</code>	Write data from a buffer into a file
<code>position = lseek(fd, offset, whence)</code>	Move the file pointer
<code>s = stat(name, &buf)</code>	Get a file's status information

Call	Description
<code>s = mkdir(name, mode)</code>	Create a new directory
<code>s = rmdir(name)</code>	Remove an empty directory
<code>s = link(name1, name2)</code>	Create a new entry, name2, pointing to name1
<code>s = unlink(name)</code>	Remove a directory entry
<code>s = mount(special, name, flag)</code>	Mount a file system
<code>s = umount(special)</code>	Unmount a file system



Κυριότερες κλήσεις συστήματος χωρίς κατηγορία

Call	Description
<code>s = chdir(dirname)</code>	Change the working directory
<code>s = chmod(name, mode)</code>	Change a file's protection bits
<code>s = kill(pid, signal)</code>	Send a signal to a process
<code>seconds = time(&seconds)</code>	Get the elapsed time since Jan. 1, 1970



Ένας απλός φλοιός

```
#define TRUE 1

while (TRUE) {
    type_prompt( );
    read_command(command, parameters);

    if (fork() != 0) {
        /* Parent code. */
        waitpid(-1, &status, 0);
    } else {
        /* Child code. */
        execve(command, parameters, 0);
    }
}
```

/ repeat forever */*
/ display prompt on the screen */*
/ read input from terminal */*

/ fork off child process */*

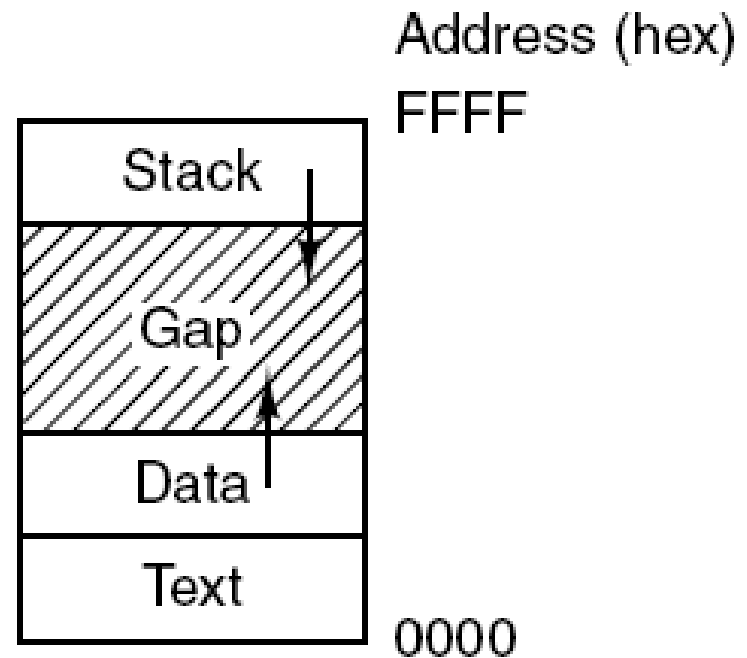
/ wait for child to exit */*

/ execute command */*



Κάθε διεργασία έχει 3 περιοχές μνήμης

- Κώδικα (text) {read only}.
- Δεδομένων (data).
- Σωρού (stack).

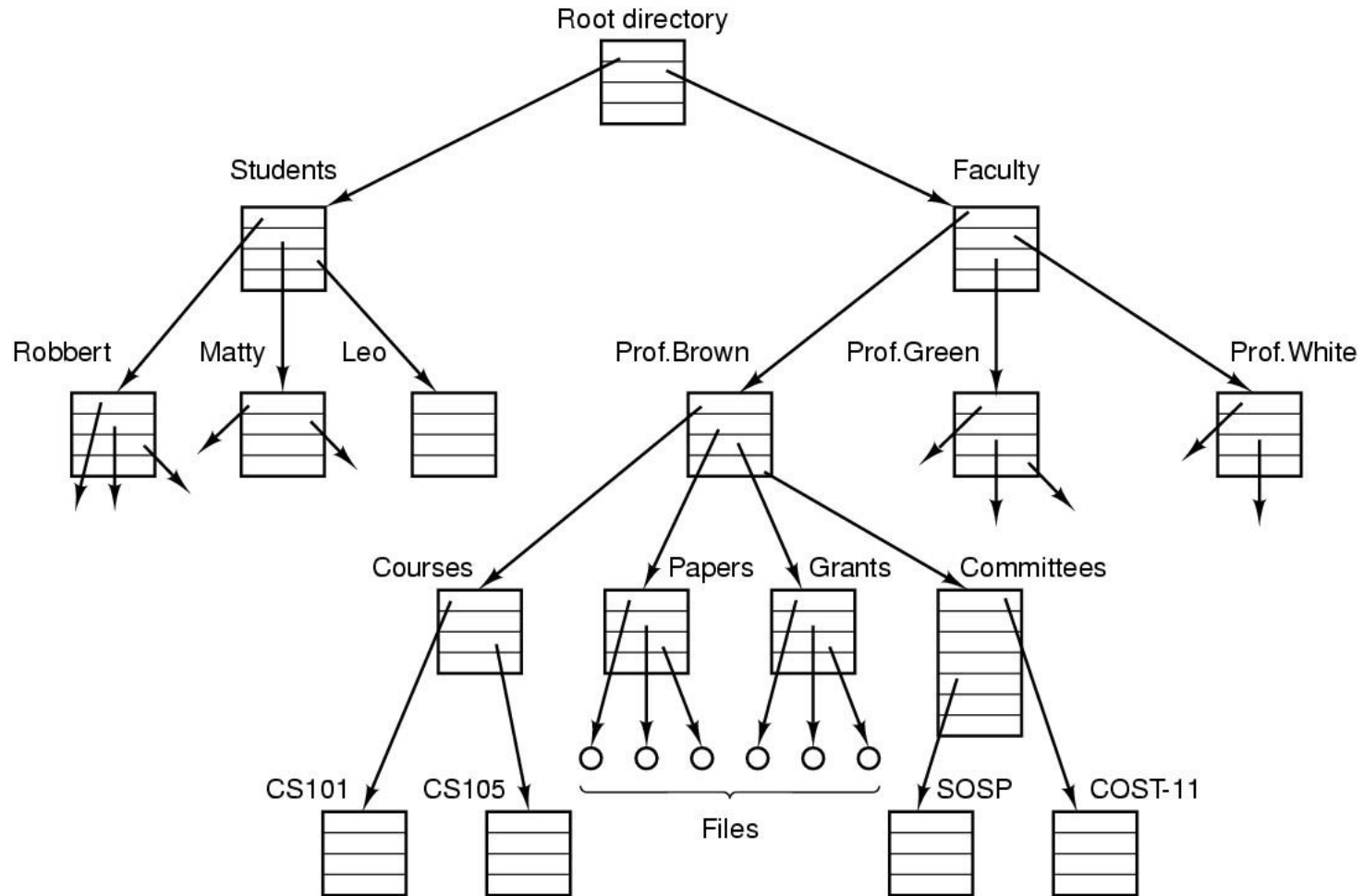


Τα Windows έχουν διαφορετικές κλήσεις συστήματος από UNIX

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time



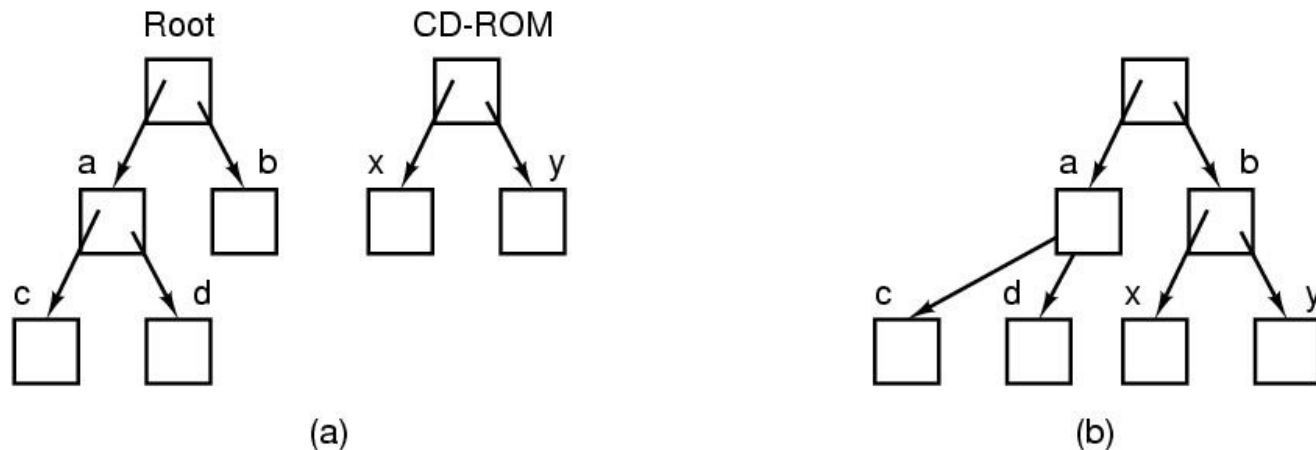
Ιεραρχία Συστήματος Αρχείων



Προσαρτημένα συστήματα

(a) Πριν την προσάρτηση τα αρχεία που βρίσκονται στο CD-ROM δεν είναι διαθέσιμα

(b) Μετά την προσάρτηση, γίνονται τμήμα της ιεραρχίας των αρχείων.



Προσαρτημένα συστήματα (παράδειγμα)

- (a) Άδειος κατάλογος.
- (b) Εντολή προσάρτησης άλλου συστήματος αρχείων στον κατάλογο.
- (c) Ο κατάλογος έχει αρχεία.
- (d) Εντολή απο-προσάρτησης.
- (e) Άδειος κατάλογος.

```
bigb5# ls -la /mnt
total 4
drwxr-xr-x  2 root  wheel  512 Jul 26 2010 .
drwxr-xr-x 23 root  wheel  512 Dec 19 16:12 ..
bigb5# mount -t ufs /dev/da0 /mnt
bigb5# ls -la /mnt
total 8
drwxr-xr-x  4 root  wheel  512 Jan 18 20:37 .
drwxr-xr-x 23 root  wheel  512 Dec 19 16:12 ..
drwxrwxr-x  2 root  operator 512 Jan 17 19:28 .snap
drwxr-xr-x 15 root  wheel  512 Jan 26 05:05 2ndbackup
bigb5# umount /mnt
bigb5# ls -la /mnt
total 4
drwxr-xr-x  2 root  wheel  512 Jul 26 2010 .
drwxr-xr-x 23 root  wheel  512 Dec 19 16:12 ..
bigb5#
```



Προγράμματα Συστήματος

- Τα προγράμματα συστήματος παρέχουν ένα βολικό περιβάλλον για ανάπτυξη και εκτέλεση προγραμμάτων. Ανάλογα με τον ρόλο τους χωρίζονται σε:
 - Χειρισμός και τροποποίηση αρχείων.
 - Πληροφορίες κατάστασης συστήματος.
 - Φόρτωση και εκτέλεση προγράμματος.
 - Υποστήριξη γλωσσών προγραμματισμού.
 - Επικοινωνίες.
 - Προγράμματα εφαρμογών.
- Η άποψη των χρηστών για το ΛΣ καθορίζεται από τα προγράμματα συστήματος σε συνδυασμό με τον διερμηνέα εντολών (και όχι τόσο από το ΛΣ καθ' αυτό - π.χ. τα Windows όπου ο χρήστης δε «βλέπει» το πραγματικό ΛΣ παρά μόνο το γραφικό περιβάλλον εντολών).



