

# (Μικρο)Πυρήνας

## Λειτουργίες

- υλοποίηση διεργασιών
- εξυπηρέτηση διακοπών και παγίδων
- υλοποίηση μηχανισμών συγχρονισμού και επικοινωνίας
- διανομή των διεργασιών στην ΚΜΕ

## Χαρακτηριστικά

- δεν είναι επανεισαγόμενος  
(τα προγράμματα του πυρήνα δε μοιράζονται σε περισσότερες από μία διεργασίες)
- λειτουργία σε κατάσταση συστήματος (προνομιούχο κατάσταση) με
- απαγόρευση διακοπών

# (Μικρο)Πυρήνας

## Ενεργοποιείται όταν

- συμβεί μία (εξωτερική) διακοπή
- μία διεργασία προκαλέσει (εσωτερικά) μία παγίδα
- μία διεργασία εκτελέσει μία κλήση του συστήματος (πρωτογενή κλήση του πυρήνα)

## Αποτελείται από

- το σύστημα αντιμετώπισης των διακοπών
- το σύστημα αντιμετώπισης των παγίδων και των κλήσεων του επόπτη
- το διανομέα

# Οργάνωση διεργασιών

## Ιεραρχική οργάνωση δένδρου

- *οικογενειακό δένδρο (family tree)* ή δένδρο δημιουργίας (creation tree) ή *ομάδα διεργασιών ενός χρήστη (process group)*
- *διεργασία γονέας (parent process)*
- *διεργασίες παιδιά (children processes)* - αδελφοί (brothers) – πρόγονοι (ancestors) - απόγονοι (descendants)
- Μία διεργασία που αναλαμβάνει ένα έργο δημιουργεί διεργασίες παιδιά κ.ο.κ.
- Οι διεργασίες παιδιά εκτελούν μέρη του έργου
- (κατα)μερισμός εργασίας (division of labour)

# Οργάνωση διεργασιών

**ρίζα (root) δένδρου ή διερμηνευτής εντολών (command interpreter) ή φλοιός (shell)**

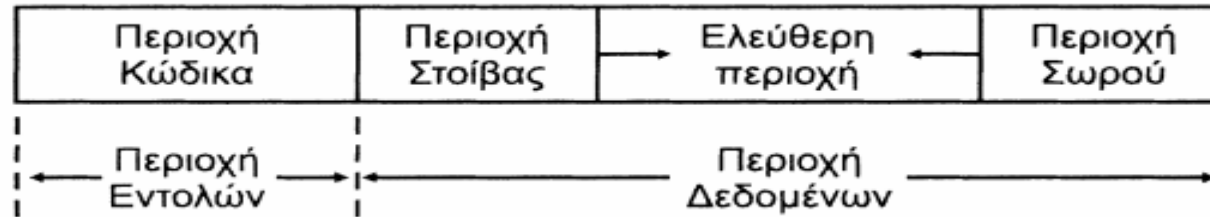
- δημιουργεί διεργασίες-παιδιά για την εκτέλεση κάθε εντολής που πληκτρολογεί ο χρήστης, οι διεργασίες-παιδιά δημιουργούν διεργασίες-εγγόνια κ.ο.κ. μέχρις ότου να εκτελεστεί πλήρως το έργο
- οι διεργασίες πρόγονοι ελέγχουν συνήθως τους απογόνους τους

# Οργάνωση διεργασιών

## Πλεονεκτήματα ιεραρχικής οργάνωσης

- απλούστευση μηχανισμών επικοινωνίας
  - επικοινωνία μέσω κλάδων οικογενειακού δένδρου
- έλεγχος καταχώρισης πόρων
  - οι διεργασίες παιδιά αποκτούν πόρους που τους παρέχουν οι διεργασίες-γονείς
  - οι διεργασίες-γονείς προγραμματίζουν χρονικά τα παιδιά τους
- λογιστικοί λόγοι
- σχετικά ελεύθερη επικοινωνία
- ανεξάρτητος χρονοπρογραμματισμός

# Εικόνα Διεργασιών



## περιοχή (τμήμα) κώδικα

- αποθήκευση επανεισαγόμενου κώδικα (εντολών) διεργασίας (μπορεί να μοιραστεί σε περισσότερες από μία διεργασίες)
- μόνο διαβάζεται και εκτελείται
- μέγεθος σταθερό και γνωστό μετά τη μεταγλώττιση και/ή σύνδεση του συστήματος

# Εικόνα Διεργασιών

## περιοχή (τμήμα) στοίβας

- αποθήκευση εγγραφών ενεργοποίησης υποπρογραμμάτων
- γράφεται και διαβάζεται
- μέγεθος μεταβλητό, δεν είναι γνωστό προκαταβολικά

## περιοχή (τμήμα) σωρού

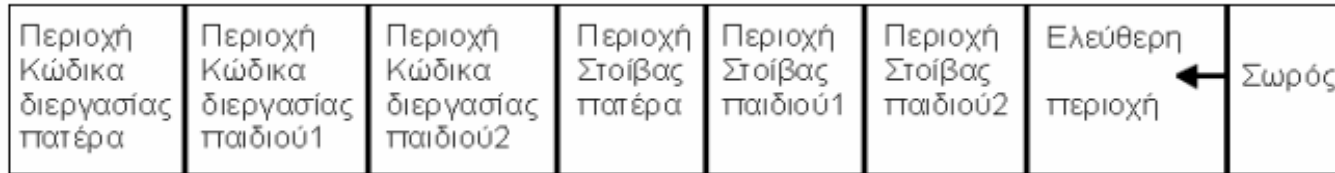
- θέσεις της μνήμης που καταχωρούνται και ελευθερώνονται δυναμικά μέσω δεικτών
- γράφεται και διαβάζεται
- μέγεθος μεταβλητό, δεν είναι γνωστό προκαταβολικά

# Εικόνα Διεργασιών

## ελεύθερη περιοχή

- η περιοχή στοίβας αυξάνεται προς μία κατεύθυνση
- η περιοχή σωρού ξεκινά από το άλλο άκρο της μνήμης που έχει καταχωρηθεί για τη διεργασία και αυξάνεται προς την αντίθετη κατεύθυνση
- η διεργασία υπερβαίνει τη διαθέσιμη μνήμη της όταν τα όρια των δύο αυτών περιοχών συναντηθούν

# Εικόνα πολλών διεργασιών στη μνήμη



περιοχές επανεισαγόμενου κώδικα διεργασιών

## περιοχές στοίβας

- κάθε διεργασία έχει τη δική της (ιδιωτική) στοίβα
- όταν οι περιοχές στοίβας των απογόνων μιας διεργασίας καταχωρούνται από την περιοχή της δικής της στοίβας τότε η στοίβα του γονέα ονομάζεται στοίβα κάκτου (cactus stack)

# Εικόνα πολλών διεργασιών στη μνήμη

## περιοχή σωρού

- κάθε διεργασία πρέπει να προσπελάζει μόνο τις δικές της δυναμικές δομές
- αμοιβαίος αποκλεισμός προσπέλασης του σωρού

