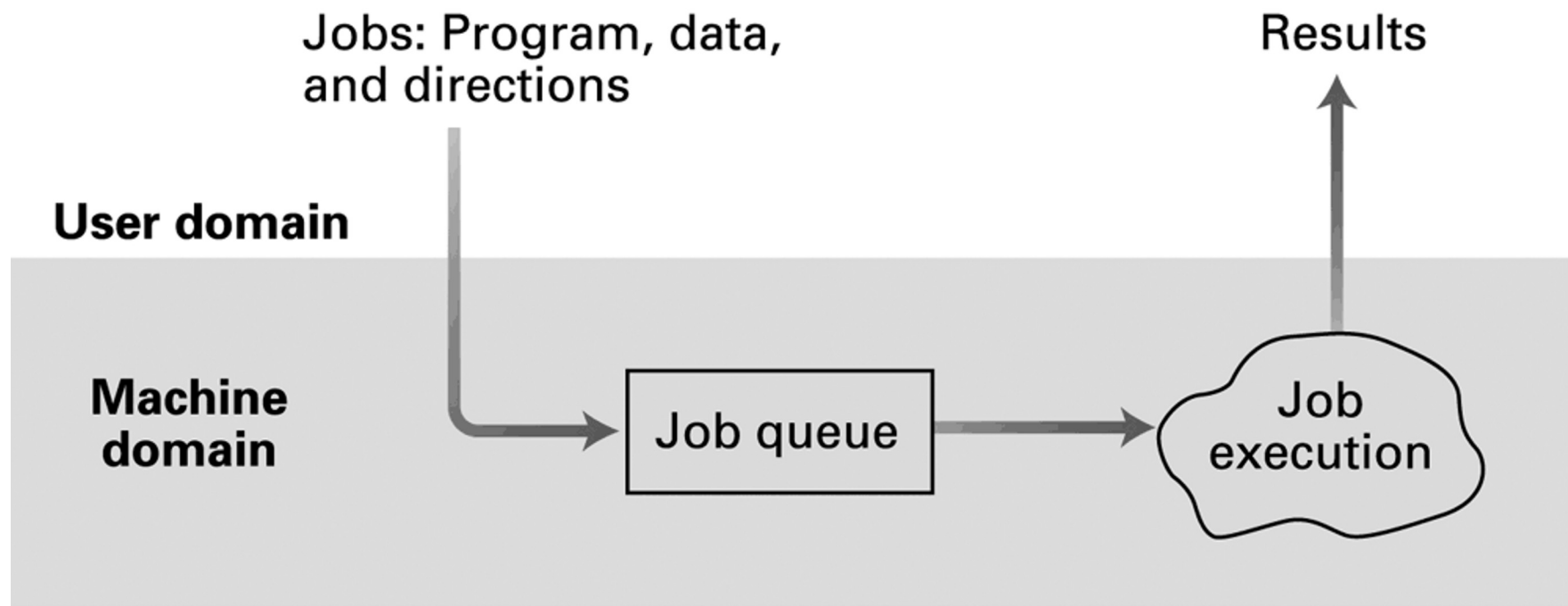
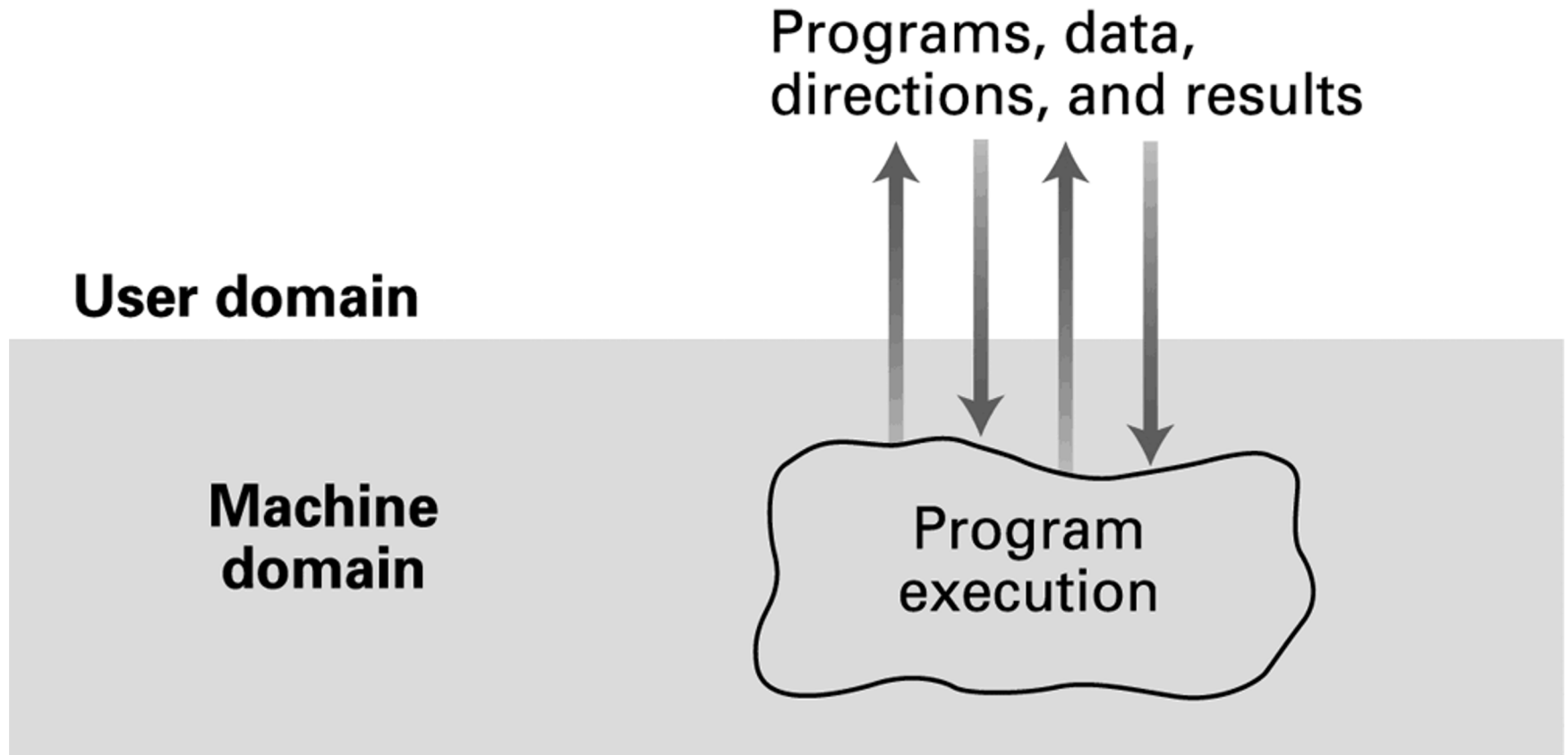

