ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Λειτουργικά Συστήματα

Ενότητα # 8: Το ΛΣ Linux

Διδάσκων: Γεώργιος Ξυλωμένος

Τμήμα: Πληροφορικής

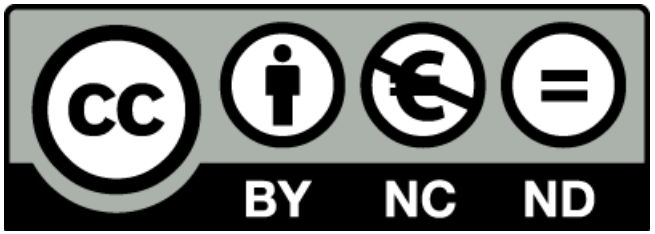
Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» 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.

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται από το βιβλίο «Σύγχρονα Λειτουργικά Συστήματα», A.S. Tanenbaum, 4η έκδοση, 2018, Εκδόσεις Κλειδάριθμος.



Σκοποί ενότητας

- Κατανόηση της ιστορικής εξέλιξης που οδήγησε από το UNIX στο Linux
- Εισαγωγή στη βασική δομή και σχεδίαση του Linux
- Κατανόηση του τρόπου με τον οποίο οι γενικές αρχές ΛΣ εφαρμόζονται στο Linux στους τομείς των διεργασιών, διαχείρισης μνήμης, εισόδου / εξόδου, συστήματος αρχείων και ασφάλειας

Περιεχόμενα ενότητας

- Ιστορία UNIX/Linux
- Γενικά για το Linux
- Διεργασίες στο Linux
- Διαχείριση μνήμης στο Linux
- Είσοδος / έξοδος στο Linux
- Σύστημα αρχείων στο Linux
- Ασφάλεια στο Linux

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Ιστορία UNIX/Linux

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 8:** Το ΛΣ Linux

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

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

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Εισαγωγή

- Γιατί να μελετήσουμε το Linux;
  - Πιο δημοφιλής παραλλαγή του UNIX
  - Υπάρχουν πολλές άλλες παραλλαγές
    - AIX, Free/Net/OpenBSD, HP-UX, SCO UNIX, Solaris
    - Κοινή βασική δομή (εν μέρει)
    - Κοινό βασικό σύνολο κλήσεων συστήματος
    - Κάθε σύστημα έχει διαφοροποιήσεις
    - Για να είμαστε σαφείς, διαλέξαμε το Linux

UNICS

- Η προϊστορία του UNIX
  - CTSS (MIT): πρωτοπόρο σύστημα χρονομερισμού
  - MIT, Bell Labs και GE σχεδίασαν το MULTICS
    - Πολύ φιλόδοξο πρόγραμμα για χρονομερισμό
  - Τα Bell Labs αποσύρθηκαν από το MULTICS
    - Η GE αποσύρθηκε από τους υπολογιστές!
  - UNICS: απλή έκδοση του MULTICS για PDP-7
    - Για να εκτελεί προγράμματα του MULTICS
    - Τελικά το όνομα κατέληξε να είναι UNIX

PDP-11 UNIX

- Σύντομα το UNIX μεταφέρθηκε στο PDP-11
  - Διέθετε υλικό προστασίας μνήμης και αρκετή μνήμη
  - Ήταν όμως 16 bit: 64 KB πρόγραμμα + 64 KB δεδομένα
- Η μεταφορά οδήγησε στην επινόηση και της C
  - Η συγγραφή ΛΣ σε assembly ήταν χρονοβόρα
  - Η C σχεδιάστηκε ειδικά για αυτή τη δουλειά
  - Το UNIX ξαναγράφτηκε σε C
- Το UNIX έγινε δημοφιλές αν και δεν ήταν προϊόν
  - Η AT&T δεν μπορούσε να μπει στην αγορά
  - Έδινε άδειες (με πηγαίο κώδικα) σχεδόν δωρεάν

Φορητό UNIX (1 από 2)

- Το UNIX Version 6 ήταν ένα μικρό σύστημα
  - 8200 γραμμές C και 900 γραμμές assembly
- Το Version 7 ήταν το πρώτο φορητό UNIX
  - 18800 γραμμές C και 2100 γραμμές assembly
  - Βάση για πάρα πολλές παραλλαγές (π.χ. XENIX)
  - Εκτός από PDP-11 έτρεχε και σε Interdata 8/32
    - Το Interdata ήταν σύστημα 32 bit
    - Αποκάλυψε πολλές εξαρτήσεις από το PDP-11
    - Το UNIX έτσι έγινε πολύ πιο φορητό