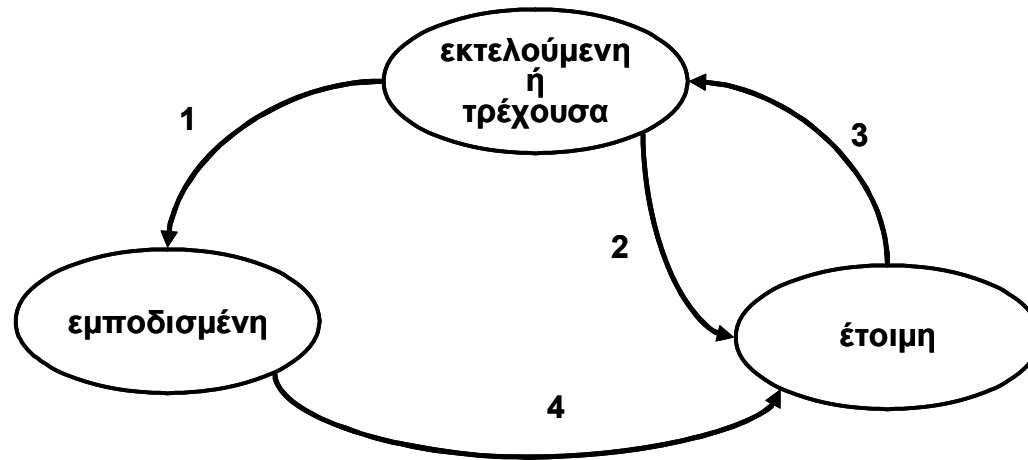
## ελεύθερη περιοχή

- ο μηδενισμός της δεν είναι απαραίτητα σφάλμα
- το σύστημα μπορεί να αναστείλει μια διεργασία μέχρις ότου μία άλλη να ελευθερώσει μία από τις περιοχές που τις έχουν καταχωρηθεί

# Καταστάσεις διεργασιών (Γενικά)

Μία διεργασία μπορεί να είναι σε μια από τις παρακάτω τρεις καταστάσεις:

- **εκτελούμενη (running)** από την ΚΜΕ ή **τρέχουσα (current)**
- **έτοιμη (ready)** να εκτελεστεί από την ΚΜΕ
- **εμποδισμένη (blocked)** περιμένοντας να συμβεί ένα γεγονός



# Καταστάσεις διεργασιών (Γενικά)

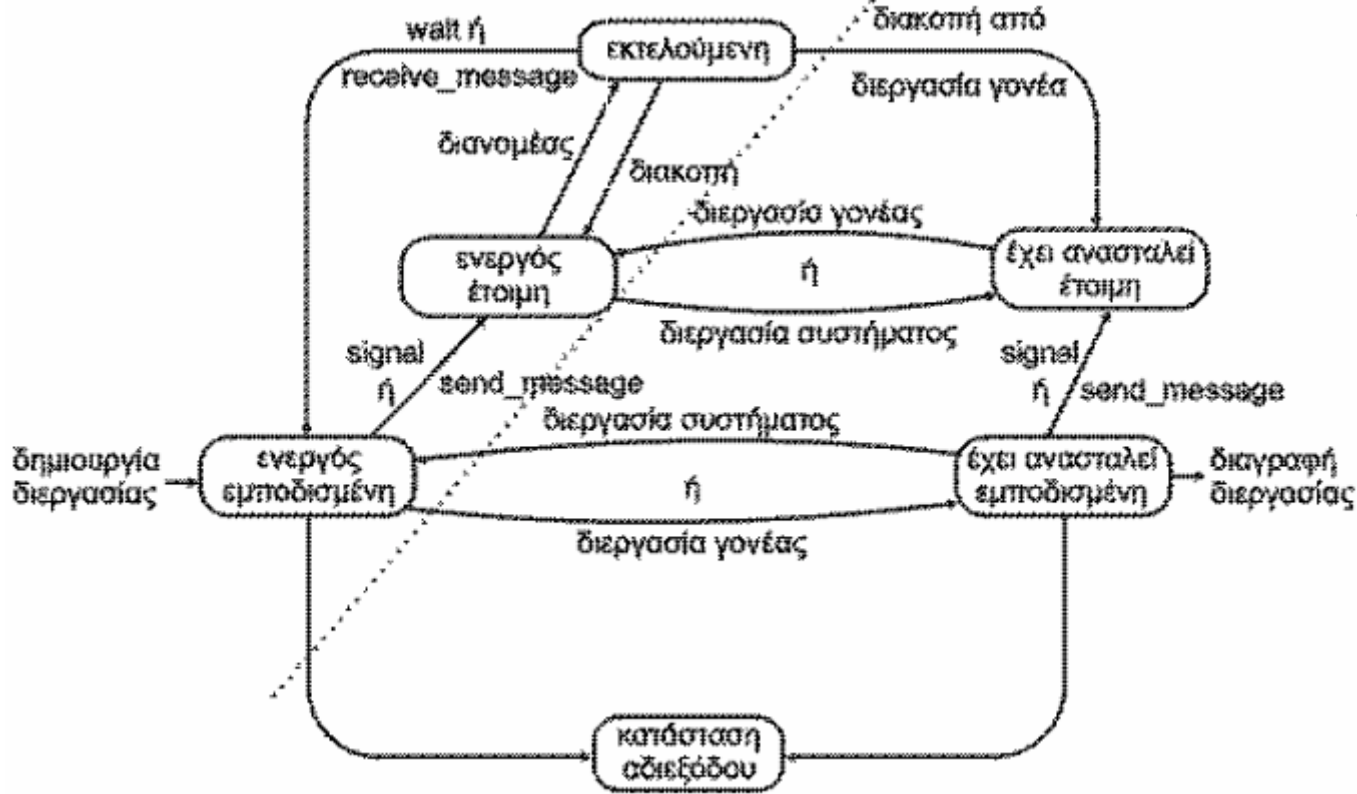
Τέσσερις μεταστάσεις (transitions):

1. η διεργασία έχει τελειώσει την εκτέλεσή της ή εμποδίζεται να προχωρήσει περιμένοντας να συμβεί ένα γεγονός
2. η διεργασία εξάντλησε το κβάντο χρόνου της
3. ο διανομέας επιλέγει την διεργασία και την (ξανά) διανέμει
4. η διεργασία απελευθερώνεται (συμβαίνει το γεγονός)

Μια εκτελούμενη ή έτοιμη διεργασία ονομάζεται **απελευθερωμένη (unblocked)**.

$$\begin{aligned} \blacksquare \text{ Σύνολο διεργασιών} &= \{\text{απελευθερωμένων}\} + \{\text{εμποδισμένων}\} \\ &= (\{\text{εκτελούμενων}\} + \{\text{ετοίμων}\}) + \{\text{εμποδισμένων}\} \end{aligned}$$

$$|\{\text{εκτελούμενων}\}| \leq \text{πλήθος επεξεργαστών}$$



## Καταστάσεις διεργασιών (Λεπτομερέστερα)

Μία διεργασία μπορεί επίσης να είναι **ενεργός (active)** ή **ανενεργός (inactive)**, δηλαδή να **έχει ανασταλεί (suspended)**. Εκτός των παραπάνω μεταστάσεων,

- μια ενεργός εκτελούμενη διεργασία μπορεί να μεταβεί η ίδια στις καταστάσεις ενεργός έτοιμη ή ενεργός εμποδισμένη
- μια ενεργός έτοιμη ή εμποδισμένη διεργασία μπορεί, εκτός των άλλων, να μεταβεί στις αντίστοιχες ανενεργές καταστάσεις, και αντίστροφα, από το γονέα της ή από το σύστημα. Τέλος
- μια (ενεργός ή ανενεργός) εμποδισμένη διεργασία μπορεί να καταλήξει στην κατάσταση του αδιεξόδου

# Καταστάσεις διεργασιών

*Γενικά:*

- **εμποδισμένη κατάσταση:** η ίδια η διεργασία έφερε τον εαυτό της στην κατάσταση αυτή
  - περιμένει ένα μήνυμα ή ένα γεγονός
- **κατάσταση έχει ανασταλεί:** το σύστημα, ο χρήστης στον οποίο ανήκει η διεργασία ή μια άλλη διεργασία ανάγκασε τη διεργασία να βρεθεί σ' αυτή την κατάσταση, π.χ.
  - διαλογική εκσφαλμάτωση
  - ^C από το χρήστη
  - λόγοι διαχείρισης μνήμης