Λειτουργικά συστήματα

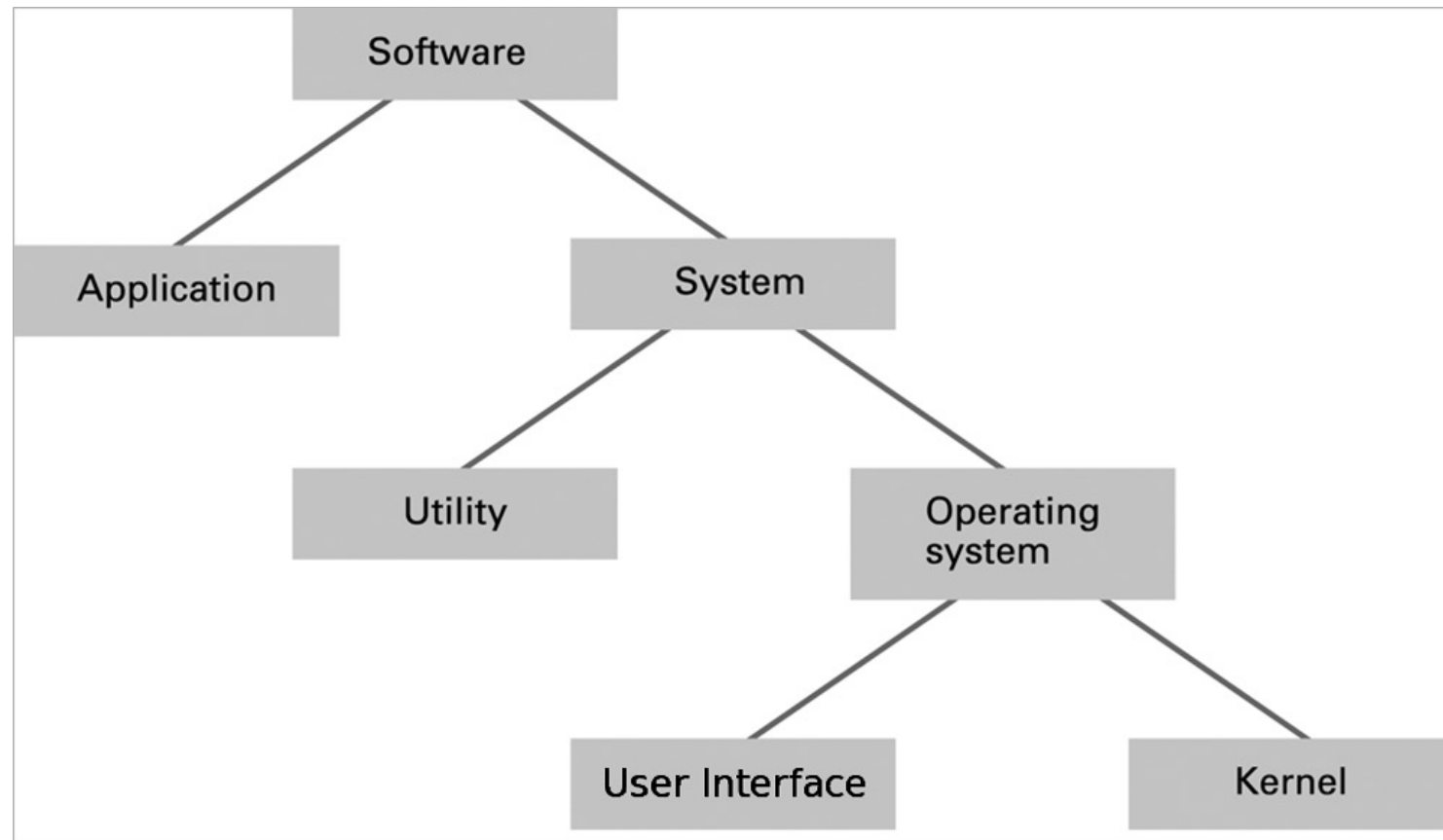
Batch processing



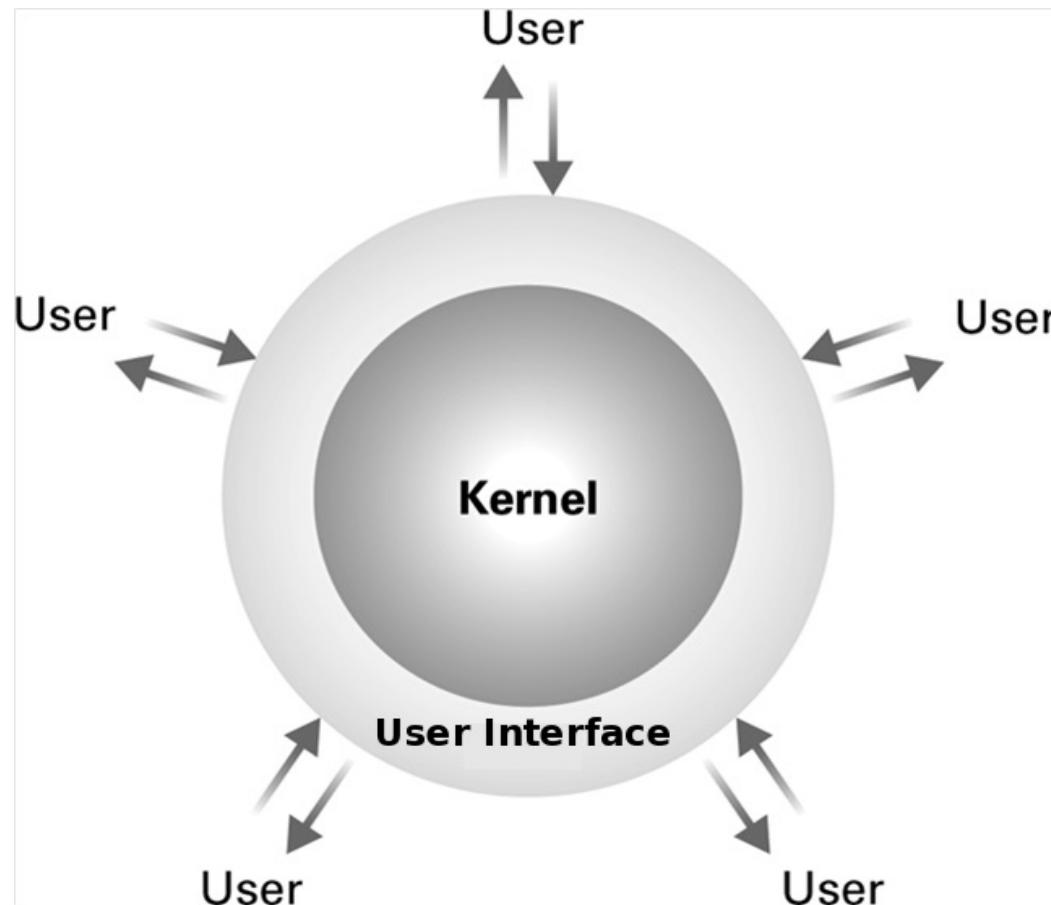
Interactive processing



Software classification



The user interface act as an intermediary between users and the operating system kernel



Operating System Components

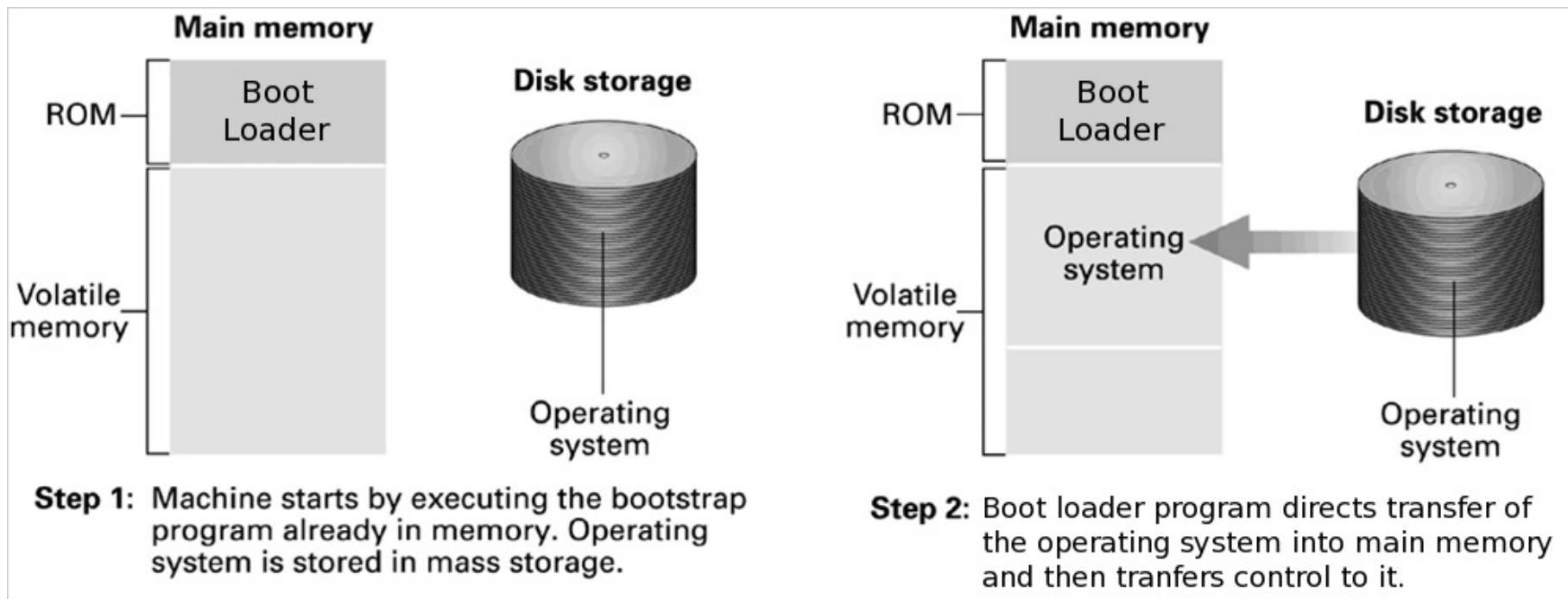
User Interface: Communicates with users

- Text based (Shell)
- Graphical user interface (GUI)

