

ΜΥΥ601 Λειτουργικά Συστήματα

Εαρινό 2024

Μάθημα 2

Διεργασίες

Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Ιωαννίνων

1

Περίγραμμα

- Εισαγωγή
- Καταστάσεις Διεργασίας
- Περιγραφή Διεργασίας
- Έλεγχος Διεργασίας
- Unix
- Κλήσεις Συστήματος

2

Εισαγωγή

- **Διεργασία**
 - Πρόγραμμα που εκτελείται
- **Απαιτήσεις λειτουργικού συστήματος**
 - Δημιουργία διεργασιών
 - Εκχώρηση πόρων
 - Εναλλαγή εκτέλεσης διεργασιών
 - Υποστήριξη επικοινωνίας μεταξύ διεργασιών

Εαρινό 2024

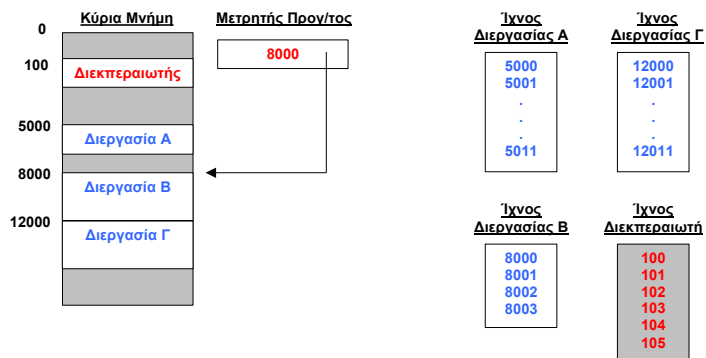
©Σ. Β. Αναστασιάδης

3

3

Παράδειγμα

- **Ίχνος (Trace)**
 - Ακολουθία εντολών που εκτελούνται από μια διεργασία



Εαρινό 2024

©Σ. Β. Αναστασιάδης

4

4

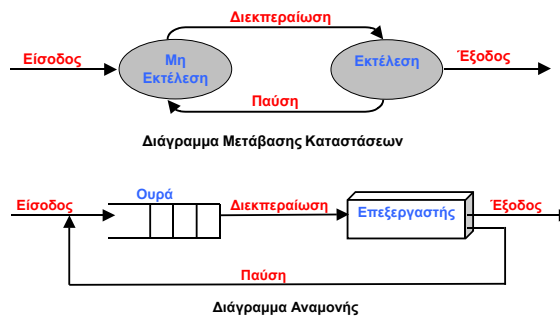
Εκτέλεση

- **Θεωρούμε**
 - Όλα τα προγράμματα στην κύρια μνήμη
 - Εναλλαγή διεργασιών λόγω λήξης χρόνου ή αίτησης Εισόδου/Εξόδου

Διεργασία A	Εναλλαγή από A σε B	Εναλλαγή από B σε Γ	Εναλλαγή από Γ σε A
01 5000	07 100	17 100	27 100
02 5001	08 101	18 101	28 101
03 5002	09 102	19 102	29 102
04 5003	10 103	20 103	30 103
05 5004	11 104	21 104	31 104
06 5005	12 105	22 105	32 105
--Λήξη Χρόνου--	13 8000	23 12000	33 5006
	14 8001	24 12001	33 5007
	15 8002	25 12002	34 5008
	16 8003	26 12003	35 5009
	--Αίτηση Ε/Ε--	--Λήξη Χρόνου--	...

Μοντέλο Διεργασίας Δύο Καταστάσεων

- **Μη εκτέλεση**
 - Διεργασία καινούρια ή σταματημένη από τον διεκπεραιωτή
- **Εκτέλεση**
 - Διεργασία που εκτελείται



Δημιουργία και Τερματισμός Διεργασίας

- **Δημιουργία**
 - Νέα εργασία δέσμης
 - Εισαγωγή διαδραστικού χρήστη
 - Υπηρεσία λειτουργικού συστήματος (π.χ. εκτύπωση κειμένου)
 - Έναρξη από πρόγραμμα χρήστη (π.χ. παράλληλη εφαρμογή)
- **Τερματισμός**
 - Κανονική ολοκλήρωση
 - Υπέρβαση μεγέθους μνήμης, παραβίαση ορίου μνήμης
 - Μη υπάρχουσα εντολή, προνομιστική εντολή
 - Τερματισμός γονέα, αίτηση τερματισμού απογόνου
 - Αίτηση υπεύθυνου συστήματος, αίτηση λειτουργικού συστήματος
 - Υπέρβαση χρονικού ορίου, χρόνου αναμονής για κάποιο γεγονός
 - Σφάλμα Ε/Ε, σφάλμα προστατευμένου πόρου, λάθος χρήση δεδομένων
 - Αριθμητικό σφάλμα