# Καταστάσεις διεργασιών

<b>Εκτελέσιμη (runable)</b>	<b>Απελευθερωμένη (unblocked)</b>	<b>Εκτελούμενη (running) ή Τρέχουσα (current)</b>	<b>Ενεργός (active)</b>
		<b>Έτοιμη (ready)</b>	
<b>Μη εκτελέσιμη (unrunable)</b>	<b>Εμποδισμένη (blocked)</b>		<b>Έχει ανασταλεί (suspended)</b>
	<b>Έχει ανασταλεί έτοιμη (suspended ready)</b>		
	<b>Έχει ανασταλεί εμποδισμένη (suspended blocked)</b>		

## Ιστορία/κύκλος ζωής (life history/cycle) διεργασίας

- μια διεργασία-γονέας (απ)αιτεί από το σύστημα τη δημιουργία μιας διεργασίας-παιδιού
- όσο δημιουργείται η διεργασία παιδί αυτή βρίσκεται στην *εμποδισμένη κατάσταση (περιμένοντας τη δημιουργία της)*
- μόλις δημιουργηθεί γίνεται *έτοιμη* και ειδοποιείται ο γονέας της
- όταν ολοκληρωθεί η εκτέλεση της βρίσκεται στην *εμποδισμένη κατάσταση (περιμένοντας τη διαγραφή της από το σύστημα)* και ειδοποιείται ο γονέας της

# Ιστορία/κύκλος ζωής διεργασίας

## Δημιουργία διεργασιών

- Αρχικοποίηση συστήματος
- Αίτηση χρήστη για τη δημιουργία νέας διεργασίας
- Αίτηση διεργασίας-γονέα για τη δημιουργία διεργασίας-παιδιού με χρήση μιας κλήσης του συστήματος

## Κατανομή πόρων μεταξύ διεργασιών

- Διεργασία-γονέας και διεργασία παιδί μοιράζονται τους ίδιους πόρους
- η διεργασία-παιδί χρησιμοποιεί ένα υποσύνολο των πόρων της διεργασίας-γονέα
- διεργασία-γονέας και διεργασία-παιδί χρησιμοποιούν ανεξάρτητα μεταξύ τους διαφορετικούς πόρους

# Ιστορία/κύκλος ζωής διεργασίας

## Εκτέλεση διεργασιών

- Διεργασία-γονέας και διεργασία-παιδί εκτελούνται (ψευδο)ταυτόχρονα
- Η διεργασία-γονέας μπορεί να περιμένει μέχρι ότου τελειώσει η εκτέλεση της η διεργασία-παιδιού

## Διαγραφή διεργασιών

- Κανονική έξοδος (εθελοντική)
  - Η διεργασία εκτελεί την τελευταία εντολή της ή αποτυγχάνει
    - λόγω σφάλματος του προγράμματός της
  - Οι πόροι της διεργασίας επιστρέφονται στο γονέα της ή στο σύστημα

## Ιστορία/κύκλος ζωής διεργασίας

- Το σύστημα ή μια διεργασία-γονέας "σκοτώνει" μια διεργασία παιδί της (μη εθελοντική έξοδος) όταν:
  - Η διεργασία-παιδί έχει υπερβεί τους πόρους που της παραχωρήθηκαν
  - Ο χρήστης αποφασίσει τη διακοπή της (CTRL/C)
  - Συμβεί ένα καταστροφικό σφάλμα του συστήματος

## Περιβάλλον διεργασιών

- η εκτέλεση μιας διεργασίας από την ΚΜΕ μπορεί να διακόπτεται και να συνεχίζεται αργότερα. Αυτό επιτυγχάνεται με το να
- αποταμιεύονται οι απαραίτητες πληροφορίες για τη συνέχιση της εκτέλεσής της
- Οι πληροφορίες που είναι απαραίτητες για την εκτέλεση μιας διεργασίας αποτελούν το **πιητικό περιβάλλον (volatile environment)** ή την **ομάδα συμφραζομένων (context block)** της διεργασίας και περιέχει τα εξής:
  - κατάσταση προγράμματος (program state)
    - μετρητής προγράμματος
    - άλλοι καταχωρητές
  - πληροφορίες μνήμης
    - καταχωρητές βάσης, ορίου ή συσχέτισης
    - περιεχόμενα πίνακα τμημάτων και/ή σελίδων

# Περιβάλλον διεργασιών

- πληροφορίες ε/ε
  - ανοικτά αρχεία
  - ταυτότητες συσκευών ε/ε
  - διευθύνσεις των ενταμιευτών
- πληροφορίες διανομής
  - προτεραιότητα
  - δείκτες ουράς/ών διανομέα
- λογιστικές πληροφορίες
  - χρόνος ΚΜΕ
  - πόροι που χρησιμοποιήθηκαν
  - πόροι που απαιτούνται (αν είναι γνωστοί)
  - ταυτότητα του χρήστη
- κ.ά.

# Αποθήκευση ομάδας συμφραζομένων

- **Μόνιμα στην κύρια μνήμη**
  - Μεγάλη μνήμη
  - Λίγες διεργασίες
  - Μικρός χρόνος μεταγωγής
  - Γενικά: περιβάλλοντα λίγων διεργασιών
- **Στο δίσκο**
  - Κύριο τμήμα (master segment) ή τμήμα κατάστασης διεργασίας (*Task State Segment-TSS*, στους 80x86)
  - Φόρτωση στην κύρια μνήμη πριν αρχίσει η εκτέλεση της διεργασίας
- **Χρόνος μεταγωγής συμφραζομένων**
  - Χρόνος επιβάρυνσης
  - Εξαρτώμενος από την υποστήριξη του υλισμικού

# Περιγραφητής διεργασίας (Process descriptor) ή Ομάδα Ελέγχου Διεργασίας (Process Control Block, PCB)

Περιλαμβάνει:

- το περιβάλλον
  - το ίδιο αν είναι αποθηκευμένο στην κύρια μνήμη
  - τη διεύθυνση του κύριου τμήματος αν αυτό φορτώνεται από τη βοηθητική μνήμη
- την ταυτότητα της διεργασίας
- την κατάστασή της
- πληροφορίες συγχρονισμού και επικοινωνίας
- χώρο διευθύνσεων (κώδικα και δεδομένων)
- κ.ά.

# Περιγραφητής Διεργασίας ή Ομάδα Ελέγχου της Διεργασίας

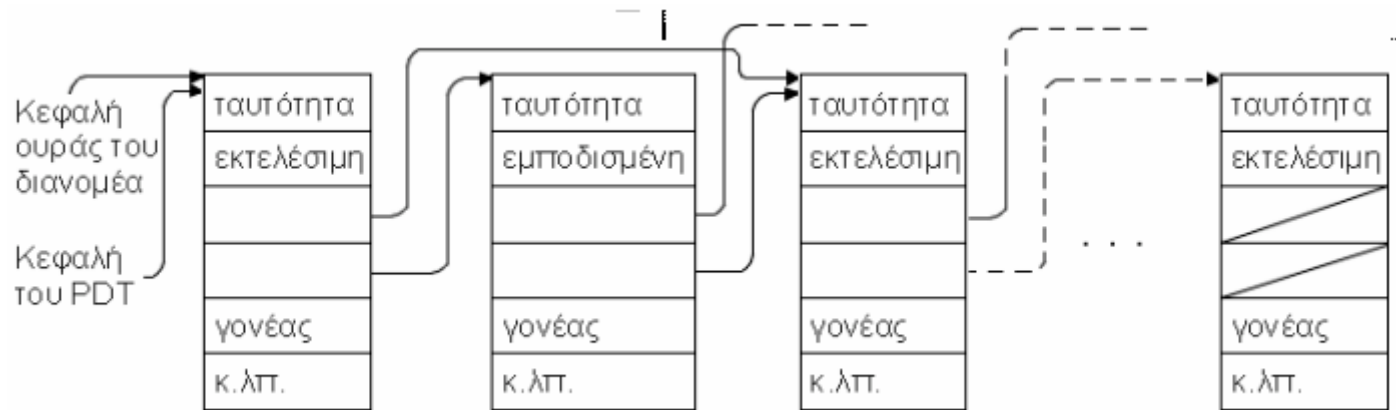
<b>Process management</b>	<b>Memory management</b>	<b>File management</b>
Registers Program counter Program status word Stack pointer Process state Priority Scheduling parameters Process ID Parent process Process group Signals Time when process started CPU time used Children's CPU time Time of next alarm	Pointer to text segment Pointer to data segment Pointer to stack segment	Root directory Working directory File descriptors User ID Group ID