Δομές ΛΣ: (1)

Μονολιθικά συστήματα (1/2)

- Είναι μια προσέγγιση χωρίς καμιά δομή.
- Το ΛΣ αποτελείται από ουσιαστικά ένα μεγάλο κομμάτι κώδικα.
- Το Λ.Σ. είναι μια συλλογή από διαδικασίες, κάθε μια από τις οποίες μπορεί να καλέσει οποιαδήποτε άλλη, όταν τη χρειαστεί.
- Η επικοινωνία μεταξύ των διαδικασιών γίνεται με παραμέτρους.
- Κάθε διαδικασία είναι ορατή σε οποιαδήποτε άλλη.
- Παραδείγματα μονολιθικών Λ.Σ : OS/360, VMS, Linux.



Δομές ΛΣ: (1)

Μονολιθικά συστήματα (2/2)

- Μονολιθικά συστήματα – βασική δομή:
 - Ένα κύριο πρόγραμμα που το οποίο επικαλείται την αιτούμενη διαδικασία παροχής υπηρεσιών.
 - Ένα σύνολο των διαδικασιών υπηρεσιών που πραγματοποιούν τις κλήσεις συστήματος.
 - Ένα σύνολο βοηθητικών διαδικασιών που βοηθούν τις διαδικασίες υπηρεσιών.



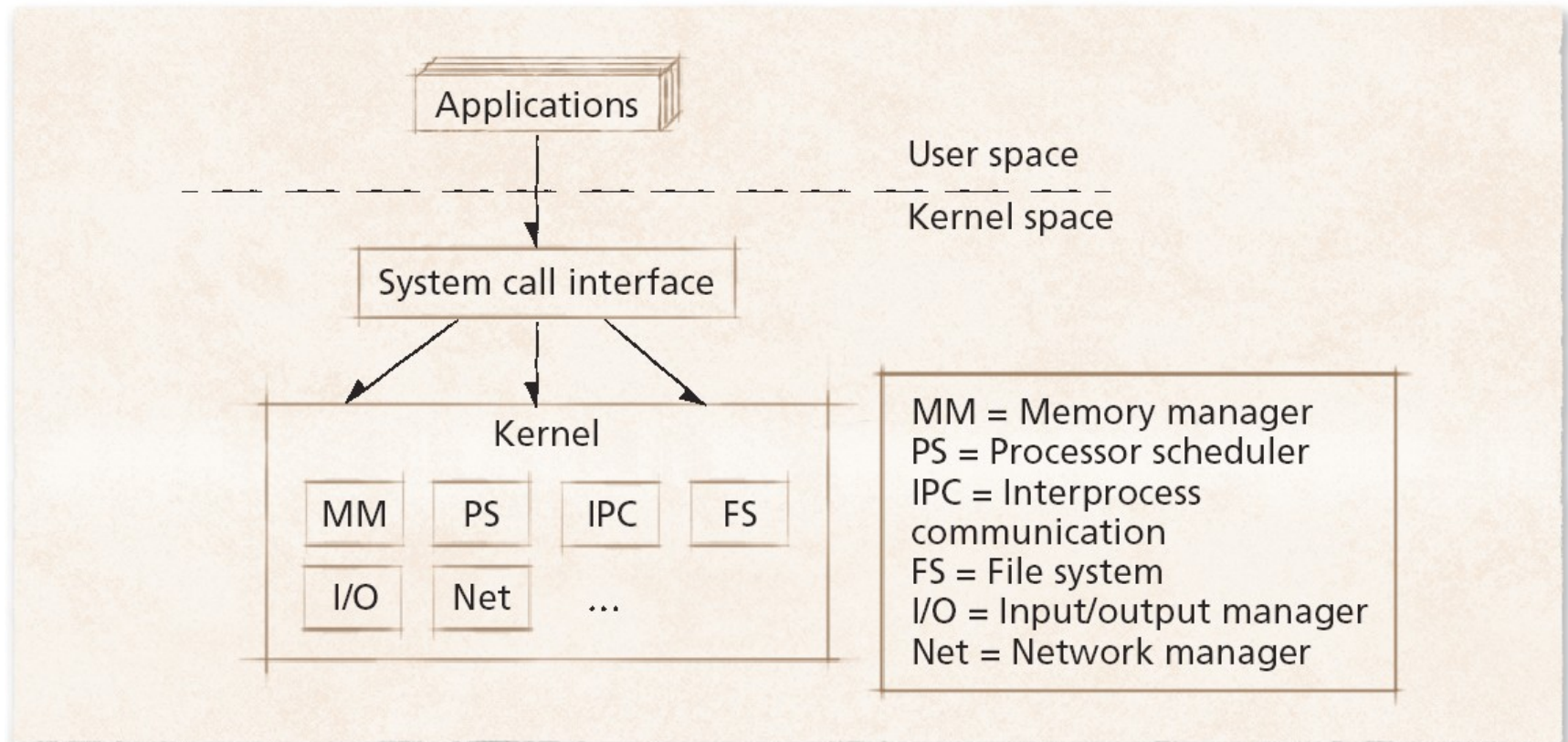
Δομές ΛΣ: (1) Μονολιθικά συστήματα (κλήση συστήματος)

- Το πρόγραμμα του χρήστη δημιουργεί μια παγίδα στον πυρήνα (εκτελείται μια ειδική εντολή η κλήση πυρήνα – *kernel call*).
- Το Λ.Σ. προσδιορίζει τον αριθμό εξυπηρέτησης που ζητήθηκε.
- Το Λ.Σ. εντοπίζει και καλεί τη διαδικασία εξυπηρέτησης.
- Ο έλεγχος επιστρέφεται στο πρόγραμμα του χρήστη.



Δομές ΛΣ: (1)

Μονολιθικά συστήματα



Δομές ΛΣ: (1)

Μονολιθικά συστήματα (*kernel*)

- Δρομολογητής διεργασιών (process scheduler).
- Διαχειριστής μνήμης (memory manager).
- Διαχειριστής συσκευών E/E (I/O manager).
- Διαχειριστής διαδιεργασιακής επικοινωνίας (interprocessor communication (IPC) manager).
- Διαχειριστής συστήματος αρχείων (file system manager).



Η Δομή του UNIX

- UNIX – περιορισμένο από τις δυνατότητες του υλικού, αρχικά το ΛΣ UNIX είχε περιορισμένη δόμηση Το ΛΣ Unix αποτελείται από δύο διαχωρίσιμα τμήματα:
 - Προγράμματα συστήματος.
 - Τον πυρήνα (kernel):
 - Αποτελείται από οτιδήποτε υπάρχει “κάτω” από την προσαρμογή των κλήσεων συστήματος και “πάνω” από το υλικό.
 - Παρέχει τις λειτουργίες συστήματος αρχείων, διαχείρισης μνήμης, χρονοπρογραμματισμού ΚΜΕ και άλλες λειτουργίες ενός ΛΣ (μεγάλος αριθμός λειτουργιών για ένα επίπεδο).

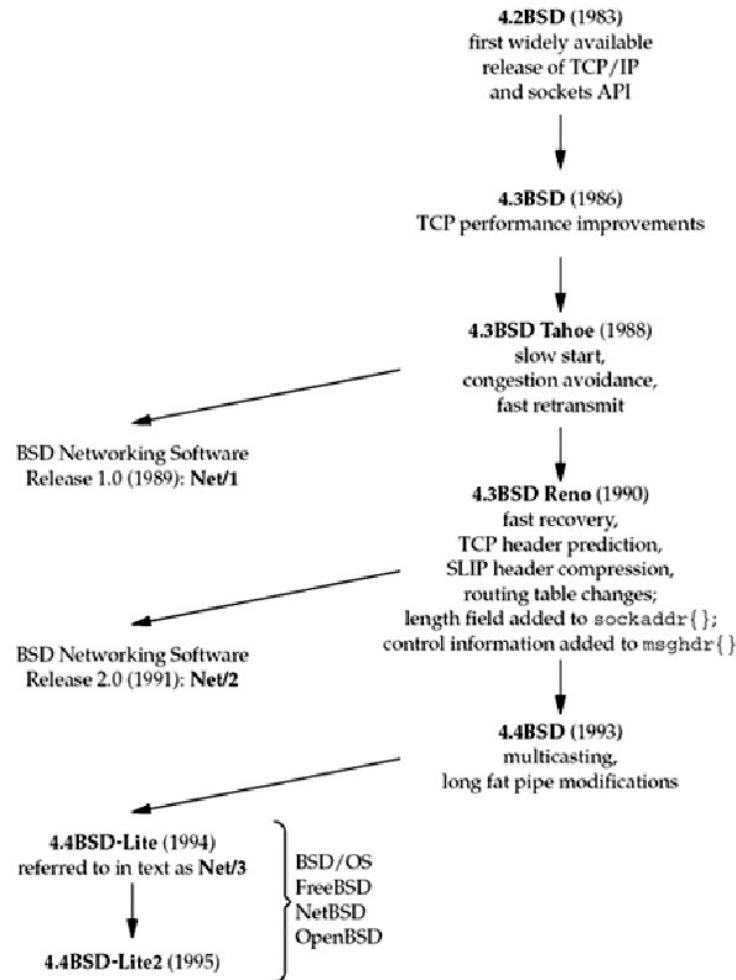


Πυρήνας και οδηγοί συσκευών

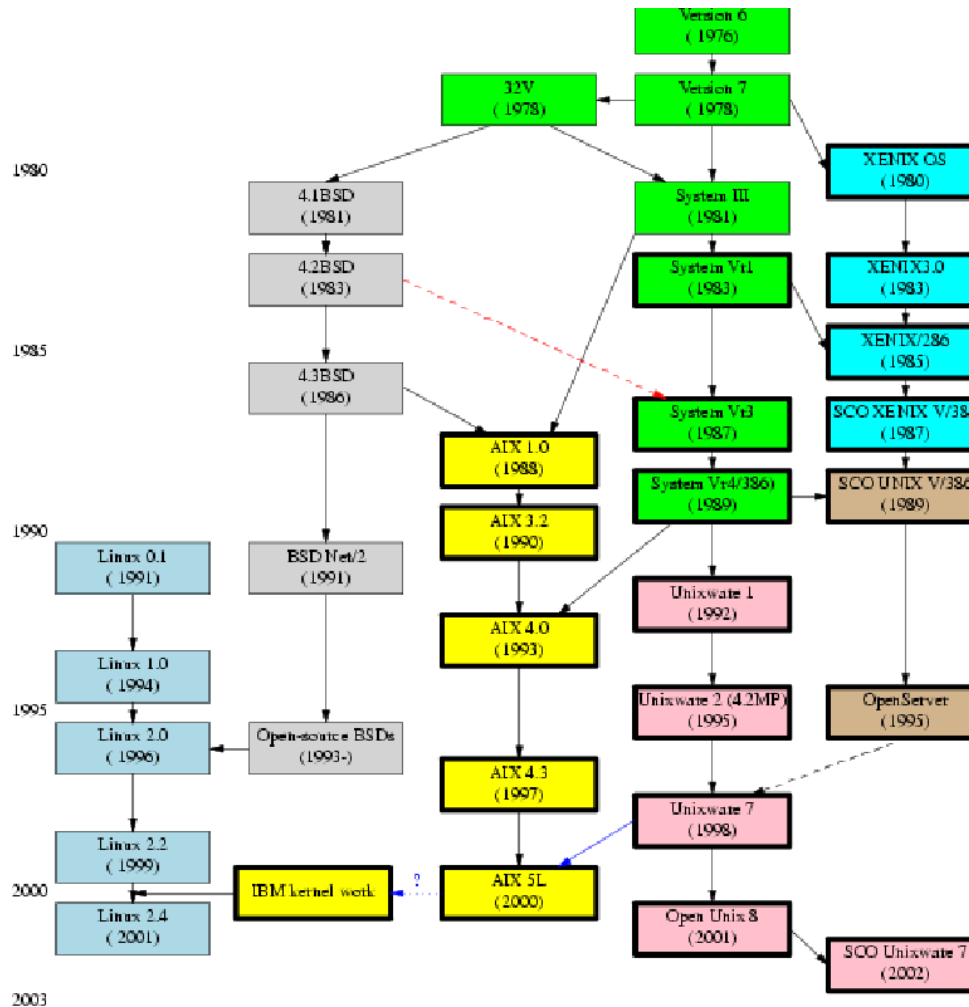
- «Ένας **πυρήνας** στη παραδοσιακή ορολογία των λειτουργικών συστημάτων, είναι ένας μικρός πυρήνας λογισμικού που διαθέτει μόνο τις άκρως απαραίτητες παροχές για την εφαρμογή των πρόσθετων υπηρεσιών του λειτουργικού συστήματος».
- Οδηγοί συσκευών
 - Προγράμματα που παρέχουν μια απλή, συνεπή διασύνδεση στις συσκευές E/E.
 - Συνήθως αποτελούν μέρος του πυρήνα.



