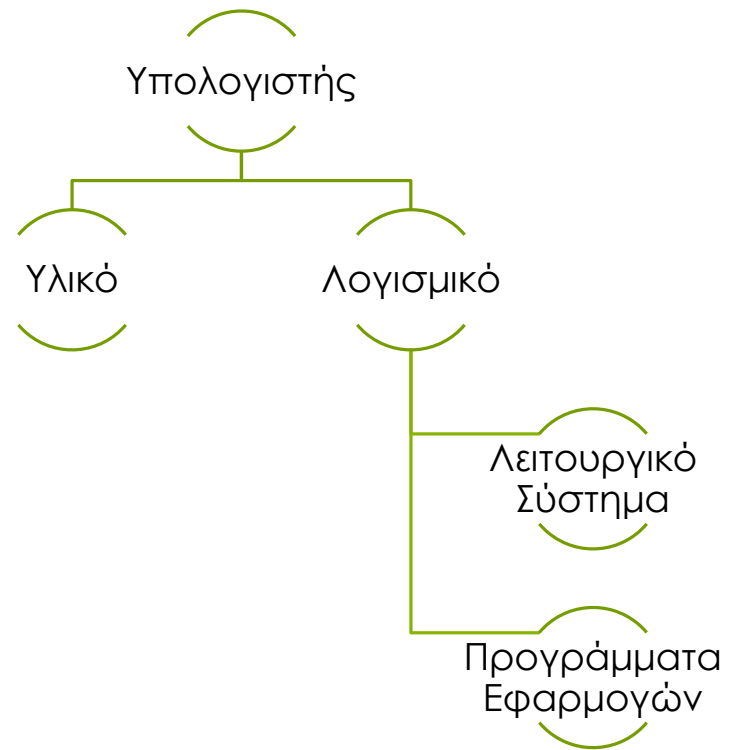


# Αρχιτεκτονικές Πληροφοριακών Συστημάτων

Λειτουργικά Συστήματα

# Λογισμικό Υπολογιστών



# Λειτουργικό Σύστημα (Operating System)

- **Λειτουργικό Σύστημα (ΛΣ)** είναι ένα σύνολο προγραμμάτων που διευκολύνει την εκτέλεση άλλων προγραμμάτων που επιθυμούν να χρησιμοποιήσουν τους πόρους του υπολογιστή. Επιτρέπει την διασύνδεση μεταξύ του χρήστη και του υλικού του υπολογιστή και ελέγχει την λειτουργία του υπολογιστή συνολικά.



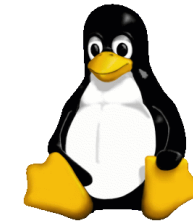
# Διαδεδομένα ΛΣ

- Windows
  - Windows 2003
  - Windows XP
  - Windows Vista
  - Windows 7
  - Windows 8
  - Windows 10
- Unix
  - Solaris
  - AIX
- Apple
  - OS X

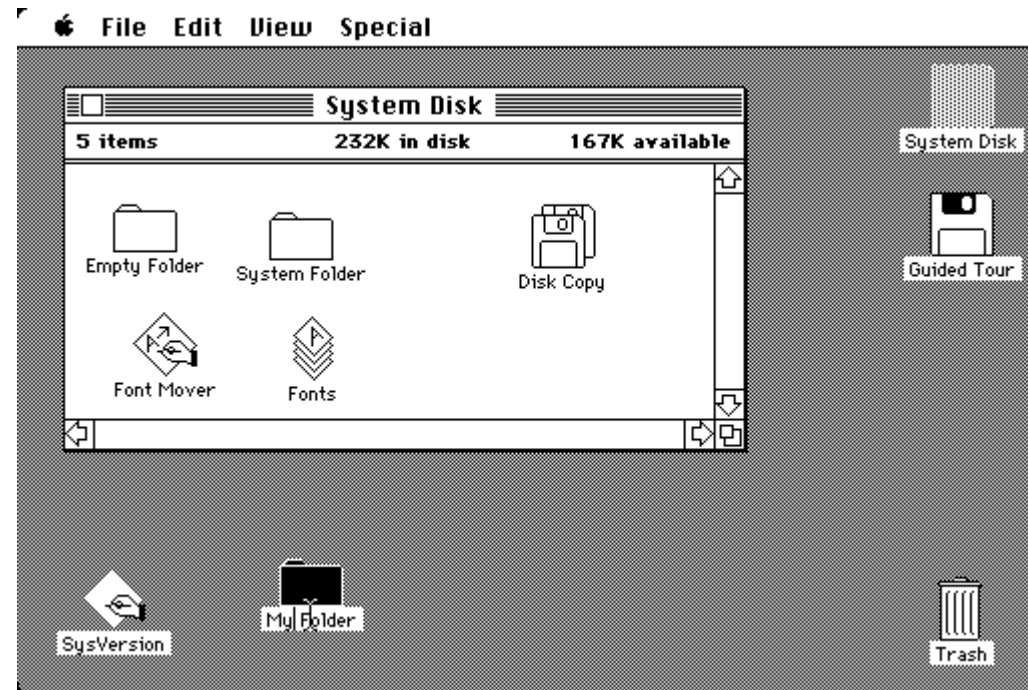


Mac OS

- Linux
  - Ubuntu
  - Xubuntu
  - Red Hat
  - Suse
  - Debian
  - Scientific Linux
  - Bodhi Linux
  - ...
- FreeBSD
- PC-BSD
- Haiku
- FreeDos