# Πίνακας Περιγραφητών Διεργασιών (Process Descriptor Table, PDT)

- περιλαμβάνει τους περιγραφητές όλων των διεργασιών που βρίσκονται στο σύστημα
- οργανώνεται σε μία γραμμική λίστα μίας ή δύο κατευθύνσεων ή σαν ένας πίνακας
- περιλαμβάνει τις παρακάτω ουρές:
  - *ουρά της ΚΜΕ (processor queue)* ή *λίστα ετοιμών διεργασιών (ready list)* δηλαδή τις εκτελέσιμες διεργασίες (εκτελούμενες και έτοιμες)

# Πίνακας Περιγραφητών Διεργασιών

- Ένα υποσύνολο της λίστας αυτής ονομάζεται
  - *λίστα καταλλήλων (eligible list) ή λίστα του διανομέα (dispatcher list)*, δηλαδή τις διεργασίες που μπορεί να διανεμηθούν (εκτελεστούν) στο τρέχον χρονικό μεσοδιάστημα
- διάφορες *ουρές αναμονής* των εμποδισμένων και/ή αναστελλόμενων διεργασιών ανάλογα με το λόγο εμποδισμού ή αναστολής τους



# Ταυτότητες διεργασιών

- Η ταυτότητα/αναγνωριστικό (*process identifier, pid*) μιας διεργασίας πρέπει να είναι μοναδική
- Εφόσον οι διεργασίες δημιουργούνται και διαγράφονται δυναμικά, δεν αρκεί ο δείκτης της θέσης του περιγραφητή μιας διεργασίας για τον προσδιορισμό της
- Το σύστημα δίνει ένα νέο *αριθμό διεργασίας (process number)* στην κάθε νέα διεργασία που δημιουργείται
- Έτσι η ταυτότητα μιας διεργασίας μπορεί να είναι το ζεύγος (δείκτης περιγραφητή, αριθμός διεργασίας)

# Επικοινωνία διεργασιών με ανταλλαγή μηνυμάτων

- Για να επικοινωνήσουν δύο διεργασίες απαιτείται ένα *κανάλι επικοινωνίας (communication channel)* δηλαδή ένας *ενταμιευτής* ή μία *ουρά μηνυμάτων*
- Έτσι η χωρητικότητα ενός καναλιού επικοινωνίας μπορεί να οριστεί ως το πλήθος των μηνυμάτων που μπορούν να παραμείνουν σ' αυτό, ή ως το μέγεθος ενταμιευτή, ή ως το μήκος της ουράς των μηνυμάτων που συνδέεται με το κανάλι
- Διακρίνουμε δύο είδη επικοινωνίας: την άμεση (direct) και την έμμεση (indirect) επικοινωνία

## Άμεση επικοινωνία διεργασιών

- Μεταξύ κάθε ζεύγους διεργασιών που επικοινωνούν ορίζεται ένα κανάλι επικοινωνίας
- Κάθε κανάλι σχετίζεται με δύο διεργασίες μόνο
- Κάθε ζεύγος διεργασιών συνδέεται μ' ένα μόνο κανάλι
- Η επικοινωνία μεταξύ των διεργασιών αυτών μπορεί να είναι μίας κατεύθυνσης (unidirectional) ή δύο κατευθύνσεων (bidirectional)
- Το κάθε κανάλι ορίζεται αυτόματα από το σύστημα

# Άμεση επικοινωνία διεργασιών

## Συμμετρική (symmetric) αναφορά

- ο αποστολέας αναφέρεται “ονομαστικά” στον παραλήπτη  
`send_message (P, μήνυμα)`
- ο παραλήπτης αναφέρεται “ονομαστικά” στον αποστολέα  
`receive_message (Q, μήνυμα);`
- όπου P, Q αναγνωριστικά/ταυτότητες παραλήπτη/αποστολέα

## Ασύμμετρη (asymmetric) αναφορά

- ο αποστολέας αναφέρεται ονομαστικά στον παραλήπτη  
`send_message (P, μήνυμα);`
- ο παραλήπτης δέχεται μήνυμα από οποιοδήποτε αποστολέα  
`receive_message (pid, μήνυμα);`
- όπου P αναγνωριστικό/ταυτότητα παραλήπτη και
- pid παίρνει ως τιμή την ταυτότητα του αποστολέα

# Άμεση επικοινωνία διεργασιών

## Μειονεκτήματα άμεσης επικοινωνίας

- η αλλαγή της ταυτότητας μιας διεργασίας συνεπάγεται την αλλαγή όλων των αναφορών προς τη διεργασία αυτή
- είναι αδύνατη η ξεχωριστή μετάφραση των προγραμμάτων των διεργασιών και
- στα κατακευματισμένα συστήματα είναι άγνωστες ταυτότητες διεργασιών που εκτελούνται στους άλλους (απομακρυσμένους) υπολογιστές

## Έμμεση επικοινωνία διεργασιών με γραμματοκιβώ\τια (mailboxes) ή ενταμιευτές (ουρές) μηνυμάτων

- Τοποθετούνται και παραλαμβάνονται μηνύματα
- Χρειάζεται το αναγνωριστικό/ταυτότητα κάθε γραμματοκιβωτίου
- Μια διεργασία μπορεί να επικοινωνεί με μία άλλη διεργασία μέσω διαφορετικών γραμματοκιβωτίων
- Δύο διεργασίες μπορούν να επικοινωνήσουν μόνο αν μοιράζονται τουλάχιστον ένα γραμματοκιβώτιο
- `send_message (M, μήνυμα) ; receive_message (M, μήνυμα) ;`
- όπου M είναι το αναγνωριστικό/ταυτότητα ενός γραμματοκιβωτίου

# Έμμεση επικοινωνία διεργασιών

## Κανάλι επικοινωνίας

- Μεταξύ κάθε ζεύγους διεργασιών που επικοινωνούν ορίζεται ένα κανάλι επικοινωνίας μόνο αν αυτές μοιράζονται ένα γραμματοκιβώτιο
- ένα κανάλι επικοινωνίας μπορεί να σχετίζεται με περισσότερες από δύο διεργασίες
- Μεταξύ κάθε ζεύγους διεργασιών μπορεί να υπάρχουν διάφορα κανάλια (το καθένα αντιστοιχεί σ' ένα γραμματοκιβώτιο)
- Η επικοινωνία μεταξύ των διεργασιών αυτών μπορεί να είναι μίας κατεύθυνσης ή δύο κατευθύνσεων

# Έμμεση επικοινωνία διεργασιών

Όταν ο ιδιοκτήτης του γραμματοκιβωτίου είναι η διεργασία τότε

- αυτό είναι συνδεδεμένο με αυτή
- έχει οριστεί ως μέρος της
- μόνο η διεργασία μπορεί να παραλάβει μηνύματα από αυτό το γραμματοκιβώτιο
- οι υπόλοιπες διεργασίες μπορούν μόνο να στείλουν σ' αυτό το γραμματοκιβώτιο (εφόσον γνωρίζουν την ταυτότητά του)

## ΠΡΟΒΛΗΜΑΤΑ

- Όταν η διεργασία παύσει να υπάρχει, το γραμματοκιβώτιο παύει επίσης να υπάρχει
- Αν μια άλλη διεργασία στείλει μήνυμα σ' ένα γραμματοκιβώτιο που δεν υπάρχει πρέπει να ειδοποιηθεί γι' αυτό