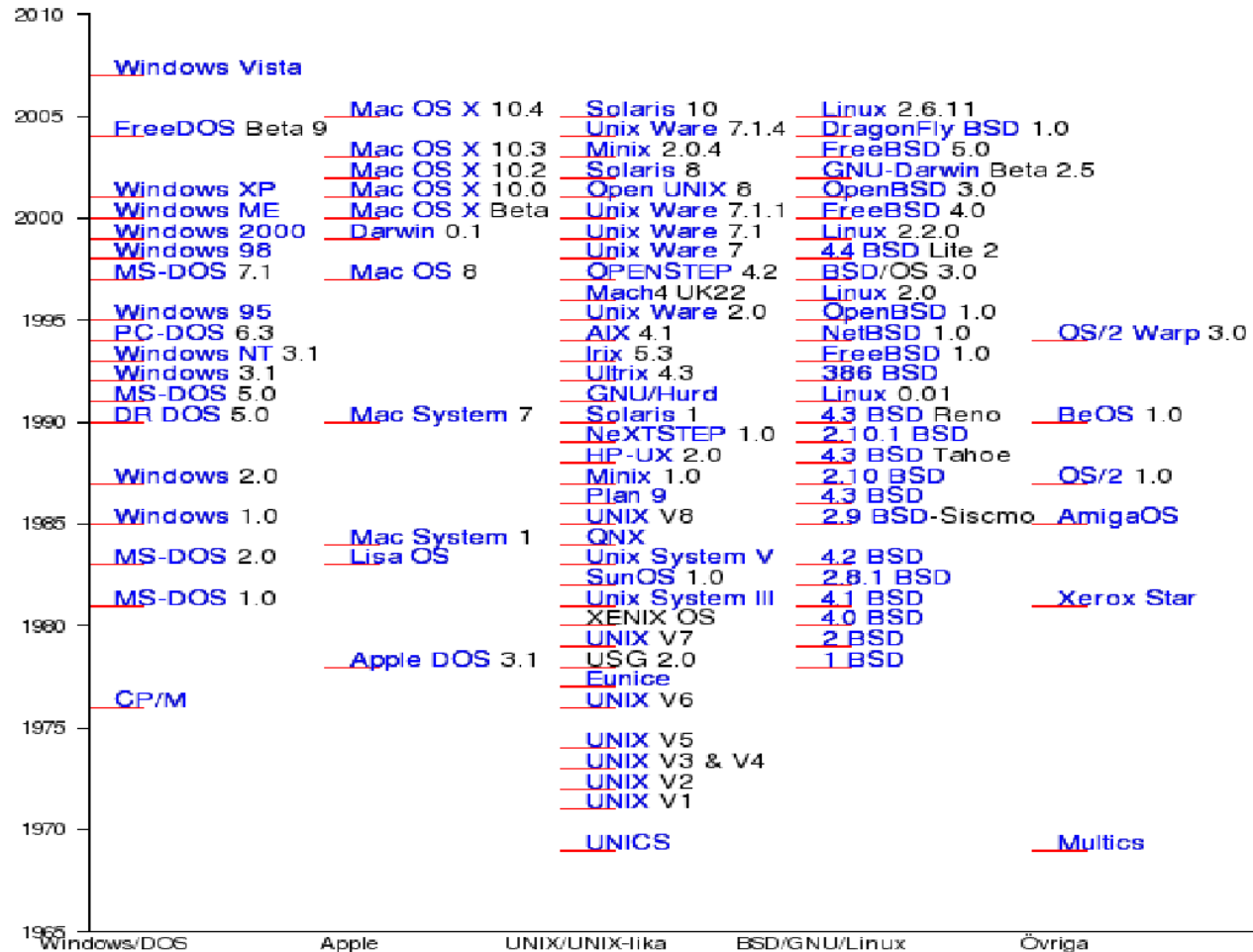
Η Ιστορία του BSD



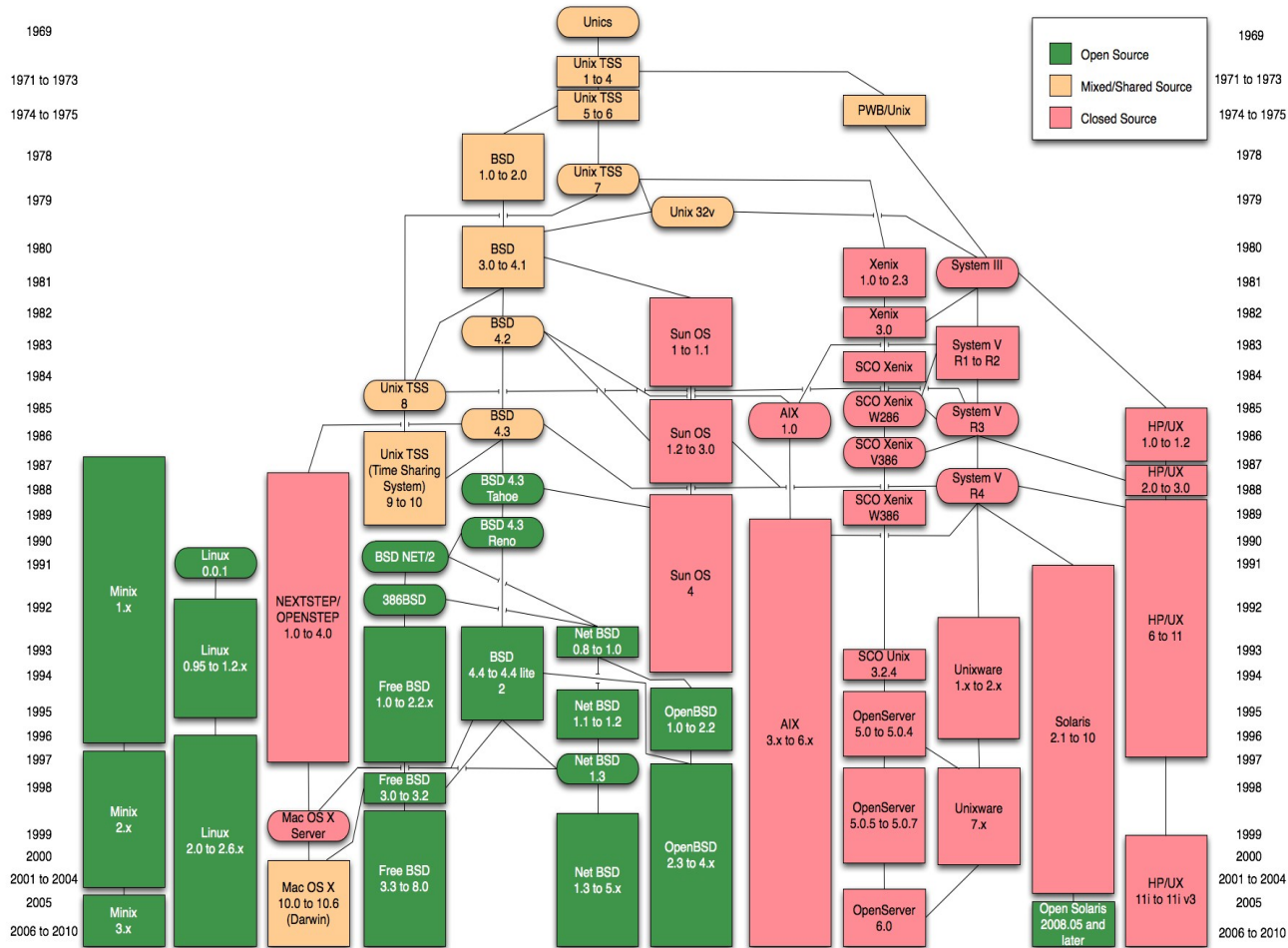
UNIX timeline



Χρήση των λειτουργικών συστημάτων (1/2)



Χρήση των λειτουργικών συστημάτων (2/2)



4.4 BSD

- Βασικές παροχές πυρήνα: χρονόμετρο και χειρισμός του ρολογιού συστήματος, διαχείριση περιγραφείς και διαχείριση διαδικασιών.
- Υποστήριξη διαχείρισης μνήμης: σελιδοποίηση και εναλλαγή.
- Γενικές διεπαφές συστήματος: E/E, έλεγχος και πολυπλεξία των λειτουργιών που γίνονται στους περιγραφείς.
- Το σύστημα αρχείων: αρχεία, καταλόγους, μετάφραση ονόματος διαδρομής, κλείδωμα αρχείων και E/E διαχείριση buffer.
- Υποστήριξη διαχείρισης τερματικού: ο οδηγός διεπαφής τερματικού και κλάδου τερματικού.
- Επικοινωνία ενδο-διεργασιών: υποδοχείς.
- Υποστήριξη επικοινωνίας δικτύου: πρωτόκολλα επικοινωνίας και γενικές παροχές δικτύου, όπως δρομολόγηση.



Το 80% του κώδικα στο 4.4 BSD είναι κώδικας ανεξάρτητης αρχιτεκτονικής

Table 2-1. Machine-independent software in the 4.4BSD kernel

Category	Lines of code	Percentage of kernel
headers	9,393	4.6
initialization	1,107	0.6
kernel facilities	8,793	4.4
generic interfaces	4,782	2.4
interprocess communication	4,540	2.2
terminal handling	3,911	1.9
virtual memory	11,813	5.8
vnode management	7,954	3.9
filesystem naming	6,550	3.2
fast filestore	4,365	2.2
log-structure filestore	4,337	2.1
memory-based filestore	645	0.3
cd9660 filesystem	4,177	2.1
miscellaneous filesystems (10)	12,695	6.3
network filesystem	17,199	8.5
network communication	8,630	4.3
internet protocols	11,984	5.9
ISO protocols	23,924	11.8
X.25 protocols	10,626	5.3
XNS protocols	5,192	2.6
total machine independent	162,617	80.4



Το 20% του κώδικα στο 4.4 BSD είναι κώδικας για συγκεκριμένες αρχιτεκτονικές

Table 2-2. Machine-dependent software for the HP300 in the 4.4BSD kernel

Category	Lines of code	Percentage of kernel
machine dependent headers	1,562	0.8
device driver headers	3,495	1.7
device driver source	17,506	8.7
virtual memory	3,087	1.5
other machine dependent	6,287	3.1
routines in assembly language	3,014	1.5
HP/UX compatibility	4,683	2.3
total machine dependent	39,634	19.6



Η διαδικασία bootstrap του 4.4BSD

Οι μηχανισμοί **Bootstrapping** χρησιμοποιούνται για να εκκινήσουν το τρέχων σύστημα. Αρχικά, ο πυρήνας 4.4 του BSD πρέπει να **φορτωθεί στη κύρια μνήμη** του επεξεργαστή. Μόλις φορτωθεί, πρέπει να περάσει **μια φάση αρχικοποίησης** προκειμένου να θέσει το υλικό σε μια γνωστή κατάσταση. Έπειτα, ο πυρήνας πρέπει να κάνει μια **αυτόματη ρύθμιση**, μια διεργασία που βρίσκει και ρυθμίζει τα περιφερειακά που είναι συνδεδεμένα με τον επεξεργαστή.