Εαρινό 2024

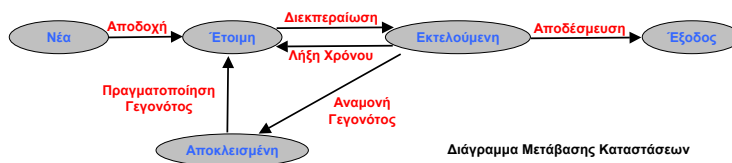
©Σ. Β. Αναστασιάδης

7

7

Μοντέλο Πέντε Καταστάσεων

- **Νέα:** Μόλις δημιουργήθηκε αλλά όχι ακόμη στη μνήμη
 - **Έτοιμη:** Προετοιμασμένη για εκτέλεση όταν της δοθεί ευκαιρία
 - **Εκτελούμενη:** Εκτελείται αυτή τη στιγμή
 - **Αποκλεισμένη:** Αδύνατο να εκτελεστεί μέχρι να συμβεί κάποιο γεγονός π.χ. Ε/Ε
 - **Έξοδος:** Δε μπορεί να εκτελεστεί άλλο π.χ. σταματημένη ή σε απόρριψη
-
- **Εκτελούμενη→Έτοιμη:** Λήξη χρόνου, εκτόπιση λόγω προτεραιότητας, οικειοθελώς
 - **Εκτελούμενη→Αποκλεισμένη:** Κλήση εξυπηρέτησης συστήματος π.χ. Ε/Ε
 - **Έτοιμη→Έξοδος, Αποκλεισμένη→Έξοδος:** Δεν εμφανίζονται στο παρακάτω σχήμα για λόγους ευκρίνειας



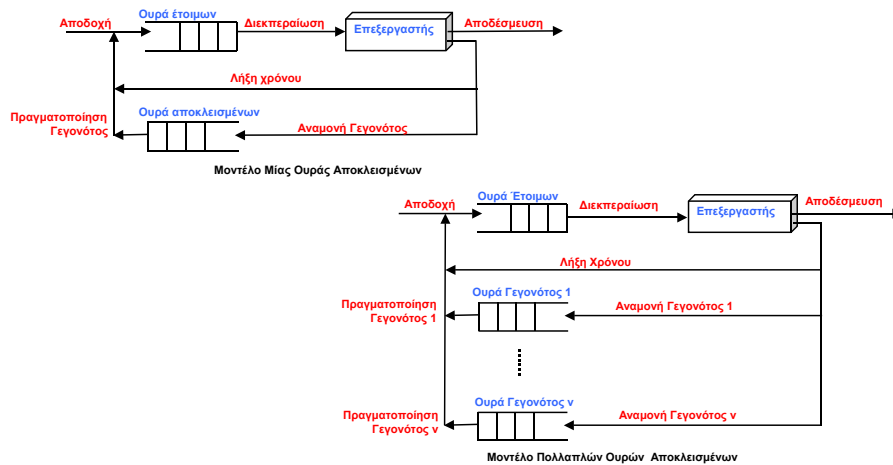
Εαρινό 2024

©Σ. Β. Αναστασιάδης

8

8

Μοντέλα Αναμονής Πέντε Καταστάσεων



Εαρινό 2024

©Σ. Β. Αναστασιάδης

9

9

Ανάγκη για Αναστολή Διεργασίας

- **Επεξεργαστής πολύ πιο γρήγορος από τις συσκευές E/E**
 - Η κύρια μνήμη διατηρεί πολλαπλές διεργασίες
 - Μπορεί όλες οι διεργασίες αποκλεισμένες εν αναμονή E/E
 - Ανάγκη για υψηλότερο βαθμό πολυπρογραμματισμού
- **Λύση 1**
 - Προσθέτουμε μνήμη για να χωρέσουν πιο πολλές διεργασίες
 - Σημαντικό κόστος μνήμης
 - Συνήθως οδηγεί σε μεγαλύτερες διεργασίες όχι περισσότερες
- **Λύση 2**
 - *Αναστολή διεργασίας* μεταφορά μέρους της διεργασίας στο δίσκο

Δύο επιπλέον καταστάσεις

1. *Αποκλεισμένη/Αναστολή* σε αναστολή εν αναμονή E/E
2. *Έτοιμη/Αναστολή* σε αναστολή έτοιμη για εκτέλεση