# Έμμεση επικοινωνία διεργασιών

Όταν ο ιδιοκτήτης του γραμματοκιβωτίου είναι το Λ.Σ. (π.χ. UNIX)

- το γραμματοκιβώτιο είναι ανεξάρτητο από τις διεργασίες
- δεν είναι συνδεδεμένο με καμία συγκεκριμένη διεργασία
- Παρέχονται στις διεργασίες κλήσεις του συστήματος για να:
  - Συνδεθούν με ένα γραμματοκιβώτιο
  - Αποσυνδεθούν από ένα γραμματοκιβώτιο

# Έμμεση επικοινωνία διεργασιών

## ΠΡΟΒΛΗΜΑΤΑ

- Τερματισμός αποστολέα, προτού στείλει ένα μήνυμα που περιμένει κάποιος παραλήπτης (ο παραλήπτης θα παραμείνει για πάντα εμποδισμένος)
- Τερματισμός παραλήπτη, προτού παραλάβει τα μηνύματα που του έχει στείλει κάποιος αποστολέας (αν ο αποστολέας χρειάζεται απάντηση, τότε θα παραμείνει για πάντα εμποδισμένος)

## ΛΥΣΕΙΣ

- Τερματισμός (από το Λ.Σ.) του παραλήπτη/αποστολέα
- Ειδοποίηση (από το Λ.Σ.) του αποστολέα/παραλήπτη ότι ο παραλήπτης/αποστολέας έχει παύσει να υπάρχει

# Έμμεση επικοινωνία διεργασιών

## ΠΡΟΒΛΗΜΑΤΑ+

- Αποκομιδή απορριμμάτων (garbage collection), δηλαδή συλλογή και διαγραφή (από το Λ.Σ.) των μηνυμάτων που δεν έχουν καταναλωθεί
- Απώλεια μηνύματος/των
  - Συστήματα ενός επεξεργαστή (μεταβίβαση μηνυμάτων μέσω κοινής μνήμης)
    - Απώλεια μηνύματος/των συνεπάγεται την αποτυχία ολόκληρου του συστήματος
  - Κατανεμημένα συστήματα-δίκτυα (μεταβίβαση μηνυμάτων μέσω γραμμών επικοινωνίας)
    - Πρωτόκολλα ανοχής σφαλμάτων

# Διαχείριση χρονομέτρου

Ένα χρονόμετρο είναι ένα κύκλωμα που παράγει περιοδικές διακοπές/παλμούς και χρησιμοποιείται από το Λ.Σ. για:

- Να διατηρεί την ώρα (της ημέρας)
- Να απαγορεύει την χρήση της ΚΜΕ παραπάνω από ένα συγκεκριμένο τεμάχιο (κβάντο) χρόνου
- Να διατηρεί “ξυπνητήρια” (ALARMS) για τις διεργασίες που έχουν ζητήσει να αφυπνιστούν μια ορισμένη ώρα
- Να εξυπηρετεί την ιχνηλάτηση (tracing) της εκτέλεσης διεργασιών από προγράμματα παρακολούθησης (monitoring programs)
- Λογιστικούς σκοπούς
  - Μέτρηση των χρόνων χρήσης των πόρων
    - Χρέωση των διεργασιών
    - Συλλογή στατιστικών στοιχείων

# Διαχείριση χρονομέτρου

Από ειδικό διαχειριστή χρονομέτρου

- αύξηση του χρόνου επιβάρυνσης του συστήματος
- οι παραπάνω λειτουργίες εκτελούνται πολλές φορές ανά δευτερόλεπτο
- Από τον πυρήνα
  - διαδικασία αντιμετώπισης διακοπών χρονομέτρου
  - πολύ γρήγορη

# Διαχείριση χρονομέτρου

## Διατήρηση ώρας

- σε μια μεταβλητή τύπου `longint` (32 bits)
- αυξάνεται κατά 1 σε κάθε παλμό του χρονομέτρου
- ώρα εκφράζεται σε αριθμό διακοπών από 1/1/1970 (e.g. UNIX)
- **Πρόβλημα:** υπερχείλιση μεταβλητής

## Λύσεις

- ώρα εκφράζεται σε διακοπές από την τελευταία εκκίνηση (bootstrap) του συστήματος
- χρήση μεταβλητής μήκους 64 bits
- χρήση μεταβλητής μήκους 32 bits, αλλά η ώρα να εκφράζεται σε δευτερόλεπτα αντί διακοπές χρονομέτρου
  - $2^{32}$  δευτερόλεπτα > 136 χρόνια
  - ανάγκη χρήσης μιας δεύτερης μεταβλητής που
    - αυξάνεται κατά 1 σε κάθε διακοπή χρονομέτρου

# Διαχείριση χρονομέτρου

## (Κατα)μερισμός ΚΜΕ μεταξύ διεργασιών

- η εκτέλεση της τρέχουσας διεργασίας διακόπτεται όταν έχει παρέλθει ένα τεμάχιο (κβάντο) χρόνου
- μεταβλητή-μετρητής (counter)
- μετρητής παίρνει αρχική τιμή από το διανομέα
- προκαλείται διακοπή για κάθε παλμό του χρονομέτρου
- σε κάθε παλμό χρονομέτρου ο μετρητής ελαττώνεται κατά 1
- όταν πάρει την τιμή 0, τότε σημαίνεται μια άλλη διακοπή και
  - η τρέχουσα διεργασία αφαιρείται από την κεφαλή της ουράς του διανομέα και
  - εισάγεται σε άλλο σημείο της ουράς αυτής (προτεραιότητες)
  - ο διανομέας διανέμει μια άλλη εκτελέσιμη διεργασία

# Διαχείριση χρονομέτρου

## Ξυπνητήρια

- Το Λ.Σ. αφυπνίζει μια διεργασία όταν περάσει ένα ορισμένο χρονικό διάστημα
  - μεσοχρόνιος χρονοπρογραμματιστής
  - time-outs (εκπνοές χρόνου)
    - διαχειριστής δίσκου
    - αποστολέας μηνύματος
  - διεργασίες χρήστη
- Μια μεταβλητή στο περιβάλλον κάθε διεργασίας που ελαττώνεται κατά 1 σε κάθε διακοπή του χρονομέτρου

# Διαχείριση χρονομέτρου

## Ξυπνητήρια

- Αναζήτησή από το Λ.Σ. της τιμής της στους περιγραφητές όλων των διεργασιών θα προκαλούσε πολύ μεγάλη χρονική επιβάρυνση. Έτσι χρησιμοποιείται συνήθως μια
- **Λίστα αφύπνισης**
  - Ταξινομημένη σε αύξουσα σειρά χρόνων αφύπνισης
  - Περιεχόμενα κάθε κόμβου της λίστας:
    - διαφορά του χρόνου αφύπνισης (σε παλμούς χρονομέτρου) από το χρόνο του προηγούμενου κόμβου
    - ταυτότητα της διεργασίας
  - Χρησιμοποιείται μια μεταβλητή-μετρητής που παίρνει ως τιμή το χρόνο αφύπνισης της διεργασίας που βρίσκεται στην κεφαλή της λίστας

# Διαχείριση χρονομέτρου

## Ξυπνητήρια

- Ο μετρητής ελαττώνεται κατά 1 σε κάθε παλμό του χρονομέτρου
- Όταν πάρει την τιμή 0 τότε
  - Η κεφαλή της λίστας και όλες οι διεργασίες που έχουν ίσους χρόνους με αυτήν αφυπνίζονται και αφαιρούνται από τη λίστα
  - Ο μετρητής παίρνει ως τιμή το χρόνο αφύπνισης της διεργασίας που βρίσκεται στη νέα κεφαλή της λίστας
  - κ.ο.κ.