# Επιφάνεια εργασίας του Macintosh το 1984



# ΛΣ για κινητά και για ταμπλέτες

- iOS
- Android
- ~~Windows Mobile~~
- Symbian



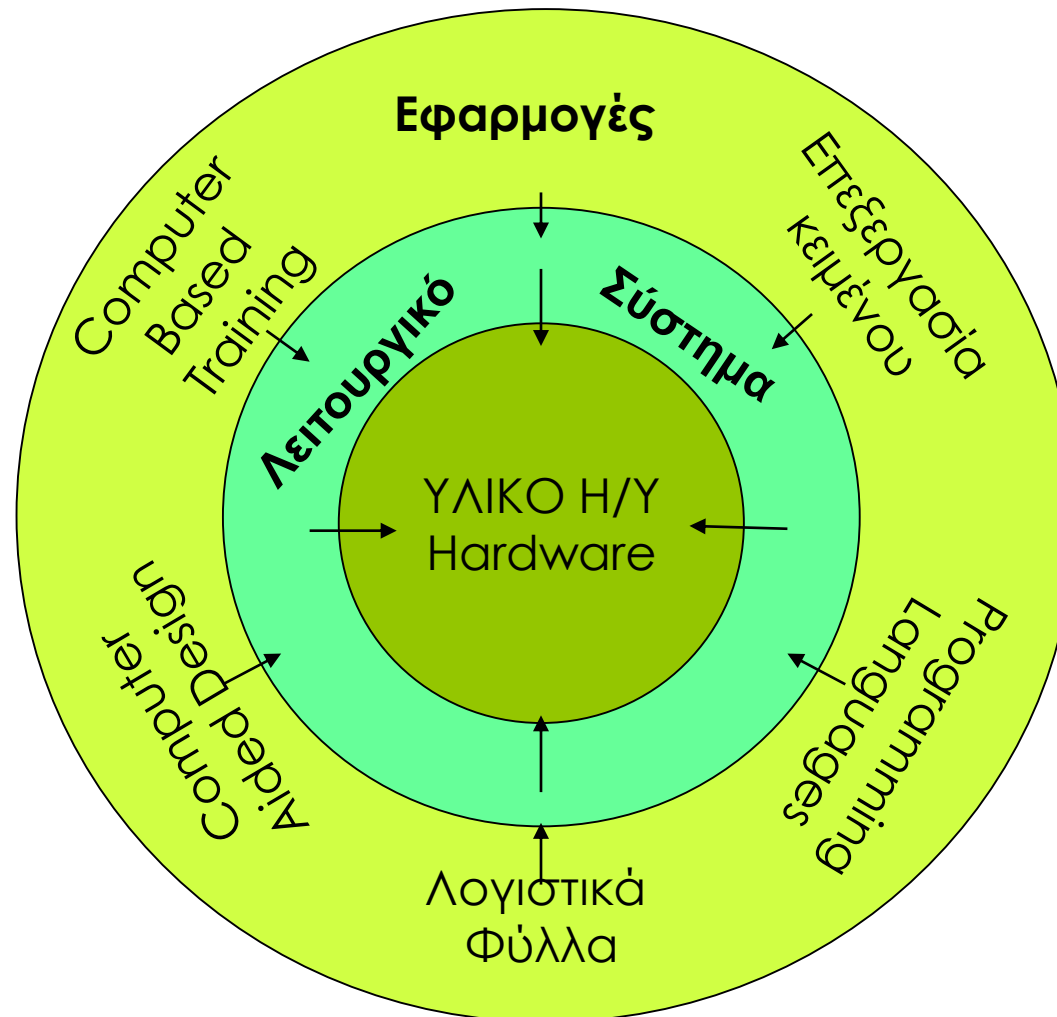
symbian  
OS

iOS

# Διάγραμμα Υλικού και Λογισμικού

Στόχοι Λ.Σ.:

- Αποδοτική χρήση του υλικού
- Ευκολία στην χρήση των πόρων



# Πολυπλοκότητα ΛΣ

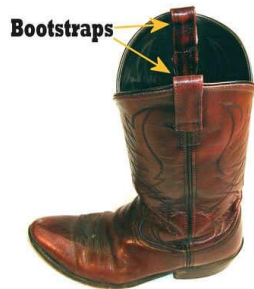
- SLOC  
(Source Lines of Code)  
αριθμός γραμμών κώδικα

Λειτουργικό Σύστημα	SLOC
Windows10	70 εκατομμύρια
MAC OS X 10.4	86 εκατομμύρια
FreeBSD	8,8 εκατομμύρια
Linux kernel 2.6.35	13,5 εκατομμύρια



# Διαδικασία εκκίνησης Η/Υ

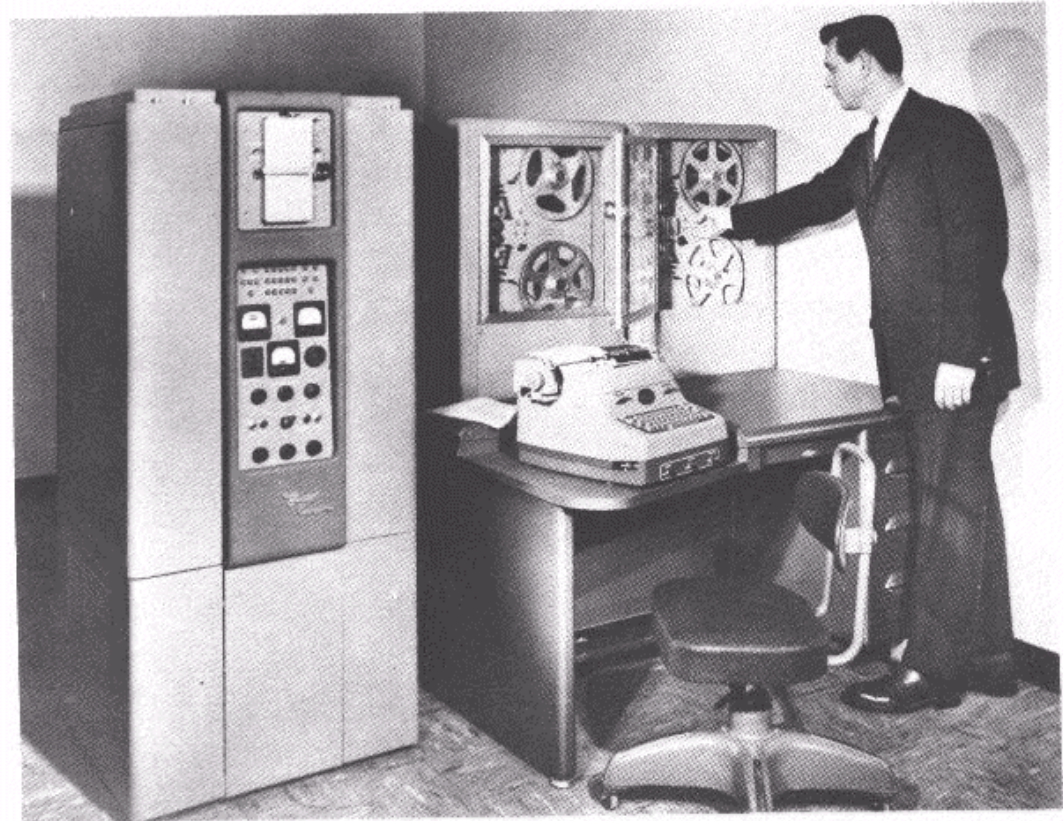
1. Εκτελείται το πρόγραμμα εκκίνησης.
2. Φορτώνεται το Λειτουργικό Σύστημα από μια δευτερεύουσα μονάδα αποθήκευσης (π.χ. σκληρό δίσκο) στην μνήμη RAM
3. Εκτελείται το Λειτουργικό Σύστημα



Το πρόγραμμα εκκίνησης (bootstrap) βρίσκεται στην μνήμη ROM του Η/Υ.