Εαρινό 2024

©Σ. Β. Αναστασιάδης

10

10



11

11

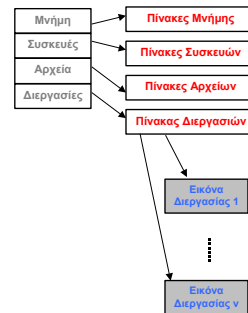
- **Εναλλαγή**
 - Αποδέσμευση κύριας μνήμης για την είσοδο έτοιμης διεργασίας
- **Αίτηση συστήματος**
 - Αναστολή εκτέλεσης προγράμματος (π.χ. μετανάστευση διεργασίας)
- **Αίτηση χρήστη**
 - Αναστολή εκτέλεσης προγράμματος (π.χ. εκσφαλμάτωση ή έλεγχος χρήσης πόρων)
- **Χρονισμός**
 - Περιοδική εκτέλεση διεργασίας συστήματος (π.χ. συντήρηση)
- **Αίτηση γονέα**
 - Εξέταση, αλλαγή ή συντονισμός απογόνων (π.χ. σφάλμα αρχείου)

12

12

Δομές Ελέγχου ΛΣ

- **Πίνακες μνήμης**
 - Καταχώρηση κύριας μνήμης σε διεργασίες
 - Καταχώρηση δευτερεύουσας μνήμης σε διεργασίες
 - Χαρακτηριστικά προστασίας μνήμης
 - Άλλες πληροφορίες διαχείρισης
- **Πίνακες Εισόδου/Εξόδου**
 - Διαχείριση λειτουργιών καναλιών και συσκευών Ε/Ε
- **Πίνακες αρχείων**
 - Ύπαρξη, θέση, κατάσταση, χαρακτηριστικά αρχείων
- **Πίνακες διεργασιών**
 - Αναφορές σε πίνακες μνήμης, Ε/Ε, αρχείων
 - Δεδομένα ρύθμισης συστήματος που καθορίζουν το περιβάλλον



Εικόνα Διεργασίας

- **Πρόγραμμα χρήστη**
 - Εκτελέσιμος κώδικας
- **Δεδομένα χρήστη**
 - Μεταβλητές προγράμματος
 - Χώρος στοίβας χρήστη
 - Μεταβλητά προγράμματα
- **Στοιβα**
 - Παράμετροι και διευθύνσεις κλήσης για διαδικασίες και κλήσεις συστήματος
- **Δομή ελέγχου διεργασίας**
 - Χαρακτηριστικά ελέγχου διεργασίας
 - Ταυτοποίηση διεργασίας
 - Πληροφορίες κατάστασης επεξεργαστή
 - Πληροφορίες ελέγχου διεργασίας



Δομή Ελέγχου Διεργασίας

- **Ορισμός**
 - Συλλογή από χαρακτηριστικά της διεργασίας που χρησιμοποιούνται από το ΛΣ
 - Η πιο σημαντική δομή δεδομένων του ΛΣ
- **Ταυτοποίηση διεργασίας**
 - Ταυτότητα της διεργασίας, της γονικής διεργασίας, του χρήστη
- **Πληροφορίες κατάστασης επεξεργαστή**
 - Καταχωρητές ορατοί στους χρήστες
 - Καταχωρητές ελέγχου και κατάστασης (π.χ. μετρητές προγ/τος, κωδικοί ελέγχου), δείκτες στοίβας
- **Πληροφορίες ελέγχου διεργασίας**
 - Πληροφορίες δρομολόγησης και κατάστασης (π.χ. προτεραιότητα, γεγονότα)
 - Συνδέσεις σε άλλες διεργασίες
 - Διαδιεργασιακή επικοινωνία (π.χ. σήματα)
 - Προνόμια μνήμης και εντολών, διαχείριση μνήμης
 - Κατοχή και χρήση πόρων (π.χ. αρχεία, συνδέσεις δικτύου)

Process Control Block

Identifier
State
Priority
Program counter
Memory pointers
Context data
I/O status info
Accounting info
...