Το σύστημα ξεκινάει σε **κατάσταση λειτουργίας ενός χρήστη**, ενώ ένα σενάριο (*script*) εκκίνησης ελέγχει τους δίσκους και ξεκινά τον έλεγχο της λογιστικής και των ποσοστώσεων.

Τέλος, **το σενάριο εκκίνησης** ξεκινά τις γενικές υπηρεσίες του συστήματος και φέρνει το σύστημα σε **πλήρη λειτουργία πολλών χρηστών**.



4.4BSD Λειτουργία

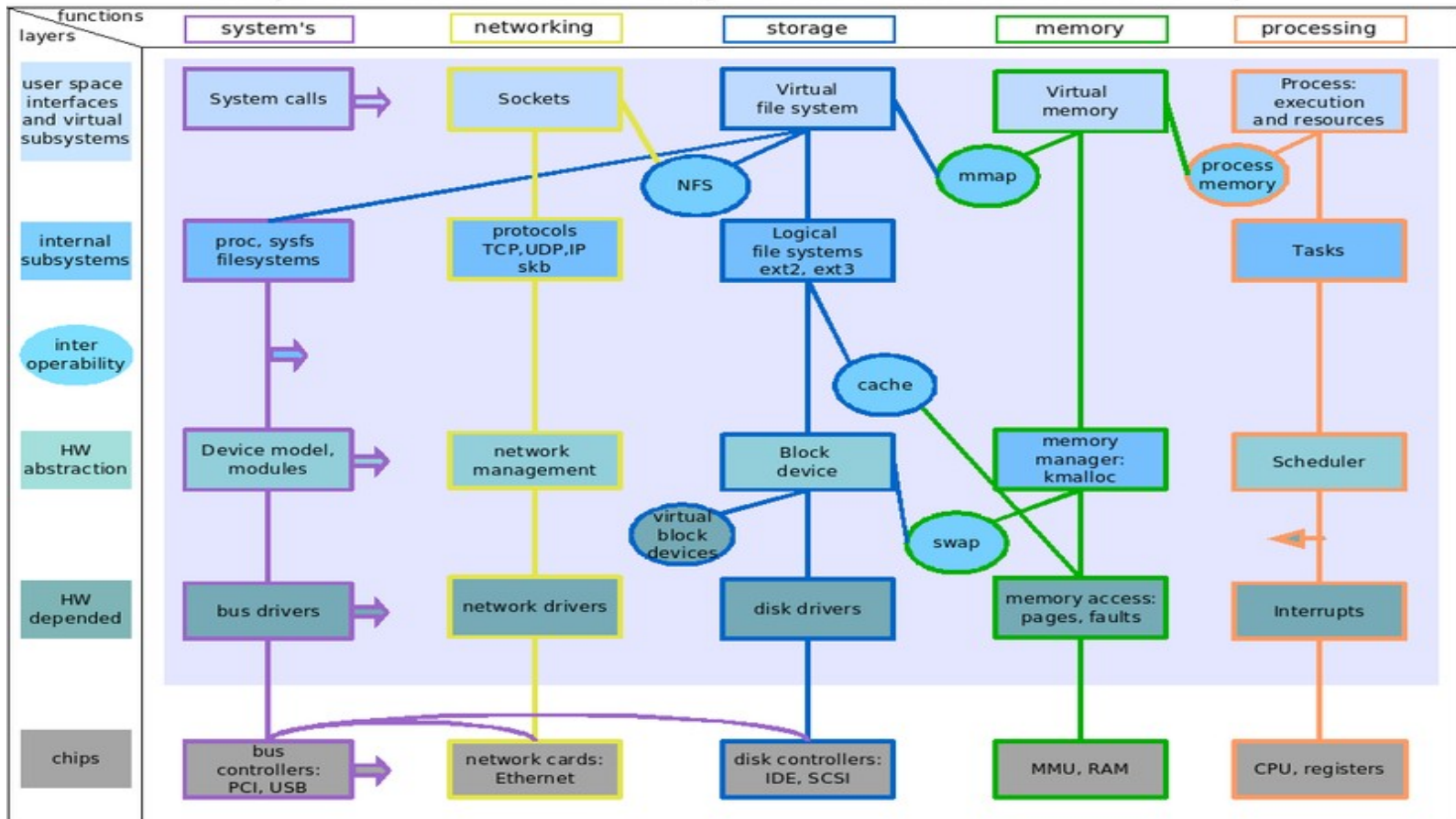
Συστήματος Πολλαπλών Χρηστών

- Κατά τη διάρκεια λειτουργίας πολλών χρηστών, **οι διεργασίες περιμένουν για τις αιτήσεις συνδέσεων στις γραμμές τερματικού και στις θύρες δικτύου που έχουν διαμορφωθεί για την πρόσβαση των χρηστών.** Όταν ένα αίτημα σύνδεσης ανιχνεύεται, **μια διαδικασία σύνδεσης γεννιέται και γίνεται η επιβεβαίωση των χρηστών.**
- Όταν η επικύρωση σύνδεση είναι επιτυχής, **ένα κέλυφος σύνδεσης δημιουργείται** από το οποίο ο χρήστης μπορεί να τρέξει πρόσθετες διαδικασίες.



Η ιεραρχία των συστατικών του πυρήνα του Linux

Simplified Linux kernel diagram in form of a matrix map



Designed with OpenOffice.org by (cc) (by-nc-sa) Constantine Shulyupin, www.linuxdriver.co.il



Δομές ΛΣ: (2) Πολυεπίπεδα (ή στρωματοποιημένα) συστήματα (1/4)

- Η στρωματοποιημένη αρχιτεκτονική προσπαθεί να βελτιώσει το σχεδιασμό των μονολιθικών πυρήνων ομαδοποιώντας συστατικά που υλοποιούν παρόμοιες λειτουργίες σε επίπεδα.
- Κάθε επίπεδο επικοινωνεί μόνον με τα γειτονικά του (επάνω και κάτω).
- Οι απαιτήσεις των διεργασιών διαπερνούν αρκετά επίπεδα πριν ολοκληρωθούν.
- Η ρυθμοαπόδοση (throughput) μπορεί να είναι μικρότερη από τα Λ.Σ. με τους μονολιθικούς πυρήνες.
- Απαιτούνται επιπλέον μέθοδοι για τη μεταβίβαση και τον έλεγχο των δεδομένων.



Δομές ΛΣ: (2) Πολυεπίπεδα (ή στρωματοποιημένα) συστήματα (2/4)

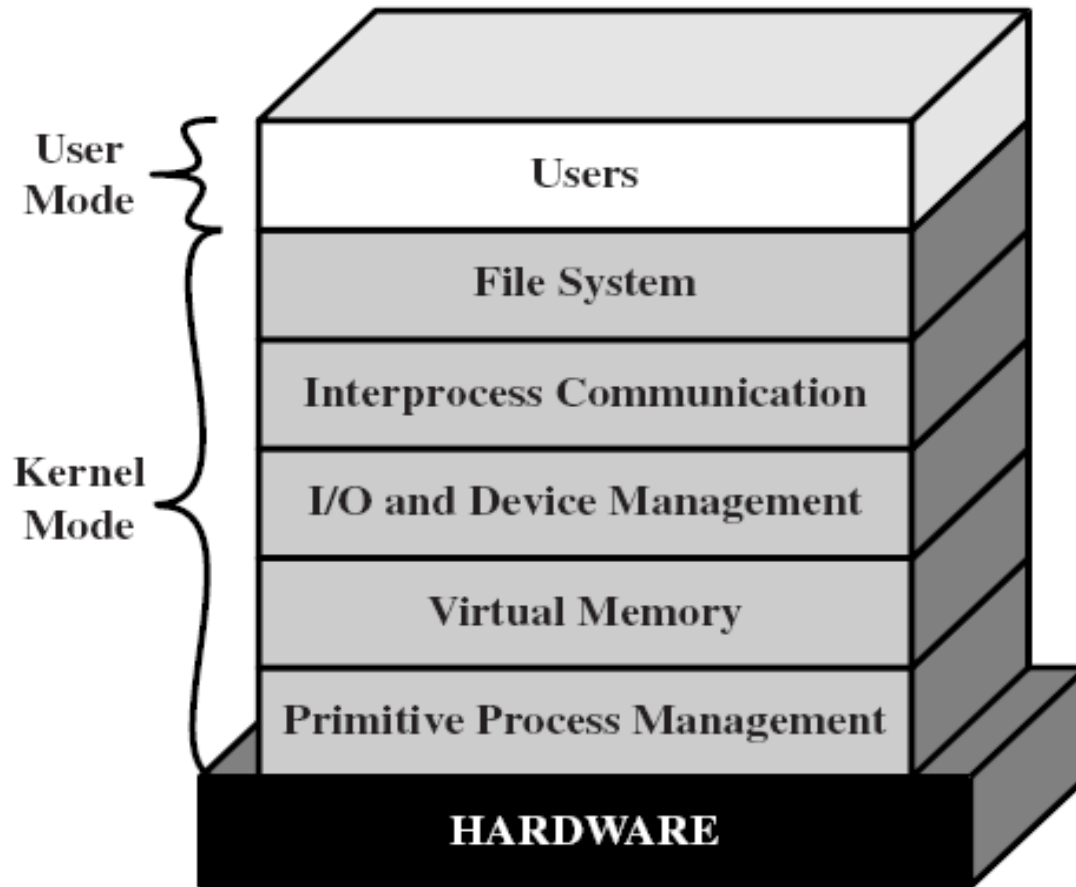
Επίπεδο	Συνάρτηση
5	Ο τελεστής
4	Προγράμματα χρήστη
3	Διαχείριση Ε/Ε
2	Επικοινωνία τελεστή – διεργασίας
1	Διαχείριση μνήμης
0	Κατανομή επεξεργαστή και πολυπρογραμματισμός



Δομές ΛΣ: (2) Πολυεπίπεδα (ή στρωματοποιημένα) συστήματα (3/4)



Δομές ΛΣ: (2) Πολυεπίπεδα (ή στρωματοποιημένα) συστήματα (4/4)