# Διαχείριση χρονομέτρου

## Ιχνηλάτηση - παρακολούθηση εκτέλεσης διεργασιών

- Εντοπισμός σημείων της διεργασίας (χρήστη ή συστήματος) που ξοδεύουν περισσότερο χρόνο εκτέλεσης
- Ιστόγραμμα των τιμών που παίρνει ο μετρητής του προγράμματος της διεργασίας
- Ειδικό πεδίο στο περιβάλλον της διεργασίας που προσδιορίζει αν η διεργασία παρακολουθείται
- Σε κάθε παλμό του χρονομέτρου το σύστημα αποθηκεύει την τιμή του μετρητή του προγράμματος της διεργασίας (μείον το μήκος της τελευταίας εντολής που εκτελέστηκε)

# Διαχείριση χρονομέτρου

## Λογιστική χρέωση

- Χρόνος εκτέλεσης διεργασίας διατηρείται σ' ένα πεδίο/μεταβλητή στο περιβάλλον της
- Σε κάθε διακοπή του χρονομέτρου αυξάνεται η τιμή της κατά 1
- Έτσι η διεργασία χρεώνεται για ΟΛΕΣ τις διακοπές που εξυπηρετούνται κατά την εκτέλεσή της (όσο αυτή είναι τρέχουσα)
- Άδικη χρέωση:
  - ο χρόνος εξυπηρέτησης διακοπών πρέπει να χρεώνεται στις διεργασίες που αφορούν στις διακοπές αυτές
    - πολλές διακοπές δεν αφορούν στις διεργασίες χρηστών αλλά στις διεργασίες του συστήματος

## Ενεργοποίηση/αφύπνιση διεργασιών

- **Αφύπνιση με χρήση σηματοφορέων** (π.χ. μεσοχρόνιος χρονοπρογραμματιστής)
  - Σε κάθε διεργασία αντιστοιχείται ένας ιδιωτικός (private) σηματοφορέας "ενεργοποίηση" (activation)
  - Η διεργασία μπορεί να εκτελέσει την εντολή `wait(activation)`
  - Ο πυρήνας ή ακριβέστερα μια διαδικασία εξυπηρέτησης μιας παγίδας ή μιας διακοπής μπορεί να εκτελέσει την εντολή `signal(activation)`
- **ΑΛΛΑ** η διεργασία μπορεί να αφυπνίζεται είτε όπως παραπάνω είτε λαμβάνοντας ένα μήνυμα από κάποιον αποστολέα. Έτσι χρειάζεται μια γενικότερη της `receive_message(p, m)` διαδικασία, η `receive_event(p, m)`

# Ενεργοποίηση/αφύπνιση διεργασιών

## Όταν μια διεργασία καλεί τη `receive_event`

- Αν ο σηματοφορέας `activation` έχει την τιμή 1, τότε η διεργασία συνεχίζει την εκτέλεση της
- Ειδάλλως (αν ο σηματοφορέας `activation` έχει την τιμή 0, τότε η διεργασία καλεί τη `receive_message`

Αν δεν υπάρχει μήνυμα για να παραλάβει η διεργασία τότε *εμποδίζεται περιμένοντας για γεγονός* (*waiting for event*), δηλαδή

- είτε για ενεργοποίηση είτε για μήνυμα

## Πλεονέκτημα

Ομοιογενείς μηχανισμοί επικοινωνίας μεταξύ

- διεργασιών μεταξύ τους
- διεργασιών και διαδικασιών εξυπηρέτησης διακοπών και παγίδων

## **Επιστροφή από τις διαδικασίες εξυπηρέτησης διακοπών και παγίδων**

- Μια παγίδα ή μια διακοπή οδηγεί στην εκτέλεση μιας διαδικασίας εξυπηρέτησής της
- Η εκτέλεση μιας τέτοιας διαδικασίας μπορεί να αφυπνίσει μια διεργασία με μεγαλύτερη προτεραιότητα από την εκτελούμενη
- Η ΚΜΕ πρέπει να δοθεί στη διεργασία που έχει την μεγαλύτερη προτεραιότητα
- Ο διανομέας είναι το απόσπασμα του προγράμματος που καταχωρεί την ΚΜΕ

ΑΡΑ οι διαδικασίες των διακοπών και των παγίδων πρέπει να επιστρέφουν στο διανομέα

# Επιστροφή από τις διαδικασίες εξυπηρέτησης διακοπών και παγίδων

Σε πολλά συστήματα αυτό γίνεται αυτόματα από το υλισμικό με χρήση ενός ειδικού δυφίου που ονομάζεται σημαία **παγίδας κατάστασης χρήστη (user mode trap)**

- μια διαδικασία εξυπηρέτησης παγίδας ή διακοπής που αφυπνίζει μια διεργασία φορτώνει τη σημαία της παγίδας κατάστασης του χρήστη με την τιμή `true (1)`
- η διαδικασία εξυπηρέτησης επιχειρεί να επιστρέψει στην κατάσταση του χρήστη εκτελώντας μια εντολή `eret`
- αν η σημαία παγίδα κατάστασης χρήστη είναι ίση με 1, τότε δημιουργείται μια νέα παγίδα που μεταφέρει τον έλεγχο στο διανομέα

## Υλοποίηση σηματοφορέων

- Οι υπολογιστές δε διαθέτουν εντολές που να επεξεργάζονται γενικούς σηματοφορείς
- Οι γενικοί σηματοφορείς μπορούν ν' αντικατασταθούν από δυαδικούς σηματοφορείς και μη αρνητικούς μετρητές, αλλά
- Οι περισσότεροι υπολογιστές δεν διαθέτουν ούτε εντολές που να επεξεργάζονται δυαδικούς σηματοφορείς

Έτσι, σε συστήματα ενός επεξεργαστή η υλοποίηση των `wait(S)` και `signal(S)` επιτυγχάνεται με απαγόρευση των διακοπών

# Υλοποίηση σηματοφορέων

**wait(S)**

**di**

lw \$t1,S

bne \$t1,\$0,L

κατάσταση τρέχουσας διεργασίας γίνεται εμποδισμένη;

αφαίρεσε τον περιγραφητή της διεργασίας από την ουρά του διανομέα;

πρόσθεσέ τον στο σύνολο (ή στο τέλος της ουράς) των

διεργασιών που είναι εμποδισμένες πάνω στον S;

L: sw \$0,S

**eret**

# Υλοποίηση σηματοφορέων

**signal(S)**

**di**

if (υπάρχουν εμποδισμένες διεργασίες στο σύνολο (ουρά)

του S)

{ αφαίρεσε τον περιγραφητή μιας διεργασίας (της  
διεργασίας που βρίσκεται στην κεφαλή της ουράς του S)  
από το σύνολο (την ουρά) των εμποδισμένων πάνω στον  
S;

κατάσταση διεργασίας = έτοιμη;

πρόσθεσέ τον περιγραφητή της διεργασίας στην ουρά του  
διανομέα;

}

else

{ li \$t1,1

sw \$t1,S

}

**eret**

# Υλοποίηση σηματοφορέων

**Οι παραπάνω υλοποιήσεις πρέπει είναι όσο το δυνατό ταχύτερες**

- χρόνος εκτέλεσης διαδικασιών υλοποίησης  
    < χρόνου κρίσης των περιφερειακών συσκευών
- διακοπές της ΚΜΕ (π.χ. διακοπή ρεύματος, σφάλματα ισοτιμίας μνήμης) πρέπει να επιτρέπονται
  - αν συμβούν δεν εξασφαλίζεται ο αμοιβαίος αποκλεισμός
- ο αμοιβαίος αποκλεισμός δεν μπορεί να εξασφαλιστεί με άπειρες ανακυκλώσεις, γιατί τότε η (μία) ΚΜΕ θα δεσμευόταν για πάντα

# Διανομέας (Dispatcher)

## Χρονοπρογραμματισμός ΚΜΕ

### Διανομέας

- βραχυχρόνιος χρονοπρογραμματιστής της ΚΜΕ
- εκτελείται πολύ συχνά (Oms), γι αυτό πρέπει να είναι
- απλούστατος (πολύ γρήγορος)
- παίρνει την πρώτη διεργασία από την ουρά του και την διανέμει στην ΚΜΕ