# Εξέλιξη των ΛΣ

- **Συστήματα δέσμης (batch operating systems-1950):** Μια εργασία το πολύ υπήρχε ανά πάσα στιγμή στο σύστημα
- **Συστήματα χρονομερισμού (time sharing).** Κάθε εργασία έχει την δυνατότητα δέσμευσης ενός πόρου για ένα μέρος του χρόνου. Λόγω της ταχύτητας εκτέλεσης των επιμέρους ενεργειών δίνεται η εντύπωση ότι το σύστημα εξυπηρετεί πολλές εργασίες ταυτόχρονα
- **Προσωπικά Συστήματα:** ΛΣ ενός χρήστη

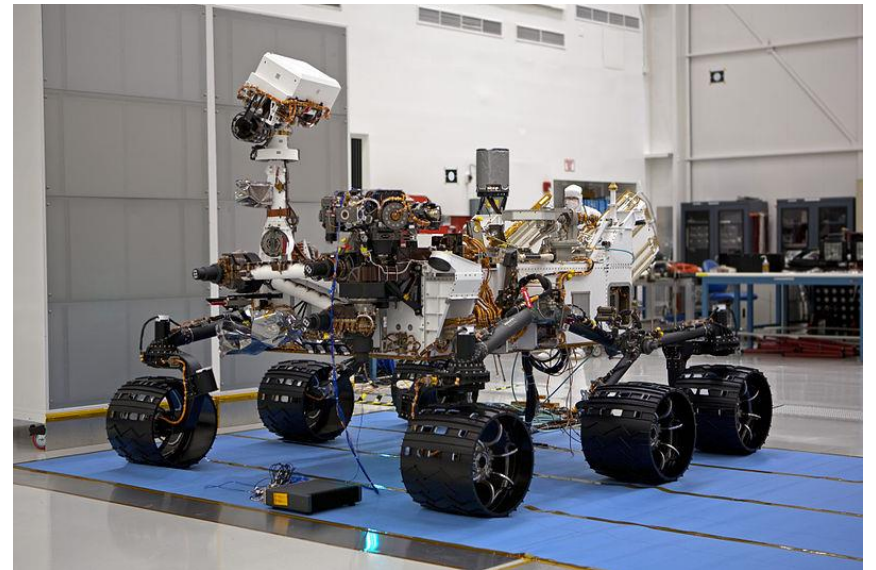


# Πολυπρογραμματισμός

- Διατήρηση πολλών διεργασιών **ταυτόχρονα** στην μνήμη του συστήματος
- Αντιστοίχιση ενός πόρου σε μια διεργασία όταν τον χρειάζεται και είναι διαθέσιμος
- **Πόροι (resources):** ΚΜΕ, Μνήμη, Σκληρός Δίσκος, Συσκευές εισόδου-εξόδου
- **Διεργασία:** Πρόγραμμα που βρίσκεται στη μνήμη και περιμένει για πόρους

# Ειδικές κατηγορίες ΛΣ

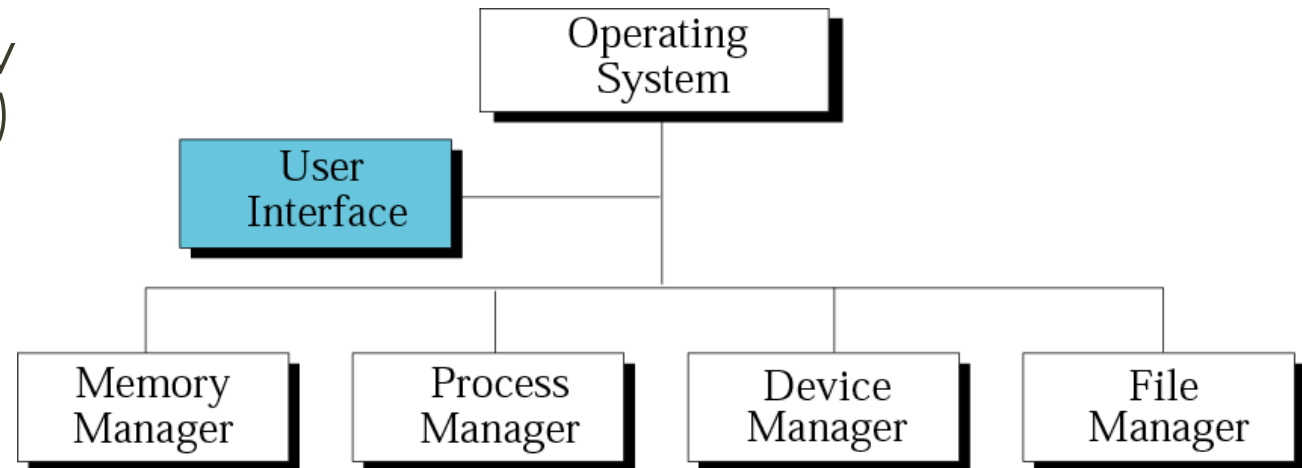
- **Παράλληλα Συστήματα:** Αφορούν υπολογιστές με πολλές Κεντρικές Μονάδες Επεξεργασίας που δίνουν την δυνατότητα της πραγματικής ταυτόχρονης εκτέλεσης πολλών εργασιών. Παρουσιάζουν αυξημένη πολυπλοκότητα
- **Κατανεμημένα ΛΣ:** Μια εργασία μπορεί να εκτελείται χρησιμοποιώντας πόρους διαφόρων υπολογιστών που βρίσκονται κατανεμημένοι σε ένα ευρύτερο δίκτυο
- **Συστήματα πραγματικού χρόνου:** Ο σχεδιασμός τους επιτρέπει να εκτελούν μια εργασία εντός συγκεκριμένου χρονικού πλαισίου (π.χ. VxWorks)