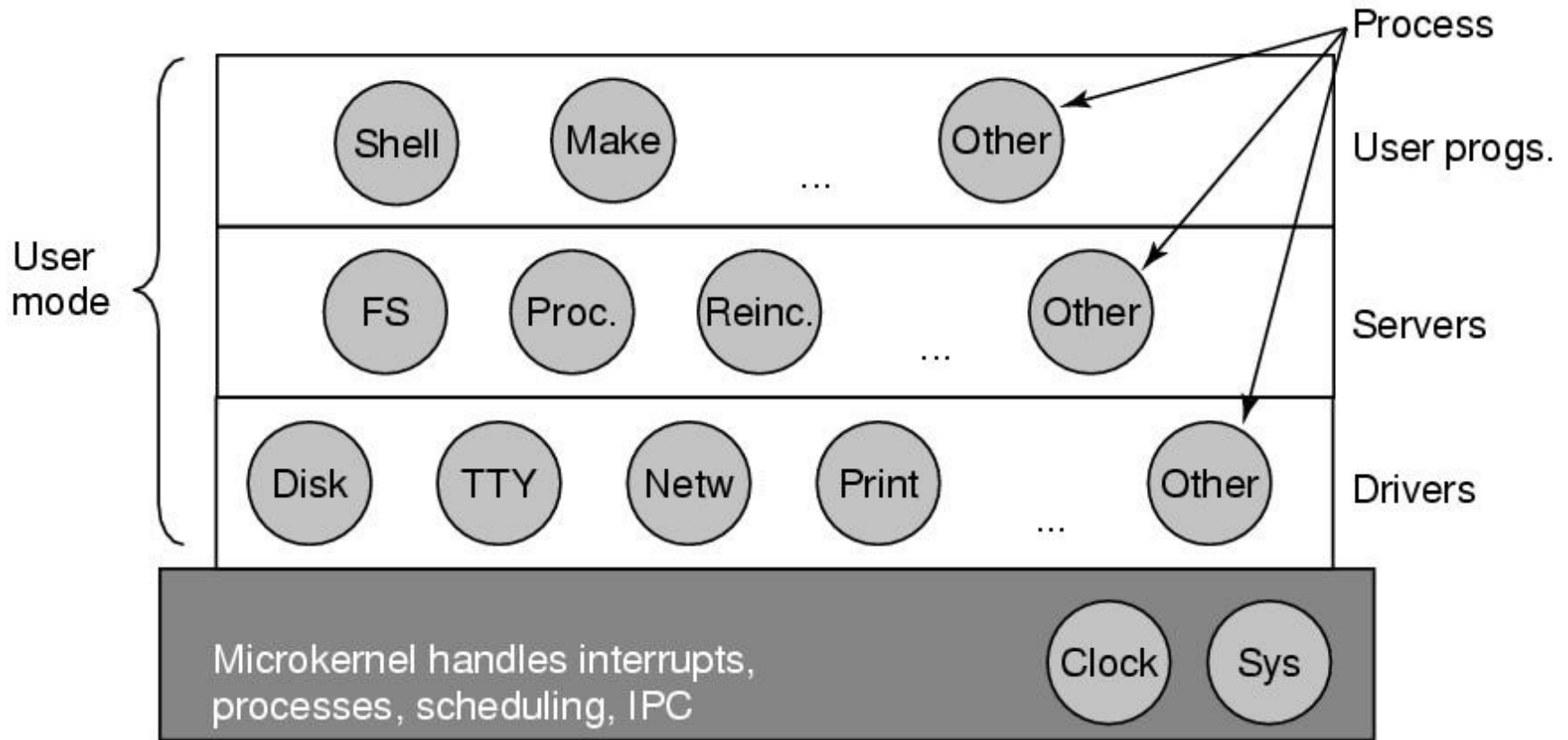


Δομές ΛΣ: (3) Μικρο-πυρήνες

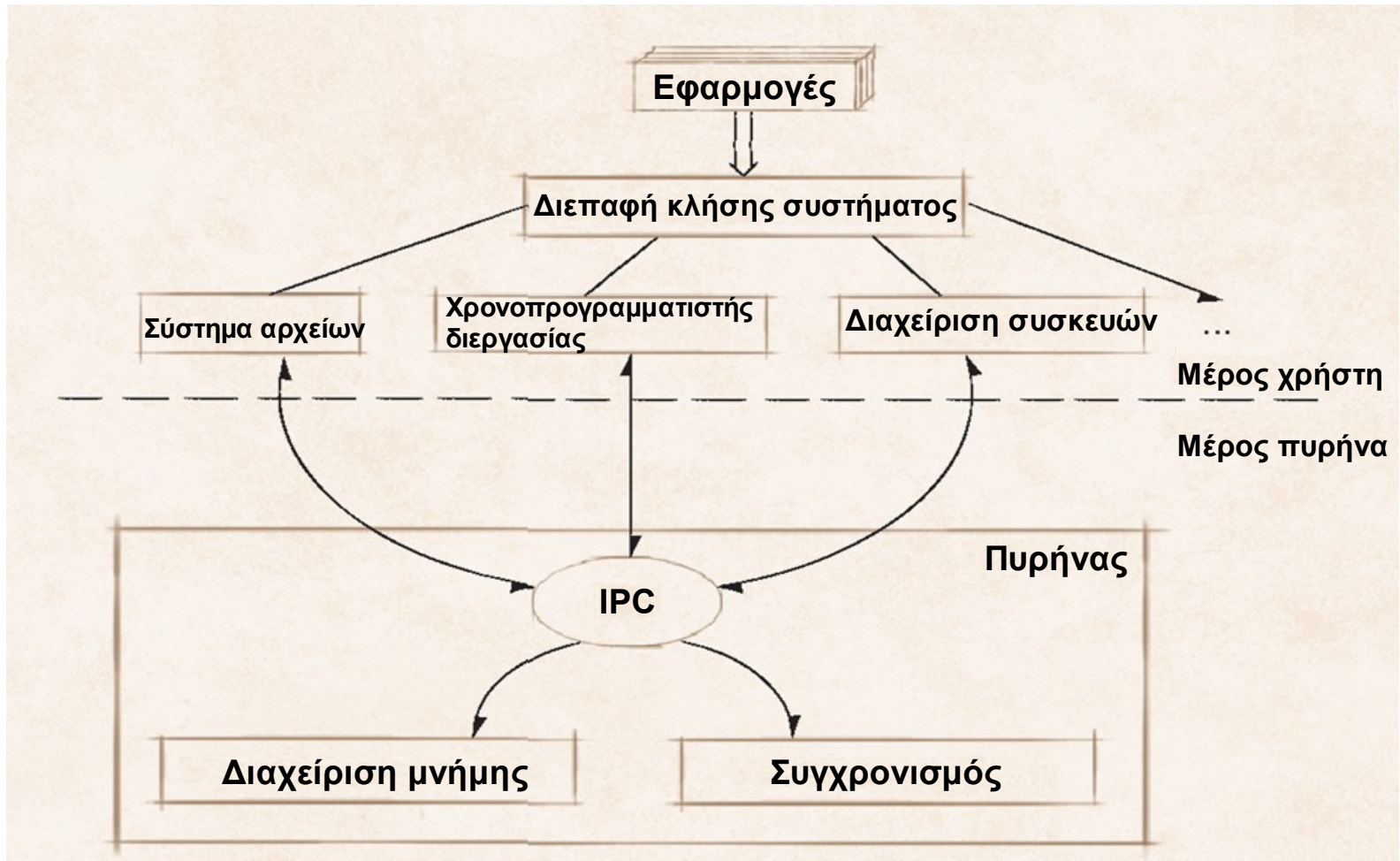
- Ένας μονολιθικός πυρήνας περιλαμβάνει τη δρομολόγηση, το σύστημα αρχείων, τη δικτύωση, τους οδηγούς συσκευής, τη διαχείριση μνήμης κ.α. Υλοποιείται ως μια μοναδική διεργασία και όλα τα στοιχεία διαμοιράζονται τον ίδιο χώρο διευθύνσεων.
- Η αρχιτεκτονική μικροπυρήνα αναθέτει λίγες λειτουργίες στον πυρήνα και **τις υπόλοιπες τις αναθέτει σε εξυπηρετητές που εκτελούνται σε κατάσταση χρήστη**. Η διεργασία του χρήστη (client process) στέλνει την απαίτηση στη διεργασία εξυπηρετητή (server process) η οποία επιτελεί τη διεργασία και επιστρέφει την απάντηση. Ο μικροπυρήνας διαχειρίζεται την επικοινωνία μεταξύ clients και servers.



Δομές ΛΣ: (3) Μικρο-πυρήνες (1/2)



Δομές ΛΣ: (3) Μικρο-πυρήνες (2/2)



Δομές ΛΣ: (4)

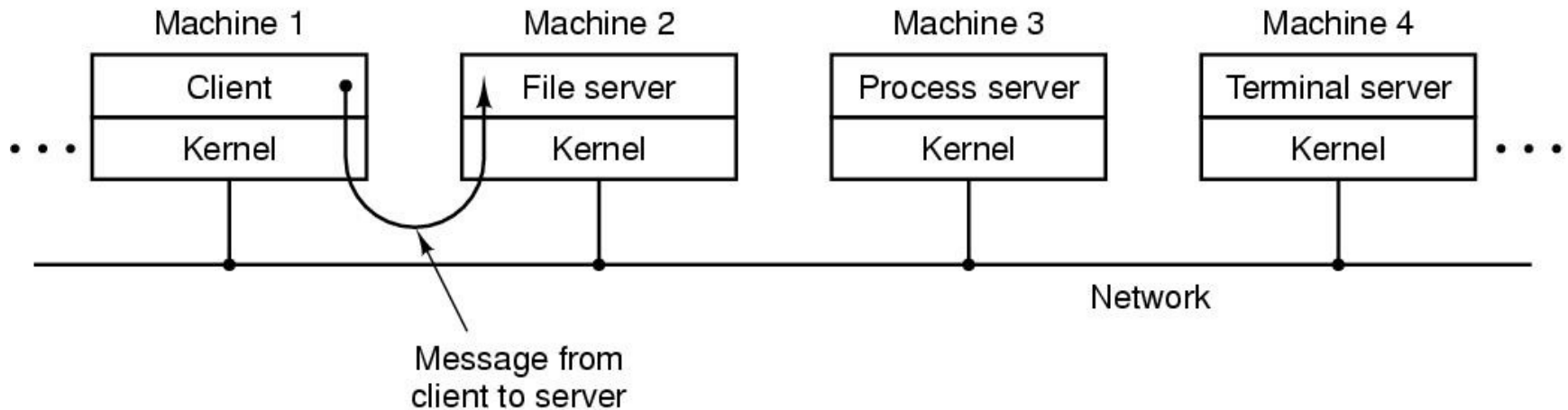
Πελάτη Διακομιστή (1/3)

- Το σχήμα της τοποθέτησης λογισμικού πάνω από τον πυρήνα ώστε να χειρίζεται τη διαδικασία client-server δεν είναι απόλυτα ρεαλιστικό.
- Κάποιες λειτουργίες του χρήστη είναι δύσκολο, ακόμη και ακατόρθωτο να πραγματοποιηθούν από το χώρο προγραμμάτων του χρήστη.
- Τρόποι επίλυσης :
 - Οι κρίσιμες διεργασίες του εξυπηρέτη (drivers I/O) να τρέχουν σε κατάσταση πυρήνα με πλήρη πρόσβαση στο υλικό αλλά να συνεχίσουν να επικοινωνούν με τις άλλες διεργασίες.
 - Να εμφυτευθεί ένας στοιχειώδης μηχανισμός στον πυρήνα αλλά η πολιτική αποφάσεων να παραμείνει στους εξυπηρετητές.