Kernel: Performs basic required functions

- File manager
- Device drivers
- Memory manager
- Scheduler and dispatcher

Figure 3.5 The booting process



Λειτουργικά συστήματα

Το λειτουργικό σύστημα εκπληρώνει πολλές επιμέρους λειτουργίες:

1. Κρύβει από το χρήστη τα χαρακτηριστικά του υλικού του υπολογιστή (χειρισμός εισόδου-εξόδου)
2. Παρέχει ευκολίες για το χειρισμό προγραμμάτων και δεδομένων (π.χ., μεταγλωτίση προγραμμάτων, συγχώνευση αρχείων)
3. Διαμοιράζει τους πόρους του Η/Υ σε χρήστες και προγράμματα – συσκευές που δουλεύουν ταυτόχρονα

Λειτουργικά συστήματα

Μια **εργασία (job)** μπορεί να θεωρηθεί σαν μια σειρά δράσεων απαραίτητων για να εκτελεστεί ένα πρόγραμμα (π.χ., μεταγλώττιση του πηγαίου προγράμματος, φόρτωμα και εκτέλεση του αντικείμενου προγράμματος)

Χωρίς λειτουργικό σύστημα ο χρήστης θα έπρεπε:

1. Να παρουσιάσει τον πηγαίο κώδικα σε κατάλληλη είσοδο
2. Να ενεργοποιήσει τον μεταγλωττιστή ώστε να διαβάσει από την είσοδο και να τοποθετήσει το αντικείμενο πρόγραμμα στη δευτερεύουσα μνήμη
3. Να ενεργοποιήσει το φορτωτή ώστε να μεταφέρει το αντικείμενο πρόγραμμα από τη δευτερεύουσα μνήμη στην κύρια μνήμη
4. Να τα τοποθετήσει τα δεδομένα που απαιτούνται στην κατάλληλη συσκευή εισόδου
5. Να αρχίσει την εκτέλεση του προγράμματος
6. Να συλλέξει το αποτέλεσμα από τη συσκευή εξόδου

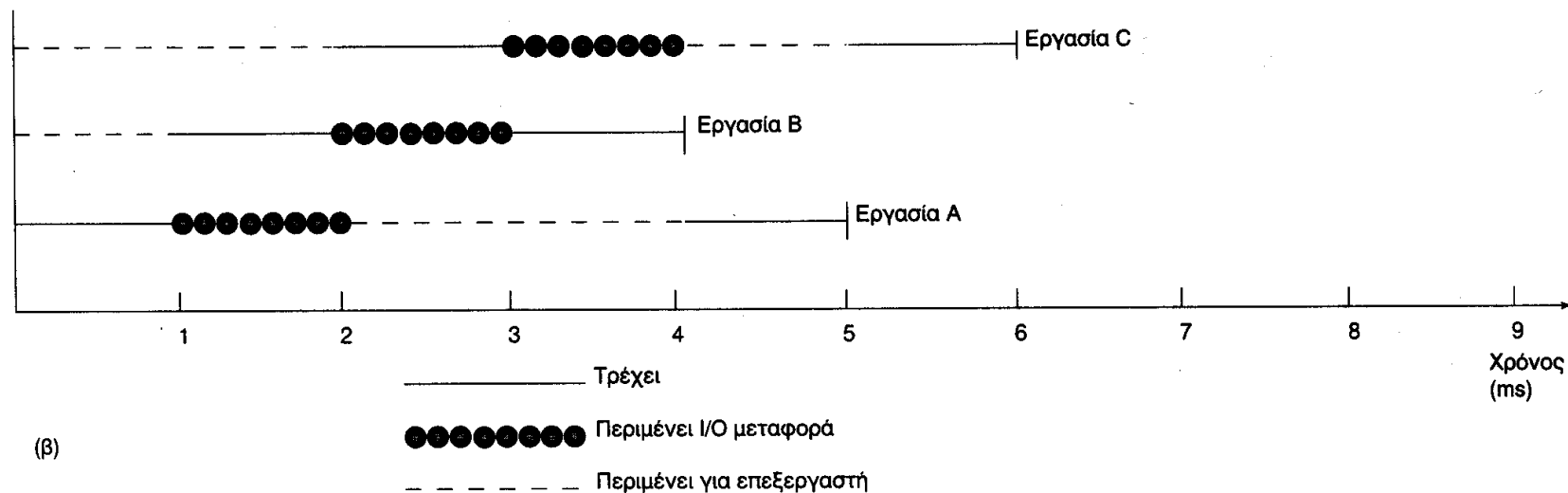
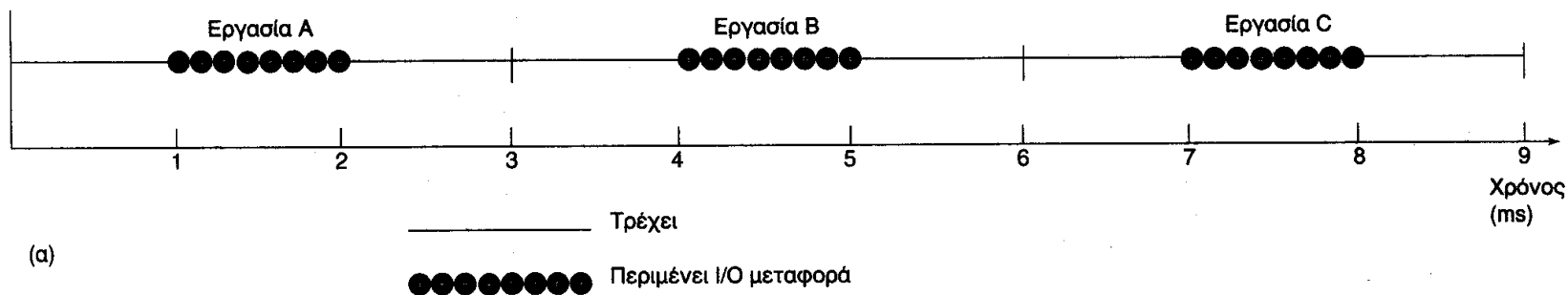
Λειτουργικά συστήματα

- Ο ρυθμός διεκπεραίωσης εργασιών είναι πολύ χαμηλός
- Βελτίωση με τη χρήση κατάλληλου προγράμματος που ενεργοποιεί το μεταγλωττιστή, το φορτωτή και το αντικείμενο πρόγραμμα την κατάλληλη στιγμή
- Ανάγκη ύπαρξης πληροφοριών ελέγχου για κάθε εργασία
- Επίσης απαραίτητη η δυνατότητα διαχείρισης λαθών
- ...ουσιαστικά ένα εμβρυακό λειτουργικό σύστημα