Mars Science Laboratory Curiosity rover  
Χρησιμοποιεί το ΛΣ VxWorks

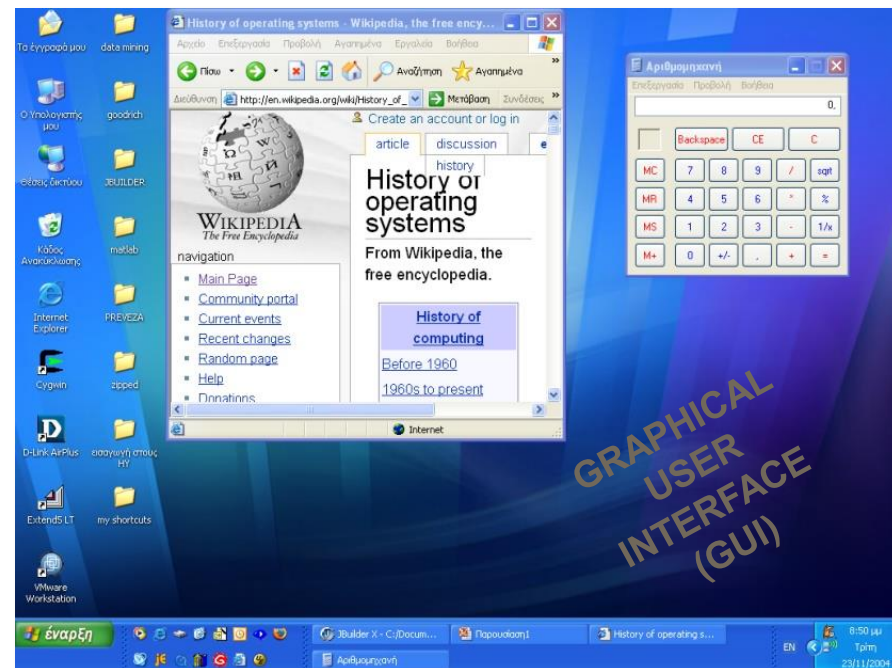
# Βασικές λειτουργίες των ΛΣ

1. Διαχείριση μνήμης  
(Memory management)
2. Διαχείριση διεργασιών  
(Process management)
3. Διαχείριση συσκευών  
(Device Manager)
4. Διαχείριση αρχείων  
(File Manager)



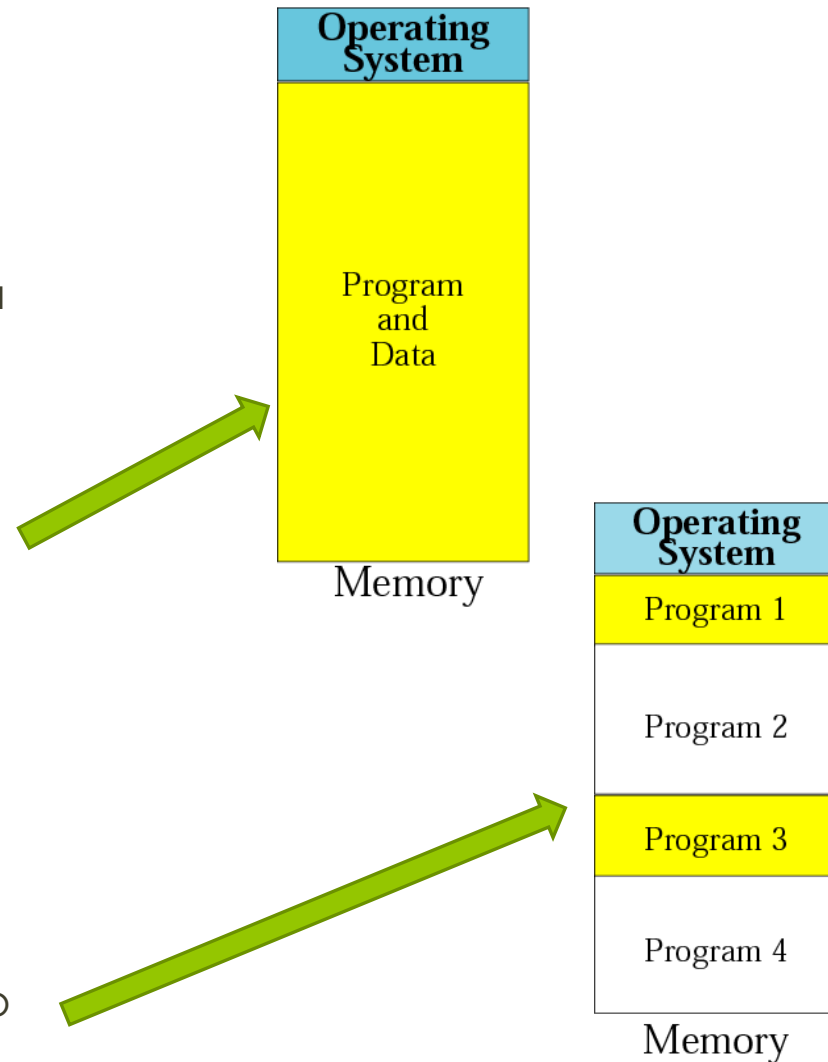
# Διεπαφή χρήστη

- Η διεπαφή χρήστη είναι ένα σύνολο προγραμμάτων που δέχονται αιτήσεις από τους χρήστες και ενεργοποιεί τα κατάλληλα προγράμματα για την διεκπεραίωση τους
- Η διεπαφή χρήστη μπορεί να είναι:
  - Γραφική διασύνδεση χρήστη (GUI)
  - Διεπαφή χρήστη γραμμής εντολών (shell)



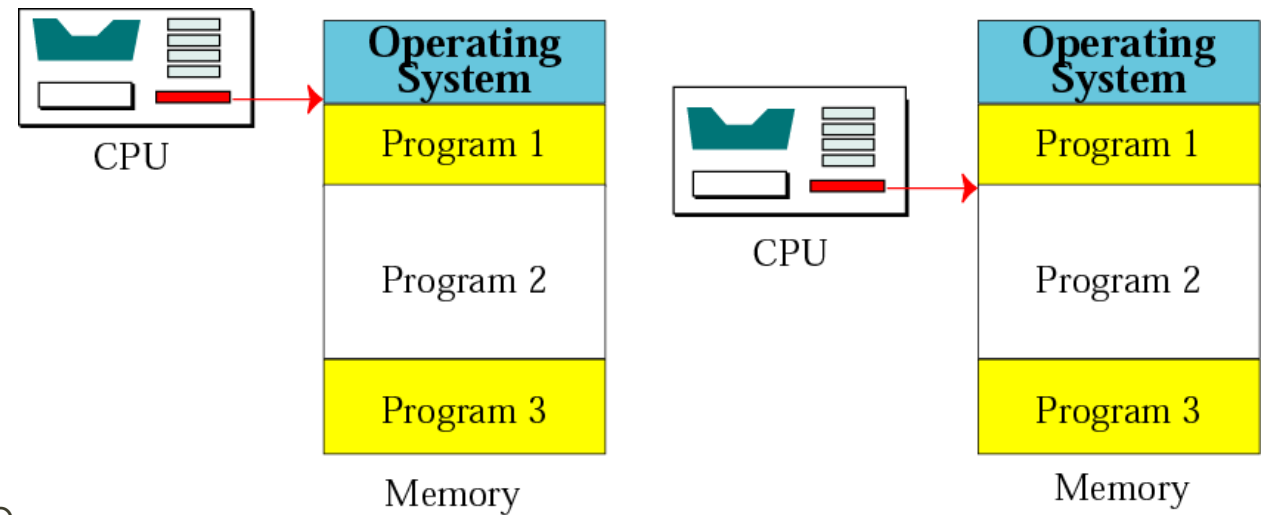
# Διαχείριση μνήμης

- Η μνήμη του υπολογιστή πρέπει να υπόκειται σε συνεχή διαχείριση έτσι ώστε να αποφεύγεται το σύνδρομο της ανεπαρκούς μνήμης.
- Κατηγορίες ΛΣ σε σχέση με την διαχείριση μνήμης
  - **Μονοπρογραμματιστικά.** Ένα μόνο πρόγραμμα υπάρχει στην μνήμη μαζί με το ΛΣ. Όταν ολοκληρώσει την εκτέλεσή του παραχωρεί την θέση του στο επόμενο πρόγραμμα προς εκτέλεση. (**Δεν χρησιμοποιείται πια**)
  - **Πολυπρογραμματιστικά.** Υπάρχουν στην μνήμη πολλά προγράμματα που εκτελούνται παράλληλα. Η ΚΜΕ περνάει από το ένα πρόγραμμα στο άλλο.