Δομές ΛΣ: (4)

Πελάτη Διακομιστή (2/3)



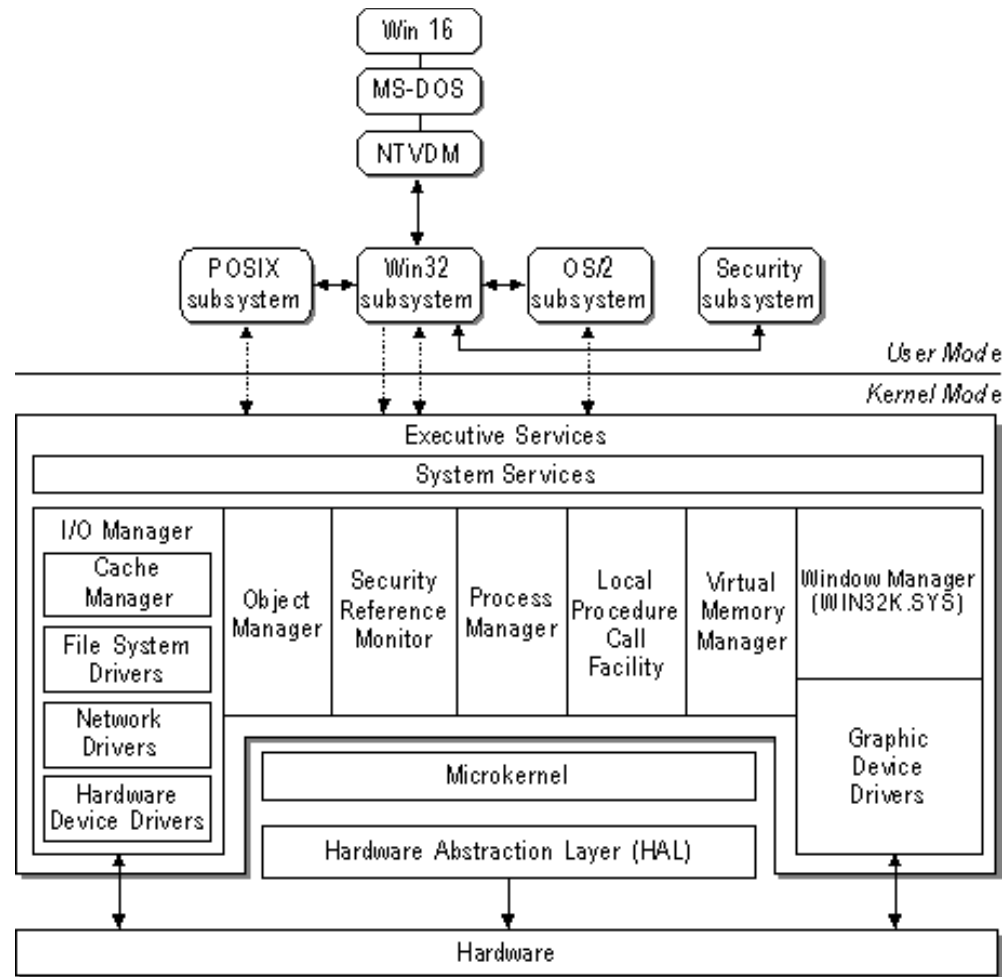
Δομές ΛΣ: (4)

Πελάτη Διακομιστή (3/3)

- Τα Windows χρησιμοποιούν το μοντέλο της πελατειακής δομής για την υλοποίηση των υπηρεσιών που προσφέρουν.
- Οι υπηρεσίες σε επίπεδο χρήστη υλοποιούνται ως διεργασίες που επικοινωνούν μεταξύ τους με το πρωτόκολλο RPC.
- Όταν μία εφαρμογή (client) χρειάζεται μία υπηρεσία, ένα μήνυμα στέλνεται μέσω του Executive σε κάποιον εξυπηρετητή (server) ο οποίος εκτελεί τις απαραίτητες εργασίες και επιστρέφει το ζητούμενο αποτέλεσμα στην εφαρμογή, πάλι μέσω του Executive.



Δομές ΛΣ: (4) Πελάτη Διακομιστή, Αρχιτεκτονική Windows



Δομές ΛΣ: (5)

Εικονικές Μηχανές (1/3)

- Η εικονική μηχανή παρέχει «n» αντίγραφα από φυσικό υλικό χρησιμοποιώντας λογισμικό.
- Έτσι, διαφορετικά ΛΣ μπορούν να εργαστούν στο ίδιο μηχάνημα ταυτόχρονα.

Προγράμματα χρήστη

OS1

OS2

OS3

OS4

Εικονική μηχανή

Φυσικό υλικό



Δομές ΛΣ: (5)

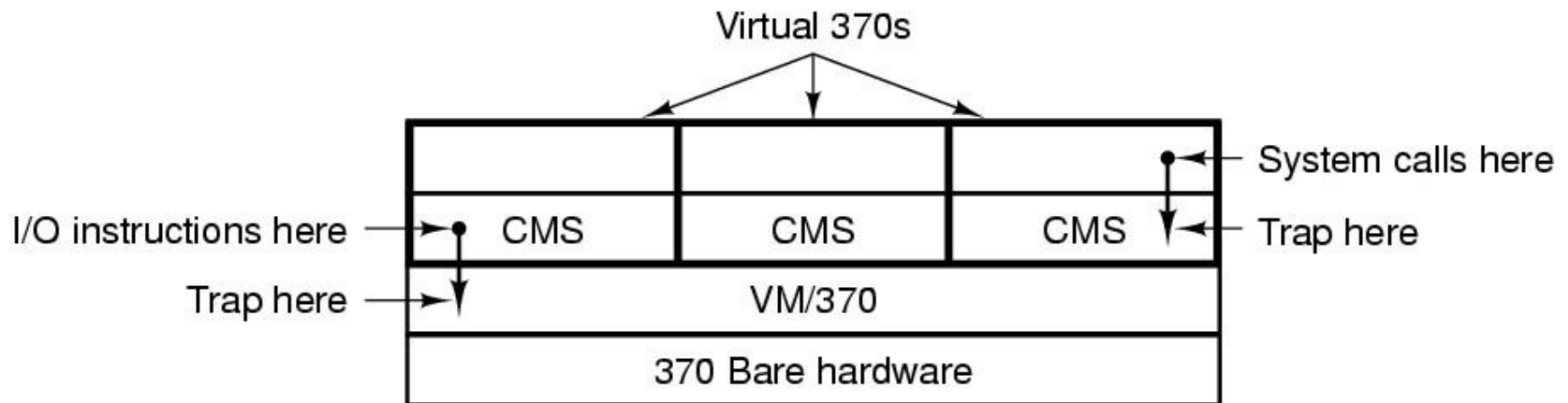
Εικονικές Μηχανές (2/3)

- Υλοποίηση σε λογισμικό ενός υπολογιστικού συστήματος, που εκτελεί προγράμματα σαν να είναι φυσική μηχανή.
 - Εικονικές Μηχανές Συστημάτων (Xen, Virtual Box, Vmware, ESX).
 - Εικονικές Μηχανές Διεργασιών (Java VM, .NET CLR).
- Εικονικές Μηχανές Συστημάτων:
 - Το ΛΣ που τρέχει στο πραγματικό υλικό και παρέχει την λειτουργικότητα των Εικονικών Μηχανών ονομάζεται host OS.
 - Το ΛΣ που εκτελείται μέσα στις Εικονικές Μηχανές ονομάζεται guest OS.
- Οι πόροι του συστήματος μοιράζονται στις Εικονικές Μηχανές (συνήθως με στατικό τρόπο), από το host OS. Οι πόροι κάθε Εικονικής Μηχανής είναι υπό τη διαχείριση του guest OS.



Δομές ΛΣ: (5)

Εικονικές Μηχανές (3/3)



Δομές ΛΣ: (5) Εικονικές Μηχανές

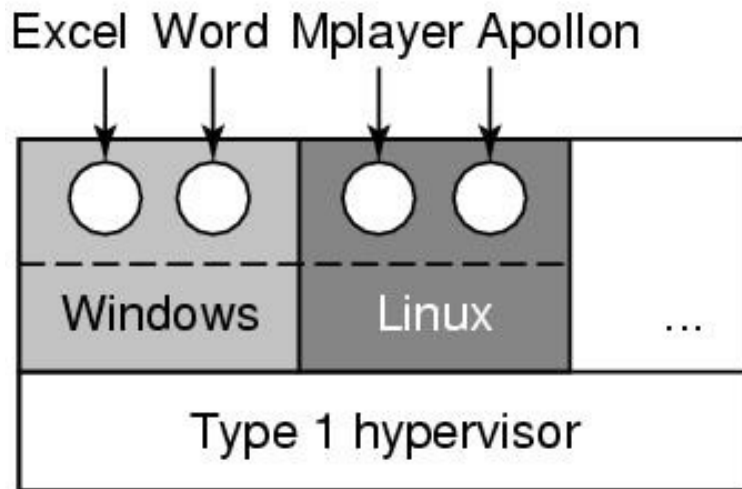
Πλεονεκτήματα & Μειονεκτήματα

- Η έννοια της ιδεατής μηχανής παρέχει **πλήρη προστασία** των πόρων του συστήματος αφού η κάθε ιδεατή μηχανή είναι πλήρως απομονωμένη από τις άλλες ιδεατές μηχανές. Αυτή η απομόνωση πάντως, δεν επιτρέπει άμεσο διαμοιρασμό πόρων.
- Ένα σύστημα ιδεατής μηχανής είναι το τέλει όχημα για έρευνα και ανάπτυξη στην περιοχή των ΛΣ. Η ανάπτυξη του συστήματος γίνεται στην ιδεατή μηχανή, αντί για την πραγματική και έτσι **δε διασπάται η κανονική λειτουργία του συστήματος**.
- Η έννοια των ιδεατών μηχανών είναι δύσκολη στην υλοποίηση λόγω του **φόρτου που απαιτείται** για την παροχή ενός πιστού αντιγράφου της υφιστάμενης μηχανής.

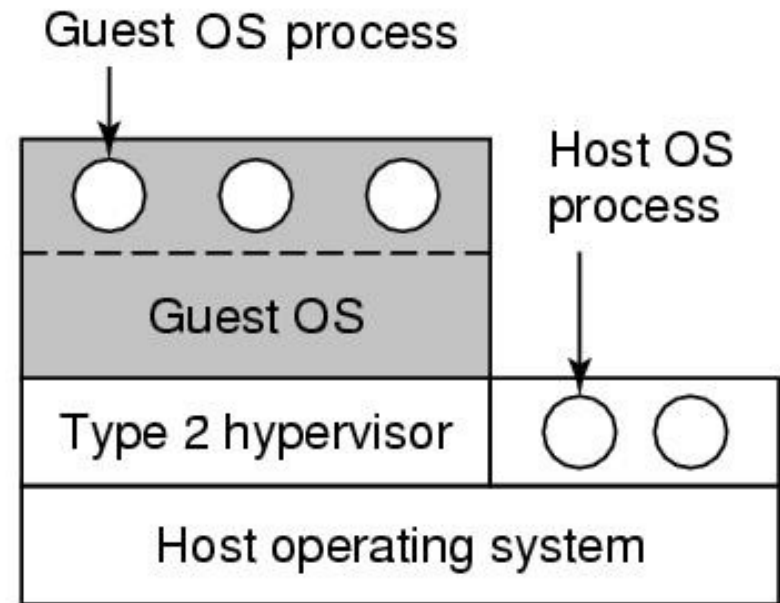


Δομές ΛΣ: (5)

Εικονικές Μηχανές (τύπου 1 & 2)