Διαχείριση λειτουργιών εισόδου - εξόδου

- Με τη χρήση του λειτουργικού ο **ρυθμός εξυπηρέτησης** εργασιών εξαρτάται ουσιαστικά από την **ταχύτητα των συσκευών εισόδου/εξόδου**
- Χρήση **διακοπών (interrupts)** που δείχνουν ότι η μεταφορά δεδομένων έχει ολοκληρωθεί
- Η διακοπή επιτρέπει στην CPU να αρχίσει την είσοδο/έξοδο, να συνεχίσει την επεξεργασία, και να ειδοποιηθεί όταν η είσοδος/έξοδος έχει ολοκληρωθεί
- Επικάλυψη εισόδου/εξόδου και επεξεργασίας με **πολυπρογραμματισμό (multiprogramming)**: συνύπαρξη προγραμμάτων και δεδομένων για πολλές εργασίες στη μνήμη την ίδια στιγμή
- Πλεονεκτήματα από τη χρήση πολυπρογραμματισμού:
 - Αύξηση ρυθμού διεκπεραίωσης
 - Βελτίωση στη χρήση πόρων (π.χ., περιφερειακά)
 - Δυνατότητα πρόσβασης πολλών χρηστών ταυτόχρονα

Πολυπρογραμματισμός



Αρχικά 333 εργασίες το δευτερόλεπτο (1 εργασία κάθε 3 ms)
Τελικά 500 εργασίες το δευτερόλεπτο (1 εργασία κάθε 2 ms)
Αύξηση της απόδοσης της τάξης του 50%

Επαύξηση δυνατοτήτων του βασικού λειτουργικού συστήματος

1. **Διανομή (dispatching)** Κατάλληλη εναλλαγή του υπολογιστή από μια εργασία σε άλλη.
2. **Χειρισμός διακοπών (interrupt handling)** Διερμηνεία κάθε διακοπής ως σήμα ότι η μεταφορά εισόδου/εξόδου έχει ολοκληρωθεί και γι ' αυτό η εργασία η οποία ζήτησε τη μεταφορά μπορεί τώρα να προχωρήσει στην εκτέλεσή της.
3. **Εκχώρηση πόρων (resource allocation)** Εκχώρηση των πόρων του υπολογιστή σε εργασίες σύμφωνα με τις απαιτήσεις (μνήμη, συσκευές εισόδου-εξόδου, επεξεργαστή)
4. **Προστασία πόρων (resource protection)** Εξασφάλιση ότι καμία εργασία δεν μπορεί να έχει πρόσβαση σε κάποιο πόρο, ο οποίος δεν της έχει εκχωρηθεί.
5. **Χρονοπρογραμματισμός (scheduling)** Απόφαση για το ποια εργασία (ή εργασίες) θα επιλεγούν για εκτέλεση όταν κάποια άλλη εργασία τελειώσει.. Η απόφαση βασίζεται σε παράγοντες όπως το πόσο επείγει η κάθε εργασία, οι πόροι που προς το παρόν διατίθενται και οι πόροι που απαιτεί κάθε εργασία.

Διανομέας

1. Η εκτέλεση ενός προγράμματος καλείται διεργασία (process)
2. Υπάρχουν διεργασίες του λειτουργικού συστήματος και των χρηστών
3. Πρέπει να γίνεται καταμερισμός της CPU μεταξύ των διεργασιών
4. Ο καταμερισμός επιτυγχάνεται με την εναλλαγή των διεργασιών στη CPU για εκτέλεση (π.χ., κάθε πεντηκοστό του δευτερολέπτου)

Όταν μια διεργασία ξεκινήσει μπορεί να βρίσκεται σε μία από τις ακόλουθες καταστάσεις:

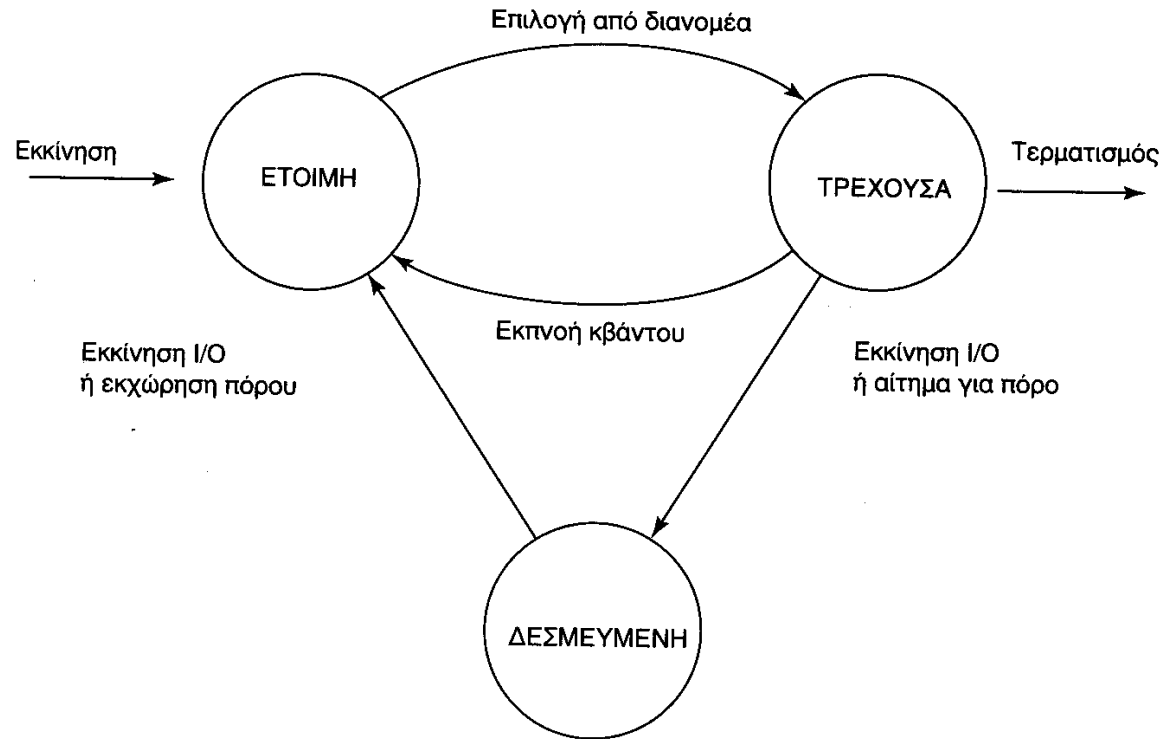
1. Τρέχουσα (running).
2. Έτοιμη (ready).
3. Δεσμευμένη ή μπλοκαρισμένη (blocked).

Διανομέας

Η ευθύνη για την εναλλαγή διεργασιών στη CPU ανατίθεται στο διανομέα (dispatcher) ο οποίος καλείται όταν:

1. Μια διεργασία απαιτεί μεταφορά εισόδου/εξόδου και άρα δεσμεύεται, ή
2. Όταν τελειώσει το χρονομερίδιο μιας διεργασίας που τρέχει, δείχνοντας ότι η αντίστοιχη CPU πρέπει να εκχωρηθεί σε μια άλλη διεργασία
3. Μια διεργασία που τρέχει, ολοκληρώσει την εκτέλεση της.

Διανομέας



Για την αποφυγή μονοπώλησης της CPU από μία διεργασία, ορίζεται ως μέγιστο ποσό χρόνου εκτέλεσης στη CPU κάθε φορά ένα χρονομερίδιο ή χρονο-τεμάχιο (time-slice), π.χ., 0,02 – 0,25 sec