# Διαμέριση (partitioning)

- Η μνήμη χωρίζεται σε τμήματα (**διαμερίσεις = partitions**) μεταβλητού μήκους.
- Κάθε διαμέριση φιλοξενεί ένα πρόγραμμα
- Η ΚΜΕ περνάει από πρόγραμμα σε πρόγραμμα εκτελώντας κάποιες εντολές του μέχρι να συναντήσει κάποια εντολή εισόδου-εξόδου ή να λήξει ο χρόνος που έχει δεσμευτεί για το συγκεκριμένο πρόγραμμα. Αποθηκεύει την διεύθυνση της θέσης μνήμης στην οποία βρισκόταν και προχωράει στο επόμενο πρόγραμμα
- Η διαδικασία επαναλαμβάνεται κυκλικά



a. CPU starts executing program 1.      b. CPU starts executing program 2.

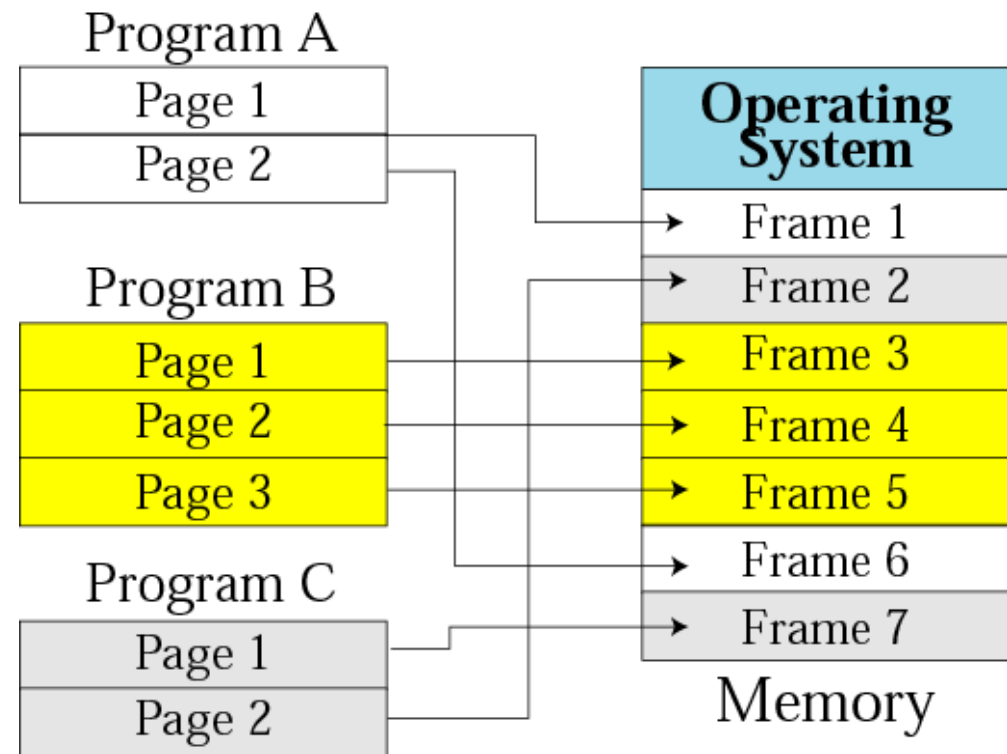


# Προβλήματα διαμέρισης

- Ο διαχειριστής μνήμης πρέπει να **προϋπολογίσει** το μέγεθος μνήμης που θα διαθέσει σε κάθε πρόγραμμα.
- Κατά την ολοκλήρωση εκτέλεσης των προγραμμάτων και την αντικατάστασή τους από άλλα η διαμέριση της μνήμης μπορεί να είναι τέτοια που να αφήνει κενά στην μνήμη (fragments) που να μην μπορούν να δεχθούν τα νέα προγράμματα αν δεν προηγηθεί **ανασυγκρότηση** της μνήμης (defragmentation)

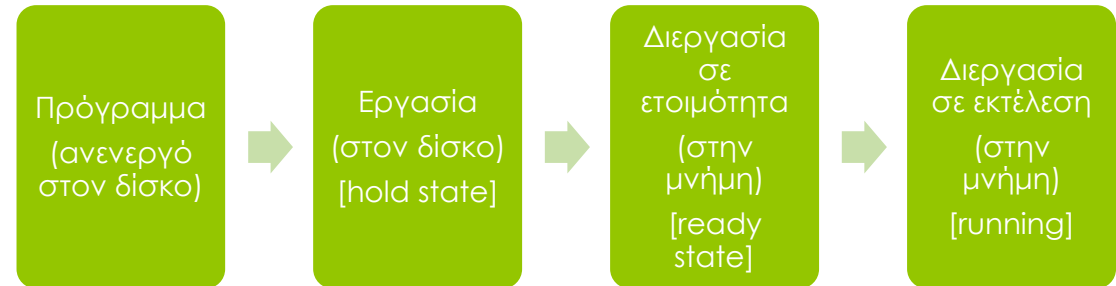
# Σελιδοποίηση (paging)

- Βελτιώνει την αποδοτικότητα της διαμέρισης
- Η μνήμη διαιρείται σε ισομεγέθη τμήματα που ονομάζονται **σελίδες**.
- Τα προγράμματα **δεν χρειάζεται** να καταλαμβάνουν συνεχόμενες θέσεις μνήμης
- Τα νέα προγράμματα δεν χρειάζεται να περιμένουν μέχρι να ελευθερωθεί μνήμη ίση με το μέγεθός τους σε συνεχόμενες θέσεις.
- Μπορούν να φορτωθούν στην μνήμη καλύπτοντας κενά μεταξύ των άλλων προγραμμάτων που ήδη τρέχουν