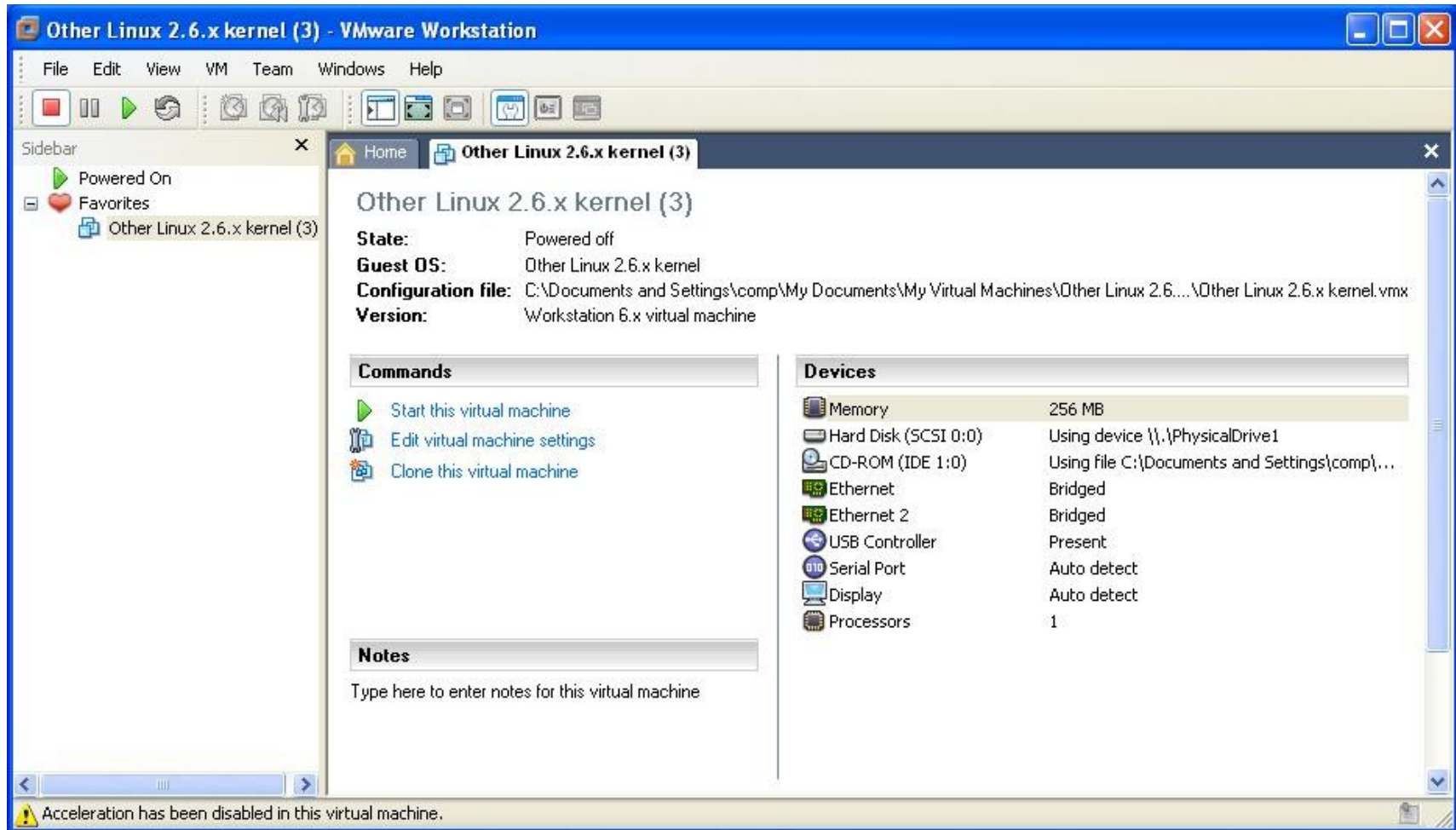


(a)



(b)

Ρυθμίσεις Virtual Machine σε Vmware



Other Linux 2.6.x kernel (3) - VMware Workstation

File Edit View VM Team Windows Help

Powered On
Favorites
Other Linux 2.6.x kernel (3)

Other Linux 2.6.x kernel (3)

State: Powered off
Guest OS: Other Linux 2.6.x kernel
Configuration file: C:\Documents and Settings\comp\My Documents\My Virtual Machines\Other Linux 2.6.x kernel.vmx
Version: Workstation 6.x virtual machine

Commands

- Start this virtual machine
- Edit virtual machine settings
- Clone this virtual machine

Devices

Memory	256 MB
Hard Disk (SCSI 0:0)	Using device \\.\PhysicalDrive1
CD-ROM (IDE 1:0)	Using file C:\Documents and Settings\comp\...
Ethernet	Bridged
Ethernet 2	Bridged
USB Controller	Present
Serial Port	Auto detect
Display	Auto detect
Processors	1

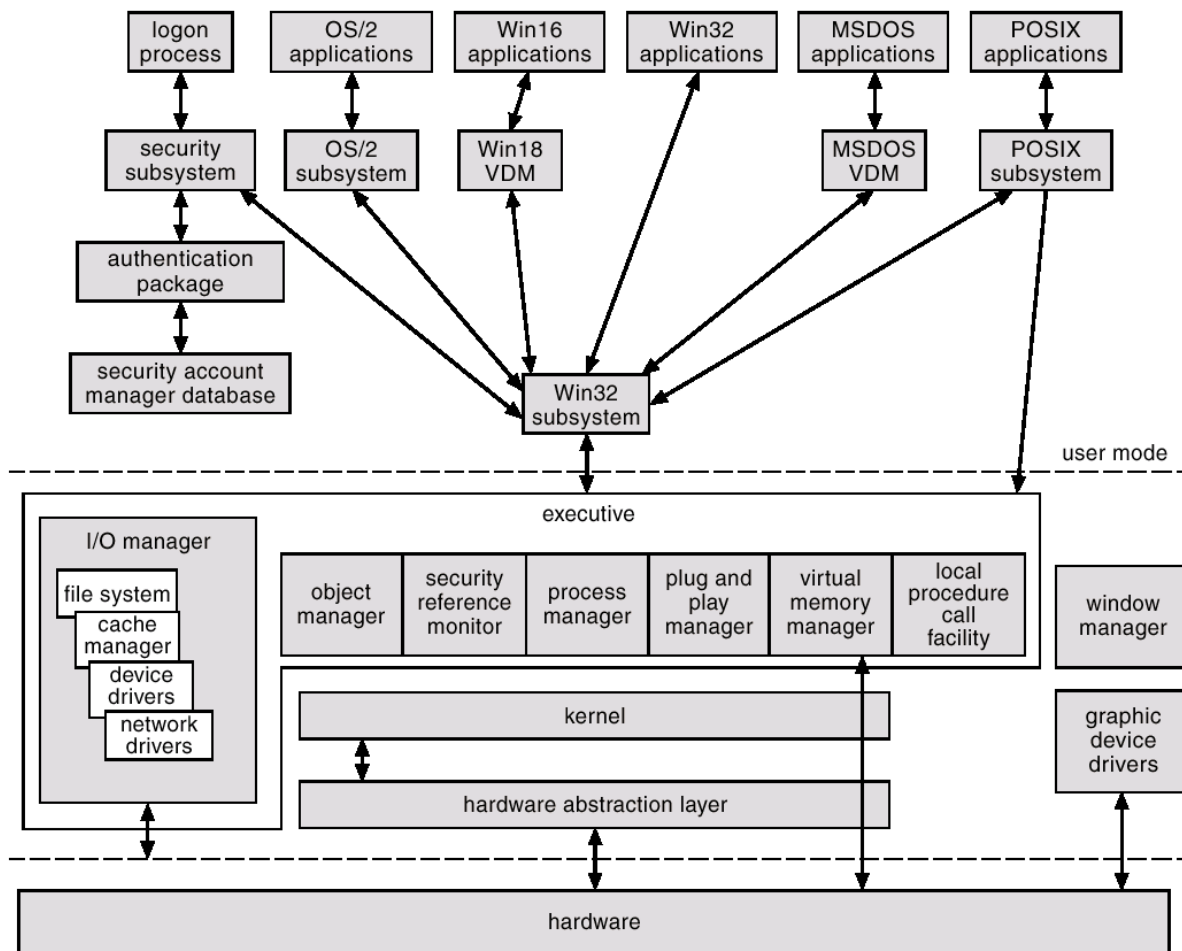
Notes

Type here to enter notes for this virtual machine

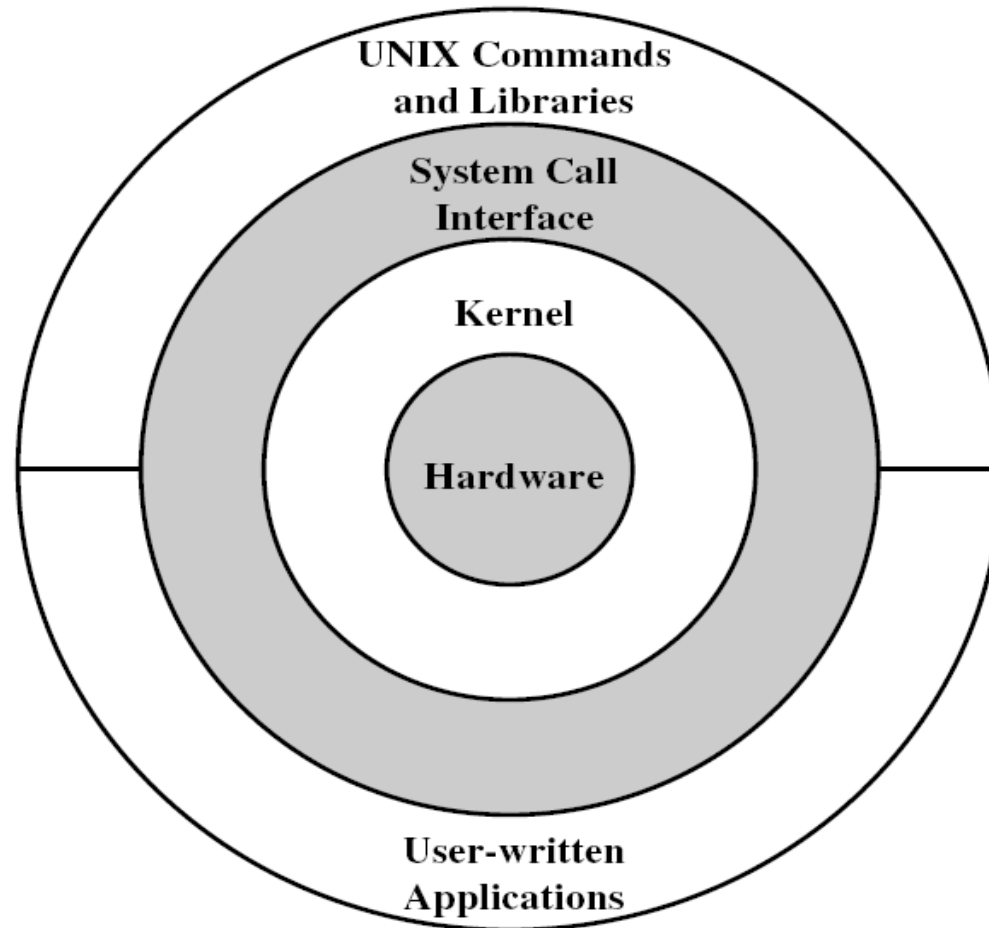
Acceleration has been disabled in this virtual machine.