Τρόποι Εκτέλεσης

- **Τουλάχιστο δύο τρόποι εκτέλεσης**
 - Προστασία του λειτουργικού συστήματος από τα προγράμματα χρήστη
- **Επίπεδο Πυρήνα**
 - Πλήρης έλεγχος επεξεργαστή, εντολών, καταχωρητών, μνήμης
 - Διαχείριση διεργασίας π.χ. δημιουργία, τερματισμός, δρομολόγηση, επικ/νία
 - Διαχείριση μνήμης π.χ. καταχώρηση χώρου διευθύνσεων, εναλλαγή
 - Διαχείριση E/E π.χ. διαχείριση ενδιάμεσης μνήμης, συσκευών, καναλιών
 - Συναρτήσεις υποστήριξης π.χ. χειρισμός διακοπών, λογιστική, παρατήρηση
- **Επίπεδο χρήστη**
 - Βασικά χρησιμοποιείται για την εκτέλεση προγραμμάτων χρήστη
- **Το επίπεδο καθορίζεται από δυαδικά ψηφία κατάστασης του επεξεργαστή**
 - Π.χ. current privilege level (CPL) στον επεξεργαστή Intel Pentium
 - Αλλάζουν μόνον από το ΛΣ εφόσον επιτρέπεται

Δημιουργία Διεργασίας

1. **Ανάθεση μοναδικής ταυτότητας στη νέα διεργασία**
 - Πρόσθεση νέου στοιχείου στον κύριο πίνακα διεργασιών
2. **Εκχώρηση χώρου για τη διεργασία**
 - Όλα τα στοιχεία της εικόνας της διεργασίας
 - Ποσότητα μνήμης προκαθορισμένη ή από τη γονική διεργασία
3. **Αρχικοποίηση της δομής ελέγχου διεργασίας**
 - Ταυτοποίηση διεργασίας
 - Πληροφορίες κατάστασης επεξεργαστή π.χ. μετρητής προγράμματος
 - Πληροφορίες ελέγχου π.χ. κατάσταση διεργασίας, προτεραιότητα
4. **Δημιουργία κατάλληλων συνδέσεων**
 - Προσθήκη σε κατάλληλες ουρές π.χ. έτοιμων διεργασιών
5. **Δημιουργία ή επέκταση άλλων δομών δεδομένων**
 - Π.χ. δομή λογιστικών πληροφοριών



Διακοπή Διεργασίας

1. **Διακοπή (interrupt)**
 - Αιτία εξωτερική στην εκτέλεση της εκτελούμενης διεργασίας
 - Αντίδραση σε ασύγχρονο εξωτερικό γεγονός
 - Πέραν τον έλεγχο στο χειριστή διακοπών
- Παραδείγματα**
- Διακοπή ρολογιού
 - Η εκτελούμενη διεργασία εξάντλησε το επιτρεπόμενο κβάντο χρόνου
 - Αλλαγή της τρέχουσας διεργασίας σε Έτοιμη και μιας άλλης σε Εκτελούμενη
 - Διακοπή Εισόδου/Εξόδου
 - Πραγματοποίηση ενέργειας Ε/Ε
 - Αλλαγή κατάστασης αποκλεισμένης διεργασίας σε Έτοιμη ή Έτοιμη/Αναστολή

Διακοπή Διεργασίας (συνέχεια)

2. Παγίδα (trap)

- Αιτία σχετική με την εκτέλεση της τρέχουσας εντολής
- Χρησιμοποιείται για χειρισμό σφαλμάτων ή συνηθών εξαίρεσης π.χ. διαίρεση με το μηδέν, σφάλμα μνήμης
- Αν είναι μοιραία η παγίδα, η διεργασία μετακινείται σε κατάσταση Εξόδου

3. Κλήση διαχειριστή (supervisor call)

- Οφείλεται σε άμεση αίτηση για εκτέλεση υπηρεσίας ΛΣ
- Χρησιμοποιείται για την κλήση συναρτήσεων ΛΣ
- Συνήθως αλλάζει μια διεργασία από Εκτελούμενη σε Αποκλεισμένη

Αλλαγή Επιπέδου Εκτέλεσης (mode switch)

- **Κατά τη διάρκεια κάθε κύκλου εντολής**
 - Ο επεξεργαστής ελέγχει για διακοπές σε αναμονή
- **Αν υπάρχει διακοπή σε αναμονή**
 1. Αποθηκεύει το περιβάλλον του τρέχοντος προγράμματος
 2. Θέτει το μετρητή προγράμματος στην πρώτη εντολή του χειριστή διακοπών
 3. Αλλάζει το επίπεδο εκτέλεσης από χρήστη σε πυρήνα για να επιτρέψει προνομιακές εντολές
- **Χειριστής διακοπών**
 - Μικρό πρόγραμμα που εκτελεί λίγες ενέργειες σχετικές με τη διακοπή
 - Π.χ. καθαρίζει τη σημαία διακοπών, επιβεβαιώνει τη διακοπή σε μια συσκευή
- **Περιβάλλον εκτέλεσης προς αποθήκευση**
 - Ό,τι αλλάζει από το χειριστή διακοπών
 - Πληροφορίες κατάστασης επεξεργαστή π.χ. μετρητής προγράμματος
 - Υλοποίηση αποθήκευσης με υλικό επεξεργαστή για μικρή επιβάρυνση