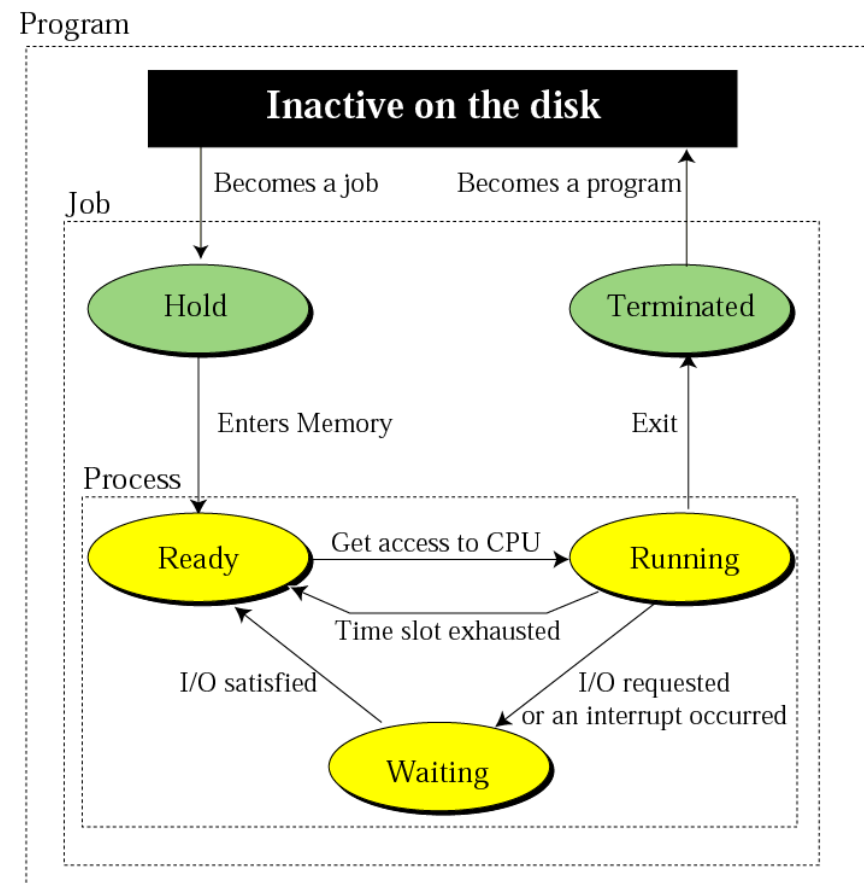
# Διαχείριση διεργασιών

- **Πρόγραμμα** (program): Ανενεργό σύνολο εντολών οι οποίες έχουν γραφεί από έναν προγραμματιστή και έχουν αποθηκευτεί σε κάποιο μέσο αποθήκευσης (π.χ. σκληρό δίσκο)
- **Εργασία** (job). Ένα πρόγραμμα μετατρέπεται σε εργασία όταν επιλεγεί για εκτέλεση.
- **Διεργασία** (process). Είναι ένα πρόγραμμα σε εκτέλεση. Πρόκειται για μια εργασία η οποία βρίσκεται στην μνήμη έχει ξεκινήσει αλλά δεν έχει ολοκληρωθεί.



## Διάγραμμα καταστάσεων με τα όρια μεταξύ προγράμματος, εργασίας, διεργασίας

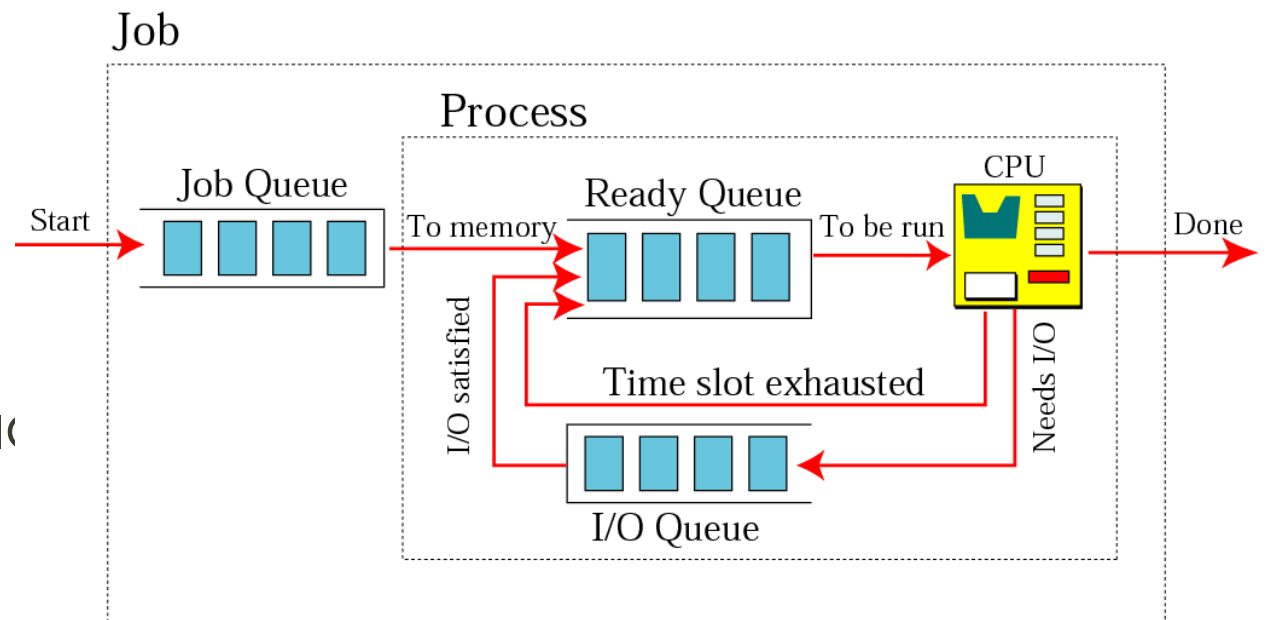
- Καταστάσεις διεργασιών
  - έτοιμη (ready)
  - σε εκτέλεση (running)
  - σε αναμονή (waiting)
- Όταν η διεργασία είναι σε κατάσταση **εκτέλεσης**:
  - Η διεργασία εκτελείται μέχρι να χρειαστεί είσοδο/έξοδο οπότε εισέρχεται σε κατάσταση αναμονής
  - Η διεργασία καταναλώνει όλο το χρονομερίδιο που της αναλογεί οπότε επιστρέφει σε κατάσταση ετοιμότητας
  - Η διεργασία τερματίζεται οπότε περνά σε κατάσταση τερματισμού



## Χρονοπρογραμματιστές - Ουρές

- Σε ένα ΛΣ υπάρχουν διάφοροι χρονοπρογραμματιστές
- Για παράδειγμα ο χρονοπρογραμματιστής διεργασιών αποφασίζει για την μετάβαση των διεργασιών από την μια κατάσταση στην άλλη.