## Διαχειριστής διεργασιών (βλ. Κεφ. 9)

- μεσοχρόνιος χρονοπρογραμματιστής της ΚΜΕ
- εκτελείται πολύ αραιότερα (Osec)
- προετοιμάζει την ουρά του διανομέα
- καθορίζει τις προτεραιότητες των διεργασιών
- ταξινομεί τους περιγραφητές των διεργασιών ανάλογα με την προτεραιότητά τους στη *λίστα ετοιμών διεργασιών*
- καθορίζει το *βαθμό του πολυπρογραμματισμού*, δηλαδή πόσες από αυτές τις διεργασίες στη λίστα των ετοιμών μπορεί να διανεμηθούν μέχρι την επόμενη ενεργοποίησή του (μεσοδιάστημα χρόνου)
- οι διεργασίες αυτές αποτελούν την *ουρά του διανομέα* ή το *σύνολο καταλλήλων (eligible set) διεργασιών*, δηλαδή το σύνολο των διεργασιών που μπορούν να διανεμηθούν στο επόμενο μεσοδιάστημα

# Διανομέας

## Κριτήρια σχεδιασμού

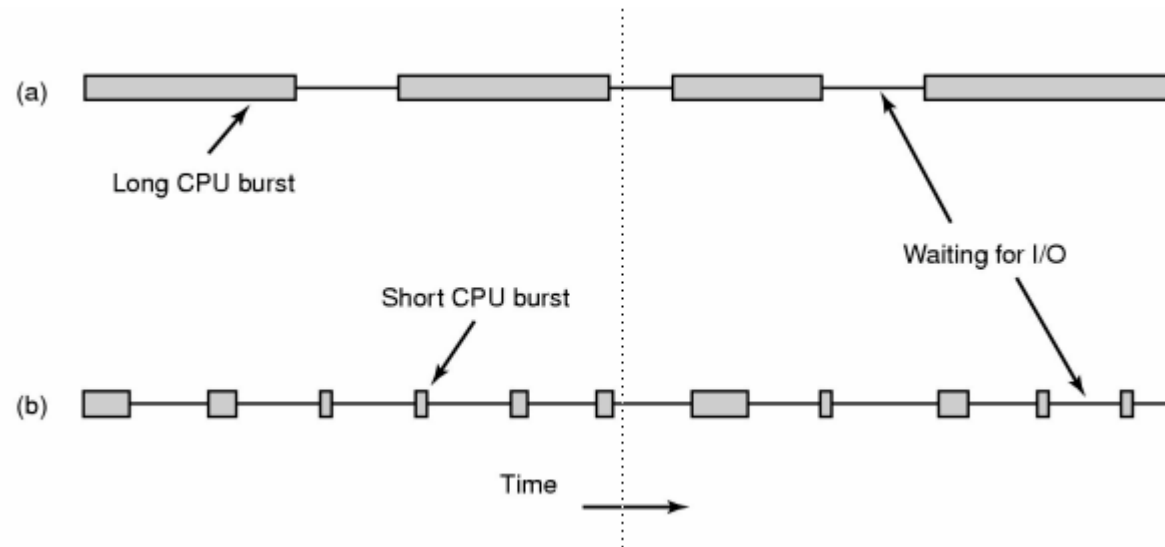
- ελαχιστοποίηση χρόνου επιβάρυνσης που απαιτείται για τη διανομή των διεργασιών (μεταγωγή των συμφραζομένων τους)
- γρήγορη αντίδραση στις μεταβολές κατάστασης (εκτέλεση της πιο κρίσιμης διεργασίας)
- *αντικρουόμενα κριτήρια*

## Κατηγορίες διεργασιών

- *περιορισμένες από την ΚΜΕ (CPU bound processes) (a)*
  - απαιτήσεις ΚΜΕ  $\gg$  απαιτήσεις E/E
- *περιορισμένες από E/E (I/O bound processes) (b)*
  - απαιτήσεις E/E  $\gg$  απαιτήσεις ΚΜΕ

# Διανομέας

## Κατηγορίες διεργασιών



# Διανομέας

## ΔΥΟ ΑΚΡΑΙΟΙ ΑΛΓΟΡΙΘΜΟΙ

### Μη διακοπτόμενη διανομή (non-preemptive dispatching)

- Μια διεργασία χρησιμοποιεί την ΚΜΕ μέχρις ότου εμποδιστεί
- Αν ελευθερωθεί μια διεργασία με μεγαλύτερη προτεραιότητα τότε αυτή περιμένει μέχρι να απελευθερωθεί η ΚΜΕ
- *Μικρός χρόνος επιβάρυνσης* (μεταγωγή συμφραζομένων)
- *Αργή αντίδραση* στις αλλαγές κατάστασης διεργασιών
- Καλή ρυθμαπόδοση για τις περιορισμένες από την ΚΜΕ διεργασίες
- Καλή επιλογή μόνο αν οι διεργασίες ελευθερώνουν πολύ γρήγορα την ΚΜΕ (π.χ. διεργασίες συστήματος)

# Διανομέας

## Διακοπτόμενη διανομή (preemptive dispatching)

- Μια διεργασία χρησιμοποιεί την ΚΜΕ, μέχρις ότου να ελευθερωθεί μία πιο κρίσιμη διεργασία
- Αν ελευθερωθεί μια διεργασία με μεγαλύτερη προτεραιότητα αυτή αποκτά αμέσως την ΚΜΕ
- *Μεγαλύτερος χρόνος επιβάρυνσης*
- *Γρήγορη απόκριση* στις διεργασίες που έχουν μεγάλη προτεραιότητα
- Πιθανότητα μονοπώλησης της ΚΜΕ από περιορισμένες από την ΚΜΕ διεργασίες με μεγάλη προτεραιότητα
- η αντικατάσταση μιας διεργασίας από μια άλλη της ίδιας περίπου προτεραιότητας ίσως δεν αξίζει τον κόπο

# Διανομέας

ΣΥΜΒΙΒΑΣΜΟΣ ανάμεσα στους δύο προηγούμενους αλγόριθμους

## Τεμαχισμός χρόνου (time-slicing)

- Μια διεργασία χρησιμοποιεί την ΚΜΕ για ένα ορισμένο *τεμάχιο χρόνου (time slice)*, γνωστό και ως *κβάντο χρόνου (quantum time)*
- *τεμάχιο χρόνου είναι το μέγιστο χρονικό διάστημα που μια διεργασία μπορεί να χρησιμοποιήσει την ΚΜΕ*
- Διανέμονται μόνο οι κατάλληλες διεργασίες
- βαθμός πολυπρογραμματισμού
- αποφυγή αλωνίσματος μνήμης

# Διανομέας

## Μετά την διανομή επιτρέπονται οι διακοπές

- Η ουρά του διανομέα μπορεί να είναι οργανωμένη σε διαφορετικές ουρές από το διαχειριστή ΚΜΕ ανάλογα με
  - προτεραιότητα
  - απαιτήσεις πόρων
  - χρόνο αναμονής
- Αυτό δεν επηρεάζει τον παραπάνω αλγόριθμο του διανομέα

# Διανομέας

## Αλγόριθμος διανομέα (επιστροφή μετά από διακοπή η παγίδα)

0. Αν δεν υπάρχει κατάλληλη διεργασία (όλες οι διεργασίες ή είναι εμποδισμένες ή έχουν ανασταλεί ή και τα δύο), τότε
  - Επίτρεψε τις διακοπές
  - Περίμενε  
(δηλαδή εκτέλεσε την εντολή `while (true) ;`  
ή εκτέλεσε μια μηδενική ή μια διεργασία αναμονής, που κάνει αυτή τη δουλειά)