Αλλαγή Διεργασίας (context switch)

- **Αλλαγή της εκτελούμενης διεργασίας σε άλλη κατάσταση**
 - Σημαντικές αλλαγές στο περιβάλλον εκτέλεσης
 - Σημαντική επιβάρυνση λειτουργικού συστήματος
- **Βήματα**
 1. Αποθήκευση των περιεχομένων επεξεργαστή π.χ. μετρητή προγ/τος
 2. Ενημέρωση της δομής ελέγχου διεργασίας
 3. Μετακίνηση της δομής ελέγχου διεργασίας σε άλλη ουρά
 4. Επιλογή άλλης διεργασίας για εκτέλεση
 5. Ενημέρωση της δομής ελέγχου της επιλεγμένης διεργασίας
 6. Ενημέρωση δομών δεδομένων διαχείρισης μνήμης
 7. Ανάκτηση των περιεχομένων επεξεργαστή της επιλεγμένης διεργασίας

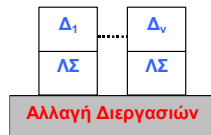
Εκτέλεση Λειτουργικού Συστήματος

- **Λειτουργικό σύστημα**
 - Κανονικό πρόγραμμα που εκτελείται από τον επεξεργαστή
- 1. **Πυρήνας μη βασισμένος σε διεργασία π.χ. παλιότερα ΛΣ**
 - Ξεχωριστή οντότητα που τρέχει σε προνομιακό επίπεδο
 - Χρησιμοποιεί μνήμη και στοίβα εκτός διεργασίας
 - Αλλαγή επιπέδου χωρίς αναγκαστική αλλαγή διεργασίας
- 2. **Λειτουργικό σύστημα βασισμένο σε διεργασίες π.χ. Minix**
 - Κύριες λειτουργίες πυρήνα ως χωριστές διεργασίες
 - Αρθρωτή δομή, ξεκάθαρες διεπαφές
 - Ευκολία δόμησης μη- κρίσιμων λειτουργιών
 - Καλύτερη απόδοση πολυεπεξεργαστών



Εκτέλεση ΛΣ (συνέχεια)

3. Εκτέλεση ΛΣ στις διεργασίες χρήστη π.χ. Unix
- Ο κώδικας πυρήνα τρέχει σε περιβάλλον διεργασιών
 - Η εικόνα διεργασίας περιλαμβάνει στοίβα πυρήνα
 - Αλλαγή επιπέδου εκτέλεσης χωρίς αναγκαστική αλλαγή διεργασίας



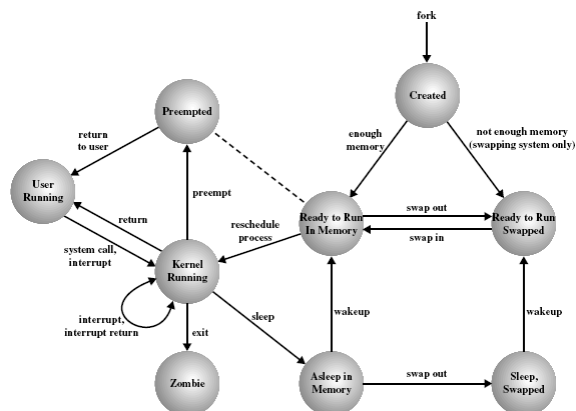
Διεργασίες Unix

- Διεργασίες χρήστη
 - Σε επίπεδο χρήστη τρέχουν προγράμματα χρήστη και εργαλεία
 - Σε επίπεδο πυρήνα εκτελούνται προνομιακές εντολές
 - Εισαγωγή σε επίπεδο πυρήνα μέσω κλήσης υπηρεσίας, παγίδας ή διακοπής
- Διεργασίες συστήματος
 - Μόνο σε επίπεδο πυρήνα
 - Εκτελούν εργασίες διαχείρισης και τακτοποίησης
- Εκτέλεση ΛΣ
 - Κυρίως γίνεται στο περιβάλλον μιας διεργασίας χρήστη