- Ουρές = Λίστες αναμονής

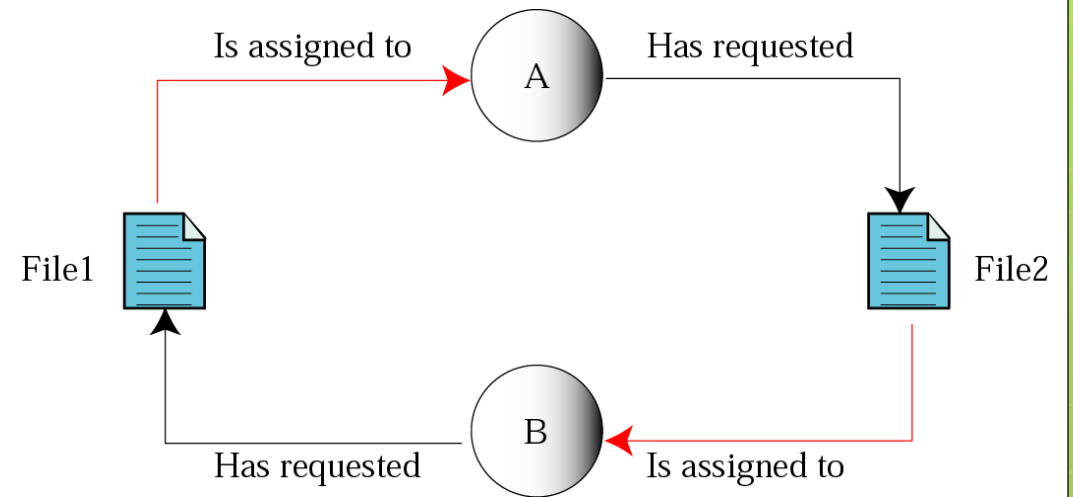


# Πολιτικές επιλογής διεργασιών προς εκτέλεση

- Ο χρονοπρογραμματιστής διεργασιών μπορεί να έχει διάφορες πολιτικές για την επιλογή της επόμενης διεργασίας που θα εκτελεστεί από την ουρά.
  - FCFS (First Come First Serve)
  - SJF (Shortest Job First)

# Αδιέξοδο (deadlock)

- Ένα σύνολο από διεργασίες βρίσκονται σε αδιέξοδο όταν κάθε διεργασία του συνόλου περιμένει για ένα γεγονός το οποίο μόνο κάποια άλλη διεργασία του συνόλου μπορεί να προκαλέσει
- Τα αδιέξοδα συμβαίνουν όταν δίνεται η δυνατότητα αποκλειστικής πρόσβασης σε διάφορους πόρους



Αδιέξοδο συμβαίνει όταν το ΛΣ δεν θέτει επαρκείς περιορισμούς στις διεργασίες όσον αφορά την χρήση των πόρων.

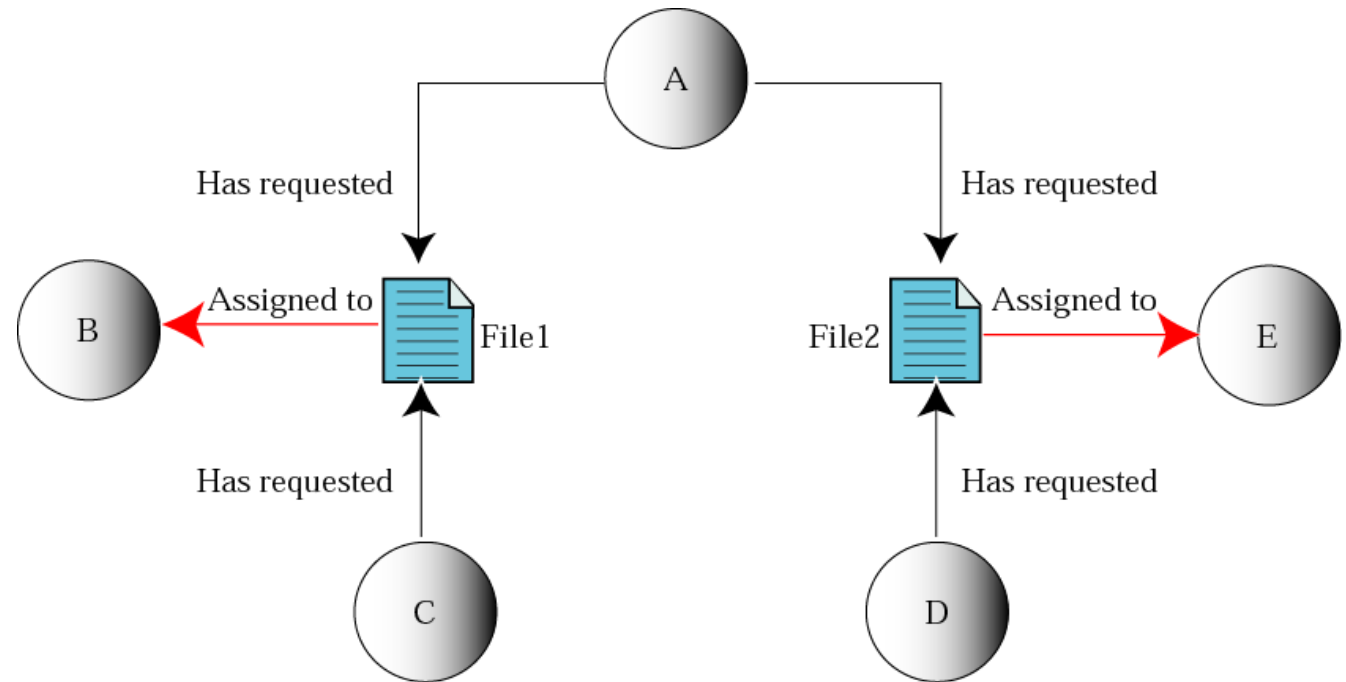
# Προϋποθέσεις για να υπάρξει αδιέξοδο

- Οι ακόλουθες **4 συνθήκες** πρέπει να ισχύουν ταυτόχρονα έτσι ώστε να μπορεί να συμβεί αδιέξοδο
  - **Αμοιβαίος αποκλεισμός** (ένας πόρος μπορεί να ανατεθεί σε μια μόνο διεργασία)
  - **Παρακράτηση πόρων** (μια διεργασία που κρατά ένα πόρο μπορεί να ζητήσει και άλλους πόρους)
  - **Μη προεκτοπιστική κατανομή πόρων** (πόροι που έχουν διατεθεί σε μια διεργασία δεν μπορούν να παρθούν με την βία πίσω)
  - **Κυκλική αναμονή** (θα πρέπει να υπάρχει μια αλυσίδα από 2 ή περισσότερες διεργασίες που η κάθε μια περιμένει ένα πόρο από την αμέσως επόμενη διεργασία στην αλυσίδα)
- Για να λυθεί ένα αδιέξοδο αρκεί να αναιρεθεί μια από τις 4 συνθήκες.