# Διανομέας

## Αλγόριθμος διανομέα (επιστροφή μετά από διακοπή η παγίδα)

1. Αν η εκτελέσιμη διεργασία με τη μεγαλύτερη προτεραιότητα είναι η τρέχουσα (αυτή δηλαδή που διακόπηκε ή που δημιούργησε την παγίδα), τότε
2. Επανάλαβε την εκτέλεσή της
  - Επίστρεψε τον έλεγχο στην τιμή της λέξης κατάστασης του προγράμματος που είχε αποθηκευθεί
  - Επίτρεψε τις διακοπές

## Διανομέας

3. Αν η εκτελέσιμη διεργασία με τη μεγαλύτερη προτεραιότητα δεν είναι η τρέχουσα, τότε
  - 3.1. Εκτέλεσε τη μεταγωγή των συμφραζομένων
    - αποταμίευσε το περιβάλλον της τρέχουσας διεργασίας
    - επανάφερε το περιβάλλον της πιο κρίσιμης διεργασίας
  - 3.2. Κάνε την πιο κρίσιμη διεργασία τρέχουσα και επανάλαβε την εκτέλεσή της
    - Επίστρεψε τον έλεγχο στην τιμή της λέξης κατάστασης προγράμματος που επαναφέρθηκε
    - Επίτρεψε διακοπές

# Διεργασίες στο σύστημα UNIX

## Ιεραρχική οργάνωση δένδρου

- διεργασία `init`
  - ρίζα του δένδρου
  - όλες οι διεργασίες χρηστών είναι απόγονοί της
  - βρίσκεται στο ευρετήριο `/etc` κι έχει `pid=1`
  - διαβάζει το αρχείο `/etc/tty` (terminals)
  - δημιουργεί διεργασίες `getty` για κάθε terminal
  - αποκοιμιέται

# Διεργασίες στο σύστημα UNIX

## διεργασία `getty`

- περιμένει να διαβάσει το όνομα ενός χρήστη
- δημιουργεί μια διεργασία `login`
- μεταβιβάζει το όνομα του χρήστη στη διεργασία `login`

## διεργασία `login`

- περιμένει να διαβάσει το σύνθημα χρήστη
- το κρυπτογραφεί (encrypts it)
- το συγκρίνει με την αντίστοιχη κρυπτογραφημένη συμβολοσειρά του αρχείου `/etc/passwd`
- αν η σύγκριση πετύχει, δημιουργεί μια διεργασία **φλοιό** (π.χ. `ssh`)

# Διεργασίες στο σύστημα UNIX

## διεργασία login

- δίνει στο φλοιό αυτό:
  - το αναγνωριστικό uid του χρήστη (αρχείο /etc/passwd)
  - το αναγνωριστικό της πρώτης ομάδας gid του χρήστη (αρχείο /etc/group)

## διεργασία φλοιός

- διερμηνευτής εντολών
- η ρίζα όλων των διεργασιών του χρήστη

# Διεργασίες στο σύστημα UNIX

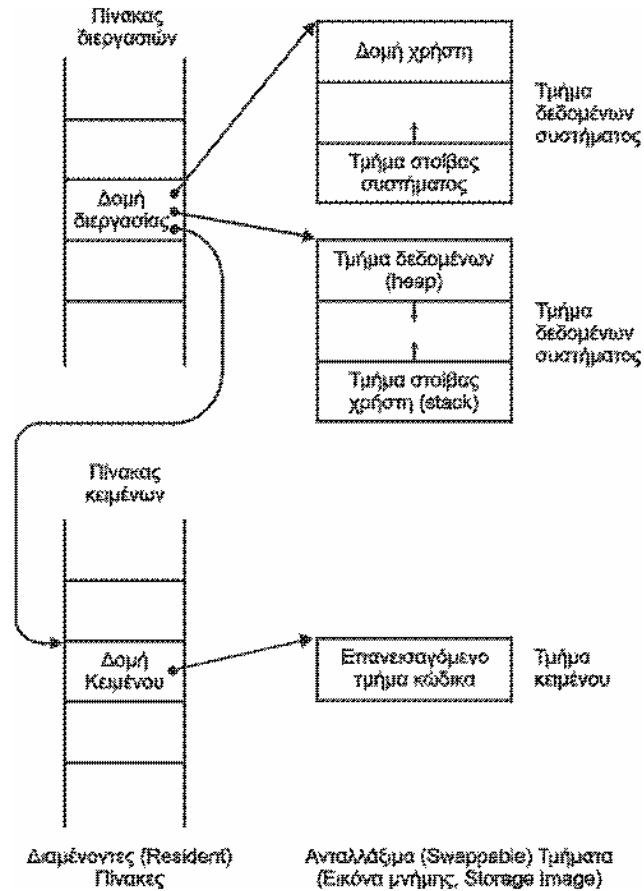
## Πίνακας περιγραφητών Διεργασιών (Process Table)

- σταθερού μεγέθους
- ορίζεται κατά τη σύνδεση (linking) του συστήματος
- η λίστα ετοιμών είναι μία γραμμική λίστα δύο διευθύνσεων
- και διαμένει μόνιμα στην κύρια μνήμη

ο περιγραφητής διεργασίας ονομάζεται **δομή διεργασίας (process structure)** και περιέχει:

- δείκτες προς τα τμήματα της διεργασίας
  - Τμήμα κώδικα = Τμήμα κειμένου (text segment)
  - Τμήμα δεδομένων χρήστη
  - Περιβάλλον = Τμήμα δεδομένων συστήματος

# Εικόνα διεργασίας (process image) στο UNIX



# Διεργασίες στο σύστημα UNIX

## Τμήμα κειμένου

- μπορεί μόνο να εκτελεστεί (execute only)

## Πίνακας περιγραφητών τμημάτων **Κειμένων (Text Table)**

- σταθερού μήκους
- ορίζεται κατά τη σύνδεση του συστήματος
- διαμένει μόνιμα στην κύρια μνήμη

# Διεργασίες στο σύστημα UNIX

Περιγραφητής (επανεισαγόμενου) τμήματος κειμένου διεργασίας ονομάζεται **Δομή Κειμένου (Text Structure)** και περιέχει:

- διεύθυνση του πίνακα σελίδων του τμήματος στη βοηθητική μνήμη
- διεύθυνση του τμήματος στην κύρια μνήμη (αν είναι φορτωμένο)
- πλήθος των διεργασιών που μοιράζονται το τμήμα
  - αν πλήθος = 0, τότε η μνήμη και το αντίστοιχο στοιχείο του πίνακα των τμημάτων ελευθερώνονται
- δείκτη της κεφαλής της λίστας των διεργασιών που μοιράζονται το τμήμα αυτό

# Διεργασίες στο σύστημα UNIX

## Τμήμα δεδομένων χρήστη

- ανταλλάξιμο τμήμα
- μπορεί να διαβαστεί και να μετατραπεί (read/write)
  - Τμήμα σωρού
    - αυξάνεται προς τα “κάτω”
    - αποθηκεύει τα δεδομένα του σωρού
  - Τμήμα στοίβας
    - αρχίζει στην τελευταία διεύθυνση του εικονικού χώρου διευθύνσεων
    - αυξάνεται προς τα “πάνω”
    - αποθηκεύει τα δεδομένα της στοίβας

# Διεργασίες στο σύστημα UNIX

## Τμήμα δεδομένων του συστήματος (Περιβάλλον)

- ανταλλάξιμο τμήμα
- προστατευόμενο τμήμα
  - η διεργασία δεν μπορεί ν' αναφερθεί σ' αυτό όταν εκτελείται στην κατάσταση χρήστη

**δομή χρήστη (user structure) = περιβάλλον** και περιέχει:

- περιεχόμενα των καταχωρητών
- περιγραφητές των ανοικτών αρχείων
- δεδομένα λογιστικής φύσεως κ.λπ.

## στοίβα για τη φάση συστήματος διεργασίας

- χρησιμοποιείται από μια διεργασία του συστήματος που κλήθηκε από τη διεργασία χρήστη