Καταστάσεις Διεργασίας

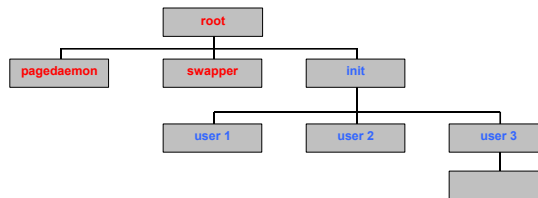
- **Δύο καταστάσεις Εκτέλεσης**
 - *Kernel Running*: εκτέλεση σε επίπεδο πυρήνα
 - *User Running*: εκτέλεση σε επίπεδο χρήστη
- **Δύο καταστάσεις Έτοιμης διεργασίας στη μνήμη**
 - *Ready to Run in Memory*: περιμένει στη μνήμη για εκτέλεση
 - *Preempted*: έτοιμη στη μνήμη μετά από ολοκλήρωση εκτέλεσης πυρήνα
- **Γιατί;**
 - Μια διεργασία που τρέχει στον πυρήνα μεταβαίνει σε κατάσταση Ready to Run in Memory πριν ξεκινήσει να τρέχει σε επίπεδο χρήστη
 - Μπορεί όμως να εκτοπιστεί όταν αλλάζει επίπεδο από πυρήνα σε χρήστη και μεταβαίνει σε κατάσταση Preempted όπου και περιμένει
- **Κατάσταση εξόδου**
 - *Zombie*: ολοκληρωμένη διεργασία που περιμένει τη γονική της να καλέσει wait()

Καταστάσεις Διεργασίας (συνέχεια)



Δέντρο Διεργασιών

- Όταν το σύστημα ξεκινάει δημιουργούνται δύο ειδικές διεργασίες
 - *swapper*: ελέγχει το πλήθος ενεργών διεργασιών στη μνήμη
 - *pagedaemon*: διατηρεί ελεύθερη μνήμη στο σύστημα
- Διεργασίες χρήστη
 - *init*: πρώτη διεργασία χρήστη - γονέας όλων των άλλων διεργασιών
 - Π.χ. δημιουργεί νέα διεργασία όταν εισάγεται διαδραστικός χρήστης



Περιγραφή Διεργασιών

- Περιεχόμενα επιπέδου χρήστη
 - *Κείμενο διεργασίας*: περιοχή με εντολές μόνο για ανάγνωση (read)
 - *Δεδομένα διεργασίας*: δεδομένα προσβάσιμα από το εκτελέσιμο
 - *Στοιβα χρήστη*: παράμετροι, τοπικές μεταβλητές, δείκτες συναρτήσεων
 - *Κοινόχρηστη Μνήμη*: μνήμη που χρησιμοποιούν οι διεργασίες από κοινού
- Περιεχόμενα καταχωρητών
 - Αποθηκεύει τα περιεχόμενα του επεξεργαστή π.χ. δείκτη στοιβάς
- Περιεχόμενα επιπέδου συστήματος
 - *Στοιβα πυρήνα*: στοιβα για εκτέλεση διαδικασιών πυρήνα
 - *Πίνακας Διεργασιών*: πληροφορίες ελέγχου από το ΛΣ
 - *Περιοχή U*: πληροφορία ελέγχου προσβάσιμη μόνο στη διεργασία
 - *Πίνακας Περιοχής Διεργασίας*: απεικόνιση εικονικών διευθύνσεων σε πραγματικές

fork()

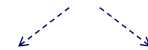
Δημιουργία διεργασίας με κλήση συστήματος fork()

1. Καταχώρηση νέου στοιχείου στον πίνακα διεργασιών
2. Ανάθεση μοναδικής ταυτότητας στη νέα διεργασία (παιδί)
3. Δημιουργία αντιγράφου του περιβάλλοντος της γονικής διεργασίας
4. Προσαύξηση του πλήθους ιδιοκτητών ανοιχτών αρχείων κατά ένα
5. Η κλήση fork() επιστρέφει
 - 0 στο παιδί & ταυτότητα παιδιού στη γονική ή
 - -1 στη γονική σε περίπτωση σφάλματος

```
if ((result = fork()) < 0) {  
    /* fork error */  
} else if (result == 0) {  
    /* κώδικας παιδιού */  
}  
/* result > 0, κώδικας γονικής */
```

γονική

Δομή Ελέγχου
→ fork();
...
Κείμενο (Text)
Δεδομένα (Data)
Στοιβά (Stack)



γονική

Δομή Ελέγχου
→ fork();
...
Κείμενο
Δεδομένα
Στοιβά

παιδί

Δομή Ελέγχου
→ fork();
...
Κείμενο
Δεδομένα
Στοιβά

Εαρινό 2024

©Σ. Β. Αναστασιάδης

29

29

exec()