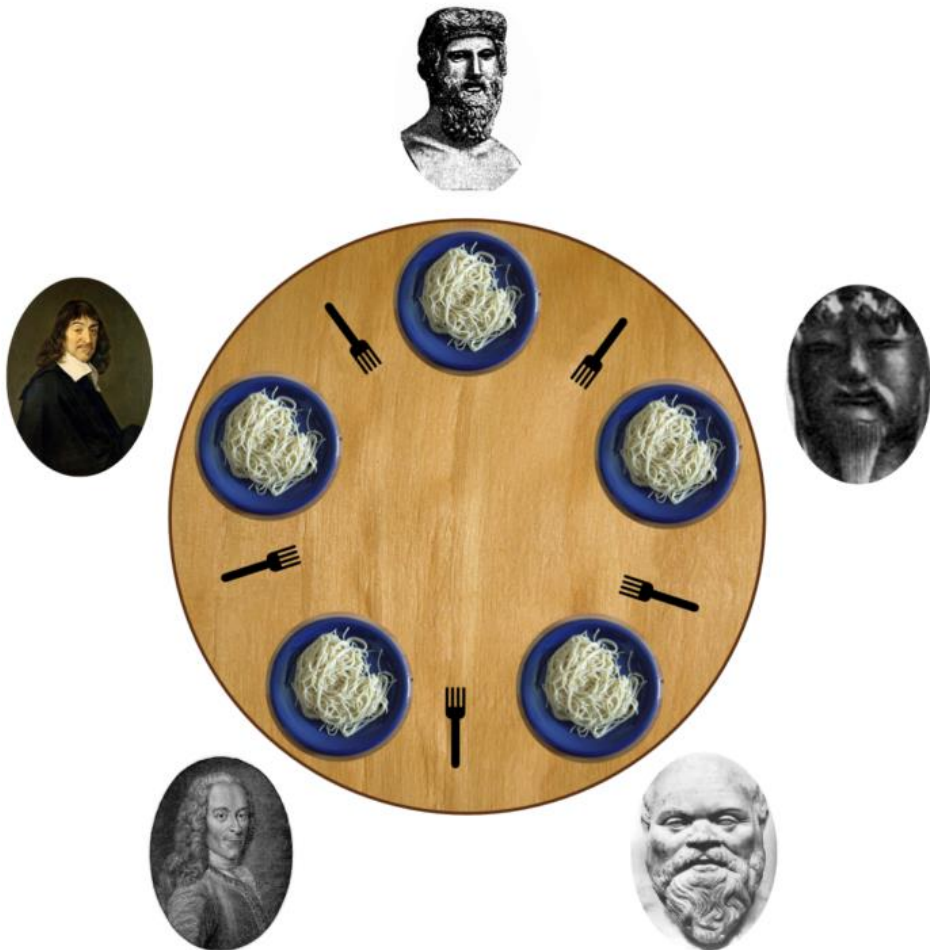
# Λιμοκτονία (starvation)

- Συμβαίνει όταν το ΛΣ θέτει πολλούς περιορισμούς στις διεργασίες όσον αφορά την χρήση των πόρων



a. Process A needs both File1 and File2.

# Δειπνούντες φιλόσοφοι



- 5 φιλόσοφοι κάθονται σε στρογγυλό τραπέζι και καθένας θα πρέπει να έχει στην διάθεσή του 2 ξυλάκια (αριστερά και δεξιά από το πιάτο του) έτσι ώστε να φάει το φαγητό που έχει μπροστά του
- Αν έχει διαθέσιμο 1 μόνο ξυλάκι τότε περιμένει χωρίς να το δεσμεύει
- Κάποιος από τους φιλοσόφους μπορεί να λιμοκτονήσει καθώς ενδέχεται να μην είναι ποτέ και τα 2 ξυλάκια ταυτόχρονα διαθέσιμα

# Διαχείριση συσκευών

- Ο διαχειριστής συσκευών (**device manager**) είναι υπεύθυνος για την προσπέλαση των συσκευών εισόδου-εξόδου (I/O)
- Οι συσκευές I/O είναι τάξεις μεγέθους **πιο αργές** από την ΚΜΕ και την μνήμη
- Καθήκοντα διαχειριστή συσκευών
  - Παρακολουθεί συνεχώς κάθε συσκευή I/O ώστε να εξασφαλίζει ότι λειτουργεί σωστά.
  - Διατηρεί ουρά με τις διεργασίες που περιμένουν εξυπηρέτηση από κάθε συσκευή την οποία ενημερώνει σύμφωνα με την πρόοδο των διεργασιών.
  - Ελέγχει τις πολιτικές πρόσβασης για την προσπέλαση στις συσκευές I/O

# Διαχείριση αρχείων

- Καθήκοντα διαχειριστή αρχείων (**File Manager**)
  - Ελέγχει την πρόσβαση στα αρχεία επιτρέποντας πρόσβαση μόνο σε όσους έχουν την κατάλληλη άδεια (ανάγνωση, εγγραφή, εκτέλεση).
  - Επιβλέπει την δημιουργία, διαγραφή και τροποποίηση των αρχείων
  - Ελέγχει την ονομασία των αρχείων
  - Επιβλέπει την αποθήκευση των αρχείων (που και με ποια μορφή)
  - Είναι υπεύθυνος για την αρχειοθέτηση και την λήψη εφεδρικών αντιγράφων.

# Unix

- Σχεδιάστηκε το 1969 στα Bell Labs
- Είναι φορητό ΛΣ καθώς έχει γραφεί σε γλώσσα C και όχι σε γλώσσα μηχανής
- Διαθέτει ισχυρές εντολές που μπορούν να συνδυαστούν σε scripts
- Είναι ανεξάρτητο από συσκευές (περιέχει ενσωματωμένα προγράμματα οδήγησης)
- Διαθέτει μηχανισμούς ασφάλειας (ταυτοποίηση χρήστη, έλεγχος πρόσβασης)

# Βασικά συστατικά Unix

Κέλυφος  
(shell)

Βοηθητικά  
προγράμματα  
(utilities)

Εφαρμογές  
(applications)

Πυρήνας  
(kernel)

# Linux

- Κατασκευάστηκε το 1991 από τον τότε φοιτητή Linus Torvalds
- Κατασκευάστηκε με στόχο την αποδοτική εκτέλεση σε μικροεπεξεργαστές της εταιρείας Intel
- Ξεκίνησε ως υποσύνολο του UNIX
- Ο πυρήνας 2.0 του Linux που κυκλοφόρησε το 1997 έγινε αποδεκτός ως εμπορικό ΛΣ