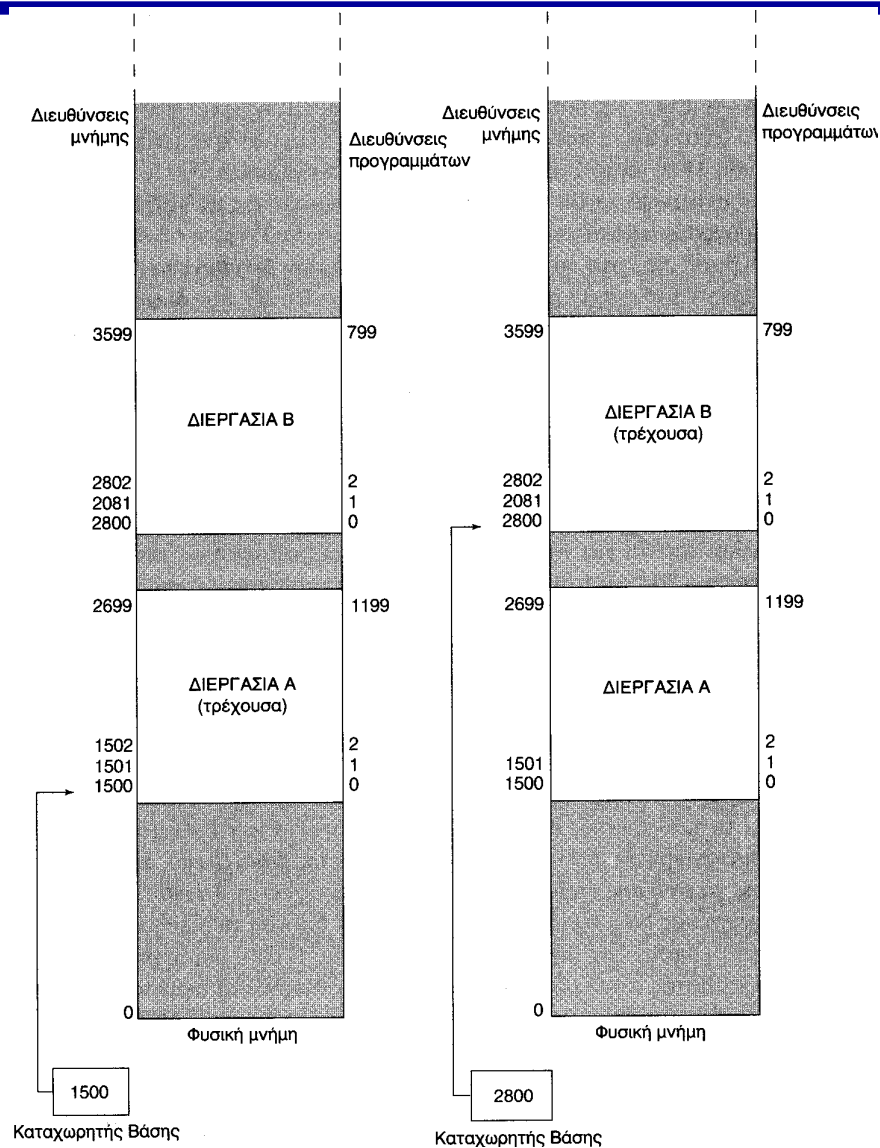
Διαχείριση μνήμης

Η διαχείριση της μνήμης έχει τρεις βασικές όψεις:

1. **Εκχώρηση (allocation)**: ορισμός μνήμης ικανής να χωρέσει το πρόγραμμα και τα δεδομένα του
2. **Προστασία (protection)**: παρεμπόδιση της προσπέλασης οποιασδήποτε διεργασίας σε περιοχή της μνήμης που δεν της έχει εκχωρηθεί.
3. **Χρησιμοποίηση (utilization)**: εκχωρούμε όσο το δυνατό μεγαλύτερο μέρος της και εξασφάλιση ύπαρξης διεργασιών στη μνήμη που είναι έτοιμες να εκτελεστούν

Διαχωρισμός μεταξύ διευθύνσεων προγράμματος και διευθύνσεων μνήμης

Καταχωρητές βάσης και ορίου



Μετασχηματισμός διεύθυνσης

module μετασχηματισμός-διεύθυνσης (α)

{αλγόριθμος μετασχηματισμού διευθύνσεων που χρησιμοποιεί τους καταχωρητές βάσης και ορίου. Το α είναι η διεύθυνση προγράμματος, β και σ είναι τα περιεχόμενα των καταχωρητών βάσης και ορίου αντίστοιχα}.

if $\alpha < 0$ ή $\alpha \geq \sigma$

then υπόδειξε λάθος διεύθυνσης

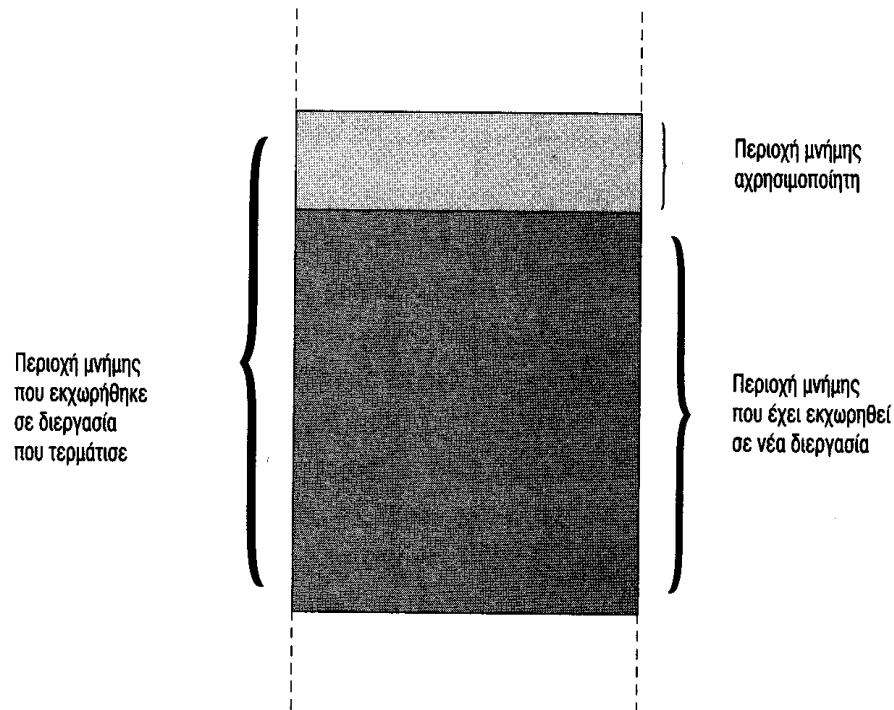
else θέσε τη διεύθυνση μνήμης ίση με $\alpha + \beta$

Διαδικασία ανταλλαγής

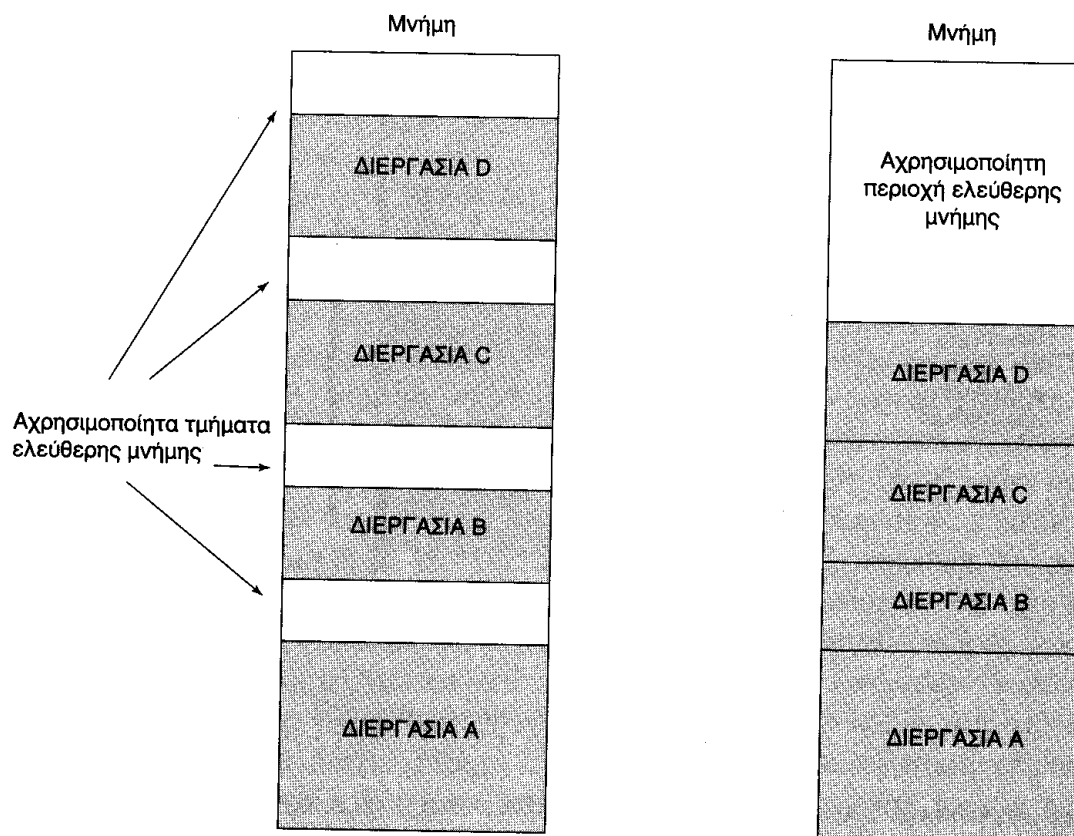
- Σκοπός του λειτουργικού είναι να κρατά στη μνήμη όσο το δυνατό περισσότερες διεργασίες
- Η περιοχή μνήμης που καταλαμβάνει μια διεργασία δε μπορεί αν χρησιμοποιηθεί από άλλη
- Διαδικασία swapping (ανταλλαγή): μεταφορά διεργασιών από την κύρια μνήμη στη δευτερεύουσα και δημιουργία χώρου για άλλες
- Όταν η διεργασία επιστρέφει στη μνήμη δεν καταλαμβάνει το ίδιο κομμάτι μνήμης
- Κριτήρια για την ανταλλαγή: προτεραιότητα διεργασίας, μέγεθος μνήμης που καταλαμβάνει, διάρκεια χρόνου που εκτιμάται ότι θα είναι δεσμευμένη, διάρκεια που έχει καταλάβει τη μνήμη