Παράδειγμα αντικατάστασης προγράμματος με exec()

1. Ο πυρήνας βρίσκει το "/bin/date" στο σύστημα αρχείων
2. Το αναγνωρίζει ως εκτελέσιμο και ότι οι χρήστες έχουν πρόσβαση
3. Κατανέμει νέα περιοχή κειμένου, δεδομένων και στοιβάς
4. Αντιγράφει εντολές και δεδομένα του "/bin/date" στη διεργασία
5. Όταν το "date" τελειώνει, η γονική λαμβάνει status μέσω της wait()

```
main() {  
    int status;  
    if (fork() == 0) /* child */  
        execl("/bin/date", "date", 0);  
    wait(&status); /* parent */  
}
```

Εαρινό 2024

©Σ. Β. Αναστασιάδης

30

30

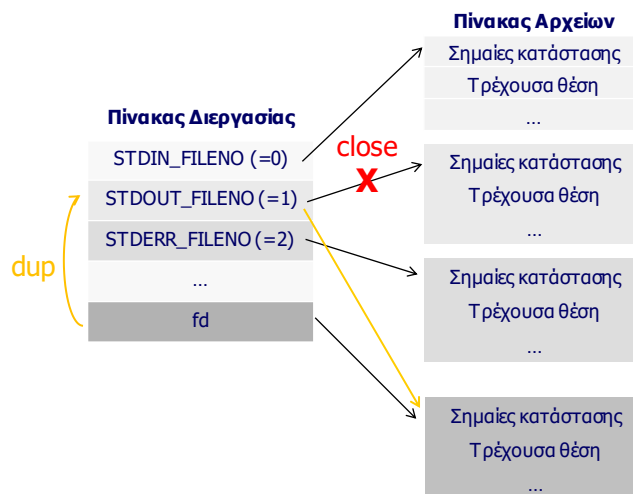
Ανακατεύθυνση Ε/Ε

Ανακατεύθυνση εξόδου σε κανονικό αρχείο (π.χ. `w > foo`)

1. Κλείνει το προκαθορισμένο αρχείο εξόδου (`STDOUT_FILENO=1`)
2. Αντιγράφει τον περιγραφέα αρχείου στην κατώτερη κενή θέση (`=1`)
3. Κλείνει τον κανονικό περιγραφέα αρχείου
4. Η κανονική έξοδος ανακατευθύνεται στο αρχείο `foo` (Ανακατεύθυνση εισόδου παρόμοια)

```
...  
fd = open("foo", O_WRONLY | O_CREAT | O_TRUNC);  
  
/* STDOUT_FILENO : POSIX stdout file descriptor (= 1) */  
close(STDOUT_FILENO);  
  
dup(fd);  
close(fd); /* ανακατεύθυνση του stdout στο αρχείο "foo" */  
...
```

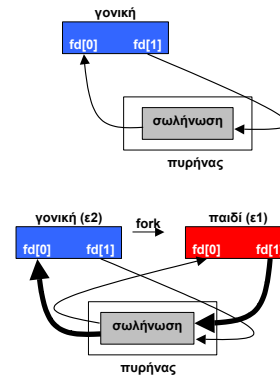
Ανακατεύθυνση Ε/Ε (συνέχεια)



Σωλήνωση

Command 1 ⇒ Command 2

1. Προκαθορισμένοι περιγραφείς αρχείων
 - STDIN_FILENO (=0)
 - STDOUT_FILENO (=1)
 - STDERR_FILENO (=2)
2. Δημιουργία σωλήνωσης στη γονική
 - Άκρα σωλήνωσης fd[0], fd[1]
 - Το παιδί κληρονομεί τη σωλήνωση
3. Στη διεργασία παιδί (εντολή 1)
 - Κλείσιμο αρχείο εξόδου (=1)
 - Αντιγραφή άκρου σωλήνωσης στην 1
4. Στη γονική διεργασία (εντολή 2)
 - Κλείσιμο αρχείου εισόδου (=0)
 - Αντιγραφή άκρου σωλήνωσης στην 0



Υλοποίηση Σωλήνωσης

```
int fd[2];
...
if (fork() == 0) {
    pipe(fd);

    if (fork() == 0) { /* εντολή 1 */
        /* προκαθορισμένη έξοδος
           σε σωλήνωση */
        close(STDOUT_FILENO);
        dup(fd[1]);

        close(fd[1]);
        close(fd[0]);

        /* exec() εντολή 1 */
        ...
    }

    /* εντολή 2 */
    /* προκαθορισμένη είσοδος
       από σωλήνωση */
    close(STDIN_FILENO);
    dup(fd[0]);

    close(fd[0]);
    close(fd[1]);

    /* exec() εντολή 2 */
    ...
}
```