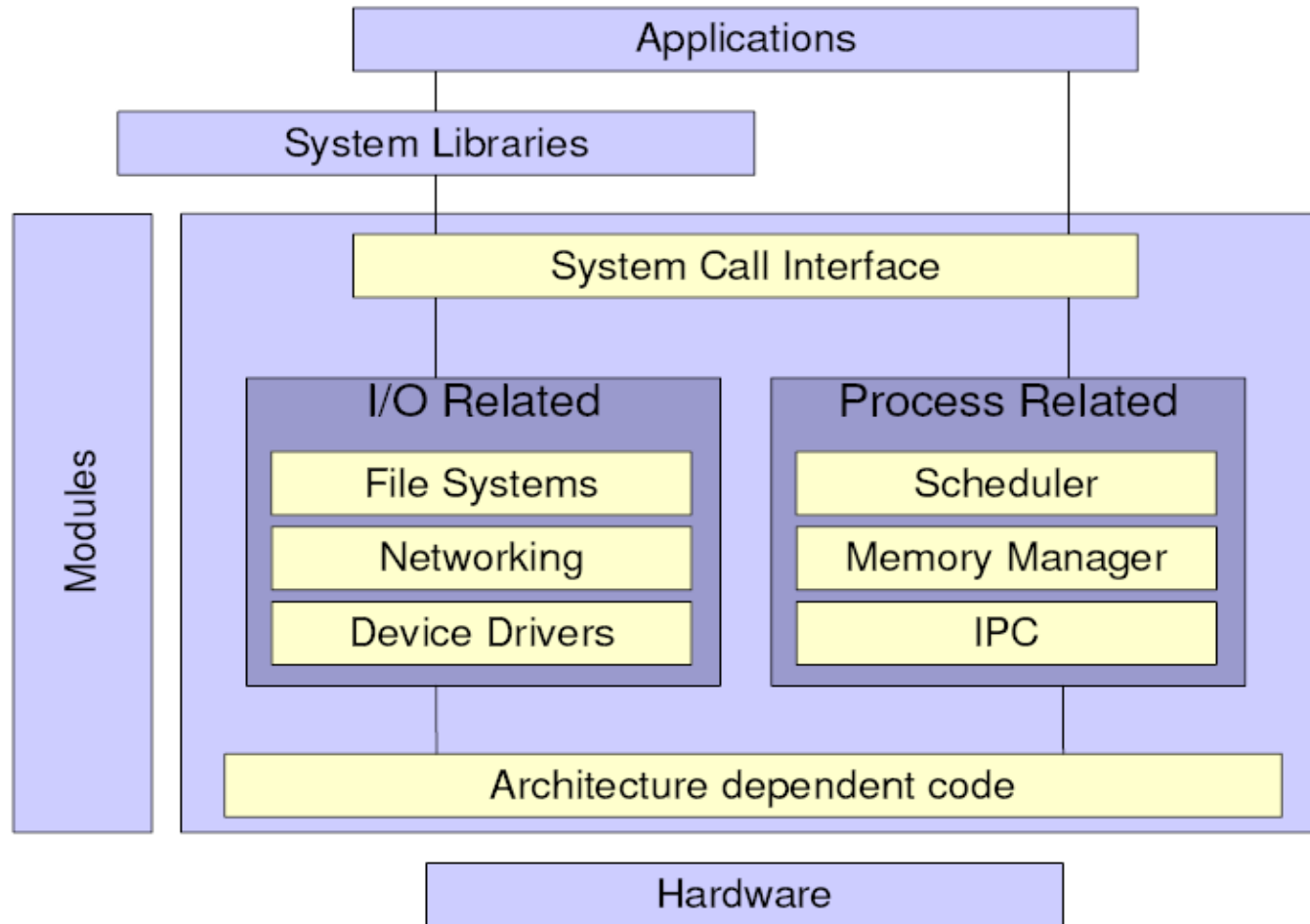
Αρχιτεκτονική ΛΣ Windows XP



Αρχιτεκτονική UNIX



Αρχιτεκτονική ΛΣ Linux

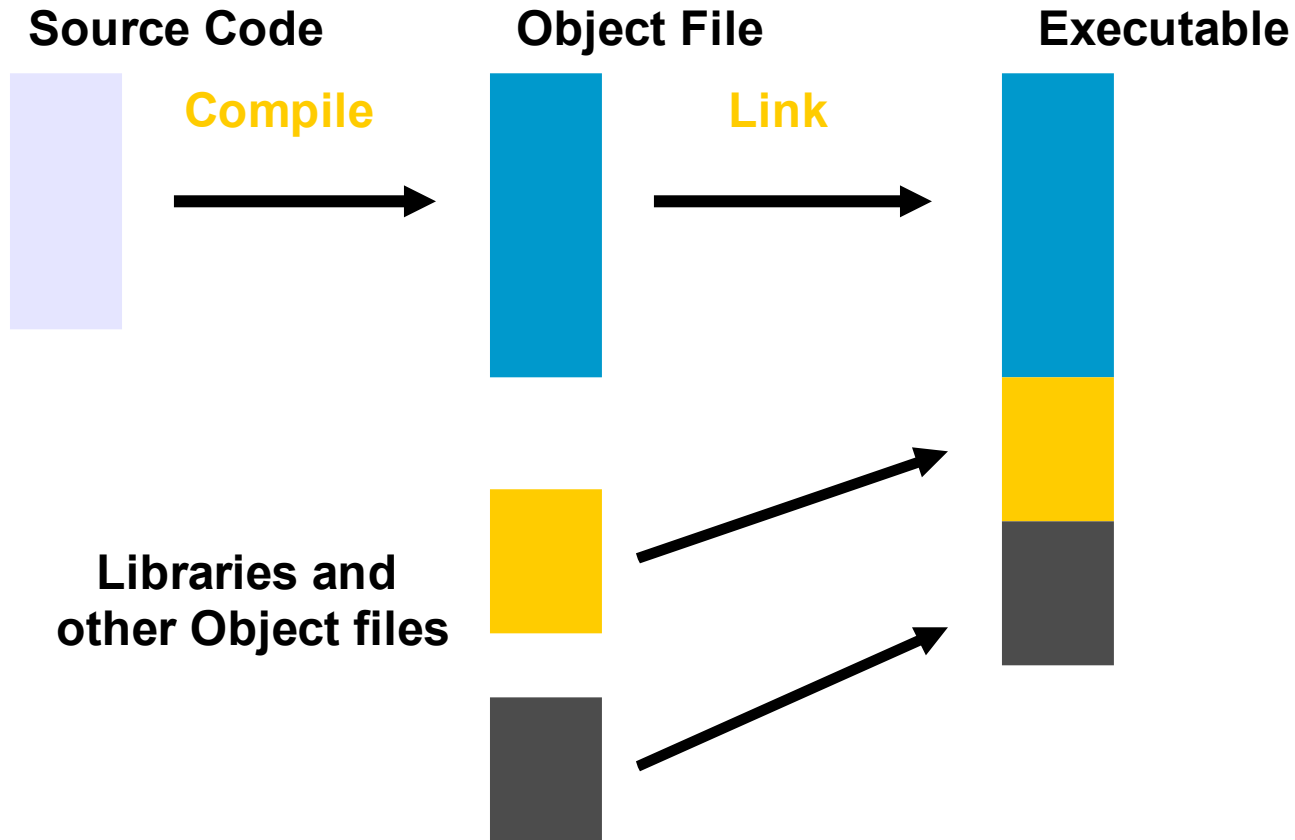


Σε ποια γλώσσα γράφονται τα ΛΣ;

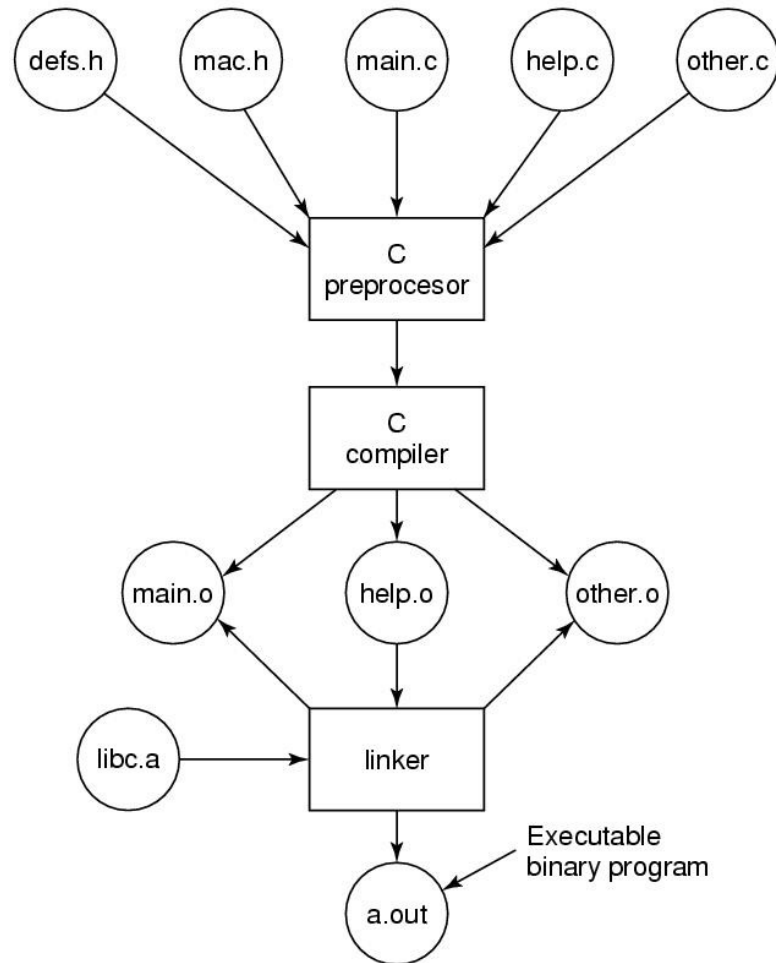
- Αν και παραδοσιακά γράφονταν σε assembly, τα ΛΣ μπορούν πλέον να γράφονται σε γλώσσες προγραμματισμού υψηλότερου επιπέδου.
- Ο κώδικας που γράφεται σε μια γλώσσα υψηλότερου επιπέδου:
 - Μπορεί να γραφεί γρηγορότερα.
 - Είναι περισσότερο συμπαγής.
 - Είναι εύκολος στην κατανόηση.
 - Είναι πιο εύκολος στην από-σφαλμάτωση (debugging).
- Ένα ΛΣ είναι ευκολότερα μεταφέρσιμο σε άλλη αρχιτεκτονική (porting) αν είναι γραμμένο σε γλώσσα υψηλού επιπέδου.



Η γλώσσα C (1/2)



Η γλώσσα C (2/2)



Ιεραρχία Εννοιών στα ΛΣ

Level	Name	Objects	Example Operations
13	Shell	User programming environment	Statements in shell language
12	User processes	User processes	Quit, kill, suspend, resume
11	Directories	Directories	Create, destroy, attach, detach, search, list
10	Devices	External devices, such as printers, displays, and keyboards	Open, close, read, write
9	File system	Files	Create, destroy, open, close, read, write
8	Communications	Pipes	Create, destroy, open, close, read, write
7	Virtual memory	Segments, pages	Read, write, fetch
6	Local secondary store	Blocks of data, device channels	Read, write, allocate, free
5	Primitive processes	Primitive processes, semaphores, ready list	Suspend, resume, wait, signal
4	Interrupts	Interrupt-handling programs	Invoke, mask, unmask, retry
3	Procedures	Procedures, call stack, display	Mark stack, call, return
2	Instruction set	Evaluation stack, microprogram interpreter, scalar and array data	Load, store, add, subtract, branch
1	Electronic circuits	Registers, gates, buses, etc.	Clear, transfer, activate, complement

Shaded area represents hardware.



A Simple Program to print a directory

```
#include <sys/types.h>
#include <dirent.h>
#include "ourhdr.h"
```

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

```
    DIR *dp;
    struct dirent *dirp;
```

```
    if (argc != 2)
        err_quit("a single argument (the directory name) is required");
    if ( (dp = opendir(argv[1])) == NULL)
        err_sys("can't open %s", argv[1]);
```

```
    while ( (dirp = readdir(dp)) != NULL)
        printf("%s\n", dirp->d_name);
```

```
    closedir(dp);
    exit(0);
}
```

Functions supplied by system libraries.
These functions will contain a trap instruction.



Χαρακτηριστικά των σύγχρονων Λ.Σ

- Εξέλιξη του υλικού:
 - Πολλοί επεξεργαστές.
 - Υψηλή ταχύτητα συνδέσεων δικτύου.
 - Πολλές και μεγάλες σε χωρητικότητα συσκευές αποθήκευσης.
- Εξέλιξη του λογισμικού:
 - Πολυμεσικές εφαρμογές.
 - Πρόσβαση στο διαδίκτυο.
 - Μοντέλο πελάτη / εξυπηρέτη (client / server).



Ενδεικτική Ερώτηση

- Ποιες από τις παρακάτω εντολές (*instructions*) θεωρείτε απαραίτητο να προστατεύονται από τους χρήστες;
 - Αλλαγή σε κατάσταση χρήστη (*user mode*).
 - Ανάγνωση του ρολογιού.
 - Απενεργοποίηση διακοπών (*interrupts*).
 - Αλλαγή σε κατάσταση επόπτη (*monitor mode*).
 - Ανάγνωση από τη μνήμη του παρακολουθητή.
 - Απόδοση τιμής στον χρονιστή.
 - Εγγραφή στη μνήμη του παρακολουθητή.



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