Κατακερματισμός μνήμης

- Ο υπερβολικός κατακερματισμός σημαίνει και χαμηλή χρησιμοποίηση της μνήμης
- Αντιμέτωπιση με προσεκτική εκχώρηση της μνήμης βάση κατάλληλων αλγορίθμων, συγχώνευση γειτονικών περιοχών αχρησιμοποίητης μνήμης, ή συμπύκνωση



Συμπύκνωση (compaction)



Μειονέκτημα των παραπάνω σεναρίων:

1. Δεσμεύεται χώρος για όλο το πρόγραμμα και τα δεδομένα
2. Το μέγεθος του προγράμματος και των δεδομένων δεσμεύεται από το μέγεθος της μνήμης

Σελιδοποίηση (paging)

- Σελιδοποίηση: τεχνική διαχείρισης της μνήμης που επιτρέπει σε μια διεργασία να εκτελεστεί ενώ μόνο μέρος της βρίσκεται στη μνήμη
- Το πρόγραμμα διαμερίζεται σε σελίδες ίσου μήκους και η μνήμη χωρίζεται σε ένα αριθμό πλαισίων σελίδας του ίδιου μεγέθους
- Αν το μέγεθος σελίδας είναι 1000 τότε η σελίδα 0 περιέχει διευθύνσεις 0-999 και η σελίδα 1 1000-1999
- Η CPU μετασχηματίζει μια διεύθυνση προγράμματος σε διεύθυνση μνήμης με τον υπολογισμό από τη σελίδα και τη θέση (π.χ., 2764 → σελίδα 2 θέση 764) και τη χρήση πληροφοριών του λειτουργικού συστήματος (πίνακας σελίδων) για το πλαίσιο σελίδας που καταλαμβάνει
- Χαμηλός κατακερματισμός αλλά χρήση από πίνακες σελίδων και αύξηση της πολυπλοκότητας

Μετασχηματισμός διεύθυνσης

module μετασχηματισμός-διεύθυνσης (a)

{ αλγόριθμος μετασχηματισμού της διεύθυνσης για τη σελιδοποίηση
 a είναι η διεύθυνση προγράμματος, z είναι το μέγεθος σελίδας }

θέσε το αριθμός σελίδας p ίσο με το ακέραιο μέρος του a/z

θέσε το θέση w μέσα στη σελίδα ίσο με το υπόλοιπο του a/z

προσπέλασε την p στη θέση του πίνακα σελίδων

if η σελίδα είναι στην μνήμη

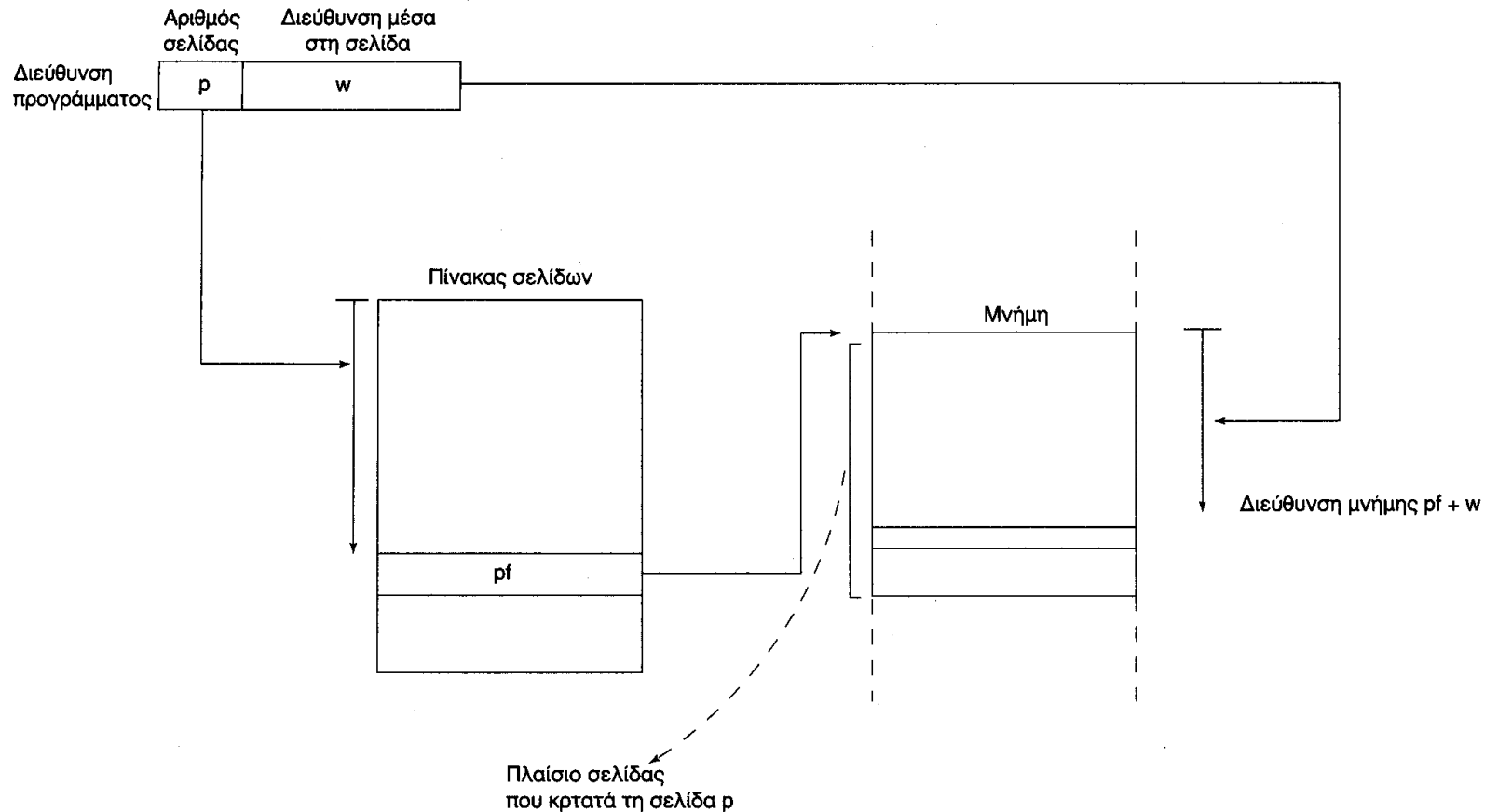
then πάρε από τον πίνακα την διεύθυνση pf του πλαισίου σελίδας
θέσε το διεύθυνση μνήμης ίσο με $pf+w$

else κάνε το λειτουργικό σύστημα να μεταφέρει τη σελίδα

στη μνήμη και ενημέρωσε κατάλληλα τον πίνακα σελίδων

{ Η διεργασία δεσμεύεται μέχρι να ολοκληρωθεί αυτό, κατόπιν
ο αλγόριθμος ξαναεκτελείται }