Σήματα

- **Διακοπές λογισμικού**
 - Χειρισμός ασύγχρονων γεγονότων
- **Παραδείγματα**
 - Είσοδος 'Ctrl-C' από το πληκτρολόγιο (SIGINT)
 - Είσοδος 'Ctrl-\'' από το πληκτρολόγιο (SIGQUIT)
 - Είσοδος 'Ctrl-Z' από το πληκτρολόγιο (SIGTSTP)
 - Σε γενική όταν τερματίζεται η διεργασία παιδί (SIGCHLD)
 - Παραβίαση περιορισμού πρόσβασης μνήμης (SIGSEGV)
 - Αίτηση τερματισμού από υπεύθυνο συστήματος (SIGKILL)
 - Αίτηση διακοπής από υπεύθυνο συστήματος (SIGSTOP)
- **Χειρισμός σήματος**
 - `void (*signal(int signo, void (*func)(int)))(int);`
 - Επιστρέφει τον προηγούμενο χειριστή αν η κλήση επιτυχής, αλλιώς SIG_ERR

SIG_IGN

- **Αγνόηση του σήματος με SIG_IGN**
 - Εφαρμόζεται σε όλα τα σήματα εκτός από SIGKILL, SIGSTOP
 - Π.χ. αγνόηση του σήματος SIGCHLD καθαρίζει τις διεργασίες zombies

```
#include <signal.h>
main(int argc, char **argv) {
    char buf[256];

    if (argc != 1) /* με παράμετρο στη γραμμή εντολής */
        /* αγνόηση του τερματισμού παιδιού */
        signal(SIGCHLD, SIG_IGN);
    /* αλλιώς τα τερματισμένα παιδιά γίνονται zombie */
    while (read(0, buf, 256))
        /* είσοδος ENTER για δημιουργία παιδιού */
        if (fork() == 0)
            exit(0);
}
```

Χειρισμός Σήματος

- Λήψη σήματος μέσω συνάρτησης καθορισμένης από το χρήστη
 - Π.χ. `wait()` σε τερματισμένα παιδιά για αποφυγή διεργασιών zombies

```
#include <signal.h>
static void sig_chld();

int main() {
    pid_t pid;

    /* εγκατάσταση χειριστή σήματος */
    if (signal(SIGCHLD, sig_chld) == SIG_ERR)
        exit(1); /* σφάλμα signal() */

    if (pid = fork()) == 0) {
        sleep(2);
        exit(0); /* παιδί */
    }

    pause(); /* η γονική περιμένει για σήμα */
}

static void sig_chld() {
    int status;

    /* επανεγκατάσταση χειριστή σήματος,
    ώστε αν έρθει νέο σήμα να γίνει
    επανάκληση της sig_chld */
    if (signal(SIGCHLD, sig_chld) == SIG_ERR)
        exit(1);

    if (wait(&status) < 0)
        exit(1); /* επιστρέφει σφάλμα */
    /* επιστροφή της κλήσης pause() */
}
```

Εαρινό 2024

©Σ. Β. Αναστασιάδης

37

37

Διακλάδωση Μεταξύ Συναρτήσεων

- `int setjmp(jmp_buf env)`
 - Αρχικοποίηση της `env` με την τρέχουσα θέση
 - Επιστροφή της τιμής 0 κατά την αρχικοποίηση
 - Επιστροφή της τιμής `val` με κλήση της `longjmp(env, val)`
- `void longjmp(jmp_buf env, int val)`
 - Διακλάδωση στην `setjmp()` που αρχικοποίησε την `env`
 - Επιστροφή τιμής `val` από την `setjmp()`

```
#include <setjmp.h>
jmp_buf jmpbuffer;

int main() {
    /* = 0 στην πρώτη κλήση */
    if (setjmp(jmpbuffer) != 0)
        /* = 1 μετά από longjmp */
        exit(1);
    ...
    /* εκτέλεση καλεί foo() */
    exit(0);
}

void foo(void) {
    /* κανονική εκτέλεση */
    if (/* σφάλμα */) {
        /* διακλάδωση στην setjmp */
        longjmp(jmpbuffer, 1);
        /* δεν εκτελείται */
    }
    /* συνέχεια κανονικής εκτέλεσης */
}
```

Εαρινό 2024

©Σ. Β. Αναστασιάδης

38

38