Μετασχηματισμός διεύθυνσης



Επιλογή σελίδας προς αντικατάσταση

Υπάρχουν τρεις επιλογές που χρησιμοποιούνται συχνά για την επιλογή της σελίδας η οποία θα αντικατασταθεί:

1. **Μεγαλύτερη παραμονή στη μνήμη (longest resident in memory)** με την υπόθεση ότι πιθανώς δεν θα χρησιμοποιηθεί ξανά.
2. **Λιγότερο πρόσφατη χρησιμοποίηση (least recently used)** με την υπόθεση πως μια σελίδα που δεν έχει χρησιμοποιηθεί πολύ πρόσφατα δεν θα χρησιμοποιηθεί πάλι στο άμεσο μέλλον .
3. **Λιγότερο συχνή χρησιμοποίηση (least frequently used)** με την υπόθεση ότι μια σελίδα που δεν έχει χρησιμοποιηθεί πολύ συχνά στο άμεσο παρελθόν είναι ομοίως πιθανό να μη χρησιμοποιηθεί στο άμεσο μέλλον.

Χρονοπρογραμματισμός και εκχώρηση πόρων

- Η ευθύνη για την εκκίνηση μιας λειτουργίας επαφίεται σε ένα στοιχείο που λέγεται χρονοπρογραμματιστής (scheduler)
- Για να αποφασίσει ο χρονοπρογραμματιστής πότε θα ξεκινήσει μια διεργασία παίρνει υπόψη του παράγοντες όπως:
 1. Η ποσότητα των πόρων (π.χ. μνήμη, χρόνος επεξεργασίας, I/O συσκευές) που απαιτεί μια διεργασία.
 2. Η ποσότητα των πόρων που είναι διαθέσιμες επί του παρόντος.
 3. Η προτεραιότητα της εργασίας με την οποία σχετίζεται η διεργασία.
 4. Το χρονικό διάστημα που περίμενε η σχετική εργασία.
- Επιλογή βάση κριτηρίων (π.χ., μεγαλύτερος βαθμός διεκπεραίωσης ή παροχή προτεραιότητας σε συγκεκριμένες λειτουργίας?)
- Στατική (απαιτούνται όλοι οι πόροι) ή δυναμική (δύσκολη στη διαχείριση – deadlocks) εκχώρηση πόρων

Χειρισμός εισόδου - εξόδου

➤ Ο χειρισμός των συναλλαγών εισόδου/εξόδου γίνεται από το λειτουργικό σύστημα :

1. Οι συσκευές I/O που είναι συνδεδεμένες σ' έναν υπολογιστή διαθέτουν μεγάλη ποικιλία λειτουργικών χαρακτηριστικών
2. Για την αποδοτική εκχώρηση των επεξεργαστών σε σχέση με το πότε αρχίζουν και τελειώνουν οι μεταφορές I/O
3. Για την ασφαλή διαχείριση μοιραζόμενων shareable – (σκληρός δίσκος) και μη μοιραζόμενων –unshareable (εκτυπωτής) συσκευών I/O

➤ Η κλήση για μεταφορά I/O ενεργοποιεί το κατάλληλο στοιχείο του λειτουργικού σε:

1. Έλεγχος της ορθότητας της αίτησης (δηλαδή ότι η συσκευή I/O είναι εκχωρημένη στη ζητούσα διεργασία).
2. Αρχικοποίηση της μεταφοράς εκτελώντας εντολές κατάλληλες για την αναφερόμενη συσκευή I/O.
3. Δέσμευσης της διεργασίας που υπέβαλλε την αίτηση.
4. Κλήση του διανομέα να επιλέξει κάποια άλλη διεργασία για να τρέξει.

Χειρισμός εισόδου - εξόδου

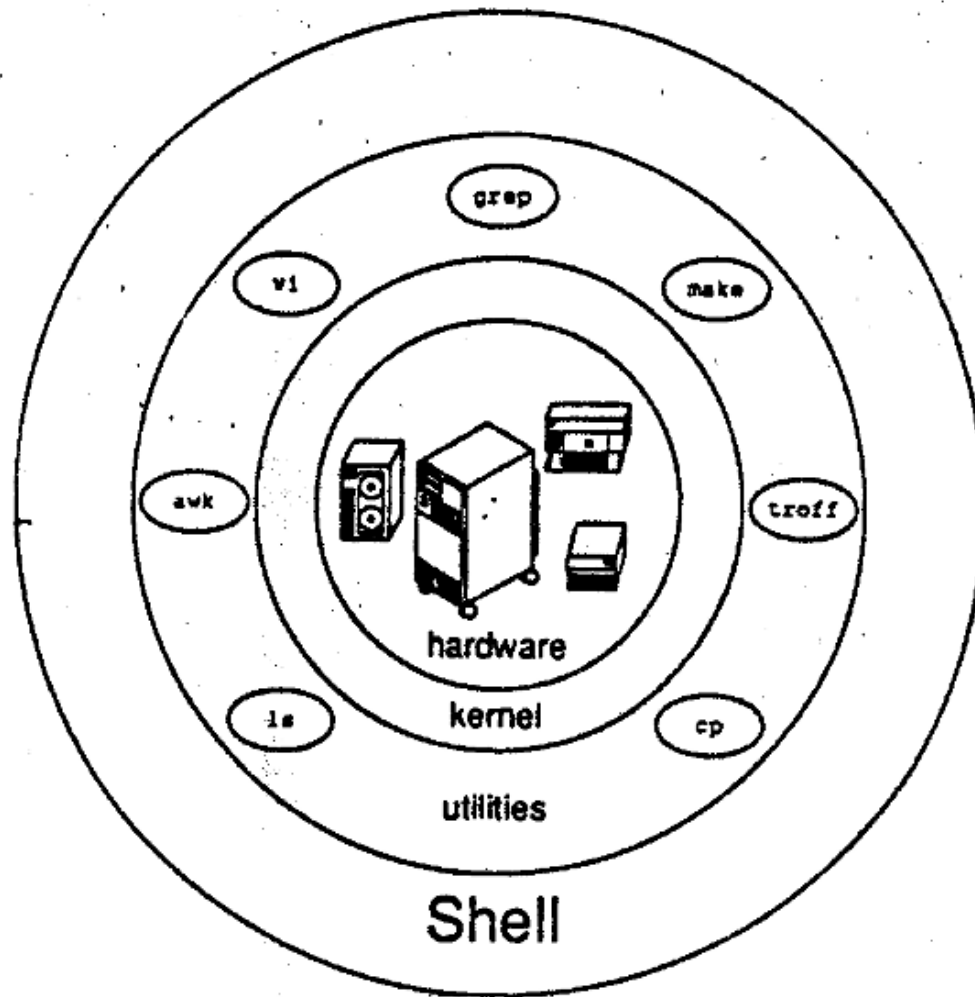
- Η ολοκλήρωση της μεταφοράς σηματοδοτείται με τη βοήθεια μιας διακοπής (η CPU εκτελεί ένα άλμα σε μια σταθερή θέση στη μνήμη)
- Αυτή η θέση περιέχει την πρώτη οδηγία του χειριστή διακοπής (interrupt handler),

Ο χειριστής διακοπής:

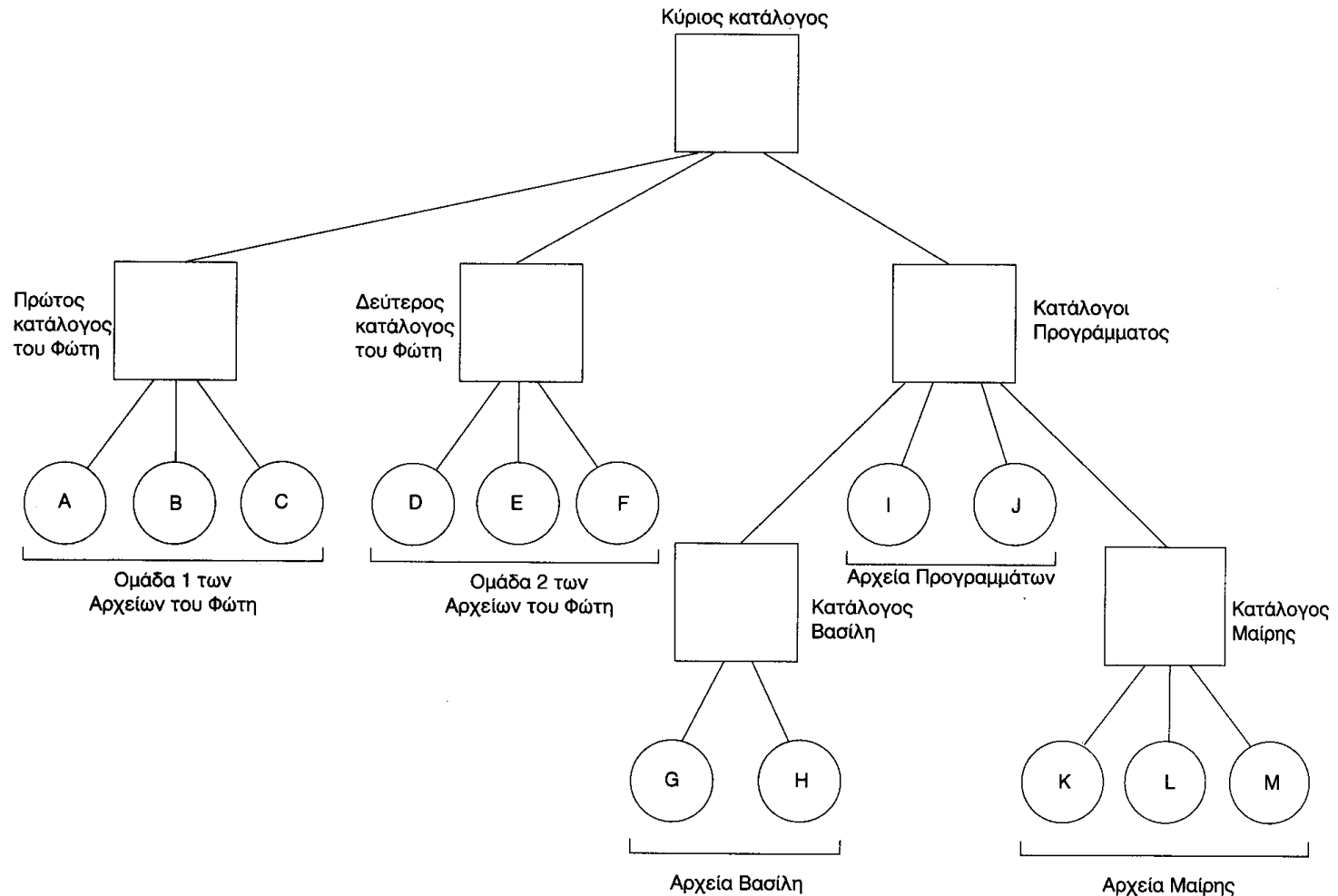
1. Προσδιορίζει την πηγή της διακοπής και συνεπώς τη διεργασία
2. Μεταβάλλει την κατάσταση αυτής της διεργασίας (δεσμευμένη → έτοιμη)
3. Ενεργοποιεί εκ νέου τη διεργασία, η οποία έτρεχε όταν σημειώθηκε η διακοπή.

Έλεγχος εργασιών

- Οι οδηγίες από τον χρήστη στο λειτουργικό σύστημα εκφράζονται σε μια γλώσσα Ελέγχου Εργασιών (job control language) ή γλώσσα προσταγών (command language).
- Οι περισσότερες απ' αυτές τις γλώσσες επιτρέπουν να μεταβιβάζονται οι ακόλουθες μορφές πληροφοριών:
 1. **Λειτουργικές πληροφορίες (operational information)** → τι λειτουργίες θέλει ο χρήστης να εκτελεστούν (π.χ. η αντιγραφή ενός αρχείου, η εκτέλεση ενός προγράμματος).
 2. **I/O πληροφορίες (I/O information)**, → από που πρόκειται να έρθουν τα προγράμματα και/ή τα δεδομένα και που πρόκειται να σταλεί η όποια έξοδος
 3. **Πληροφορίες πόρων (resource information)**, → εκτίμηση των πόρων που χρειάζονται για να εκτελεστεί η λειτουργία και χρησιμοποιούνται από τον χρονοπρογραμματιστή (scheduler)
 4. **Ταυτότητα του χρήστη (user identification)**, που καθορίζει ποιος είναι ο χρήστης και δίνει τα διαπιστευτήριά του (π.χ. αριθμός λογαριασμού του χρήστη και το συνθηματικό (password)).

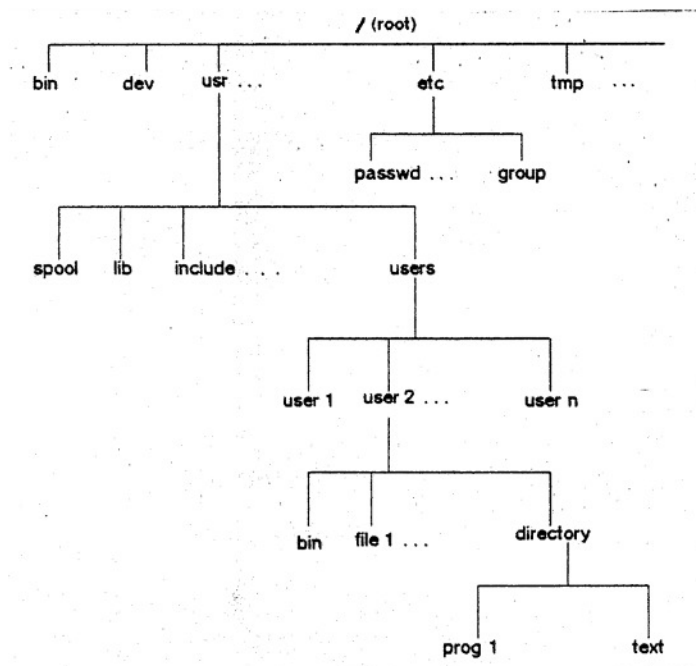


Σύστημα αρχειοθέτησης



Εντολές UNIX

- Διεργασίες: `ps -ef`
- `history`
- Navigating
- `pwd`
- `ls`
- `more`
- `mv`
- `cp`
- `chmod`
- `cat <test > result`
- Meta – characters
- `Ssh -l <username> helios.uop.gr`



Εντολές UNIX

1. Δημιουργήστε ένα φάκελο με το όνομα UNIX
2. Αντιγράψτε από το /etc το αρχείο passwd στο φάκελο UNIX
3. Δείτε ποιος είναι ο ιδιοκτήτης του αρχείου
4. Αλλάξτε τα δικαιώματα στο αρχείο passwd που έχετε αντιγράψει έτσι ώστε να μπορεί να διαβάσει και να γράψει μόνο ο ιδιοκτήτης του αρχείου
5. Να διαβάσετε με μία εντολή τα περιεχόμενα του αρχείου passwd και να τα βάλετε ταξινομημένα σε ένα νέο αρχείο με το όνομα result χρησιμοποιώντας την κατάλληλη εντολή UNIX
6. Να πάρετε το περιεχόμενο του result και να μετατρέψετε όλα τα μικρά γράμματα σε κεφαλαία με μία εντολή του UNIX. Το αποτέλεσμα να αποθηκευτεί σ' ένα νέο αρχείο με το όνομα result1
7. Στο αρχείο result να αντικαταστήσετε με μια εντολή του UNIX όπου υπάρχει το γράμμα a με το γράμμα X

Εντολές UNIX

- `cat <passwd`
- `cat <passwd |sort >result`
- `tr '[:lower:]' '[:upper:]' < result > result1`
- `tr 'a' 'X' < result > result2`

<https://ryanstutorials.net/bash-scripting-tutorial/bash-variables.php>