Φορητό UNIX (2 από 2)

- Φορητός μεταγλωττιστής της C
  - Δημιουργήθηκε παράλληλα με το φορητό UNIX
  - Παρήγαγε κώδικα για διάφορες μηχανές
    - Μικρές αλλαγές για χρήση σε νέα μηχανή
  - Επέτρεπε τη μεταγλώττιση του UNIX
    - Συνήθως μεταγλωττιστής και UNIX πήγαιναν μαζί
- Τελικά η AT&T άρχισε να πουλάει UNIX
  - Αρχικά το System III
  - Στη συνέχεια το System V Release 2, 3, 4

Berkeley UNIX

- Το Berkeley ξεκίνησε από το Version 7 σε PDP-11
- Πολλές προσθήκες με χρηματοδότηση DARPA
  - Το αποτέλεσμα ήταν το BSD UNIX
  - Η DARPA ενθάρρυνε τη χρήση του στο ARPAnet
- Η έκδοση 4BSD ήταν ιδιαίτερα σημαντική
  - Έτρεχε στα VAX που ήταν εξαιρετικά δημοφιλή
  - Εικονική μνήμη με σελιδοποίηση
  - Γρήγορο σύστημα αρχείων με μεγάλα ονόματα
  - Υλοποίηση TCP/IP και υποδοχών (sockets)
  - Πολλές νέες εφαρμογές (csh, vi)

Πρότυπο UNIX (1 από 2)

- Οι εταιρείες έφτιαξαν δικές τους εκδόσεις
  - SunOS και Ultrix ήταν εξελιγμένα BSD
  - Solaris και OSF/1 βασίζονταν στο System V
- Στα τέλη του ‘80 υπήρχαν πολλά UNIX
  - Παραλλαγές των System V και BSD
  - Το καθένα με ελαφρά διαφορετικές κλήσεις
    - Το MS-DOS είχε συμβατότητα σε δυαδικό επίπεδο
  - Οι πρώτες προσπάθειες τυποποίησης απέτυχαν

Πρότυπο UNIX (2 από 2)

- Το πρότυπο POSIX
  - Προσπάθεια υπό την αιγίδα της IEEE
  - Προδιαγράφει τις βασικές κλήσεις βιβλιοθήκης
  - Αφήνει στις υλοποιήσεις τις κλήσεις συστήματος
  - Περιέχει τον κοινό πυρήνα System V και BSD
  - Μοιάζει αρκετά με το UNIX Version 7
  - Επιπλέον πρότυπα για προγράμματα και δικτύωση
  - Δεν περιγράφει πλήρες σύστημα UNIX

MINIX (1 από 2)

- Το UNIX δεν ήταν κατάλληλο για μελέτη
  - Οι εταιρείες δεν έδιναν άδειες πηγαίου κώδικα
  - Αδύνατον να κατανοηθεί τόσο μεγάλο σύστημα
- MINIX: επιστροφή στο Version 7 UNIX
  - 11800 γραμμές C και 800 γραμμές assembly
  - Διαθέσιμο σε 80386 με πηγαίο κώδικα
  - Ίδια διεπαφή με Version 7 (κλήσεις συστήματος)
  - Υλοποίηση με μικροπυρήνα
  - Πλήρης περιγραφή του σε βιβλίο

MINIX (2 από 2)

- Το MINIX 2.0 ήταν συλλογική προσπάθεια
  - Πολλοί είχαν συνεισφέρει κώδικα
  - Το MINIX 1.0 ήταν έργο του A. Tanenbaum
- Το MINIX 3.0 ήταν πολύ πιο αξιόπιστο
  - Καθαρή υλοποίηση μικροπυρήνα
  - Οι οδηγοί συσκευών έφυγαν από τον πυρήνα
- Το (παλιό) MINIX όμως δεν ήταν για όλους
  - Δεν είχε την απόδοση του μονολιθικού πυρήνα
  - Ήταν (επίτηδες) απλό, χωρίς πολλά χαρακτηριστικά

Linux

- Νέα υλοποίηση ενός συστήματος UNIX
  - Βασίστηκε στο MINIX αλλά με μονολιθικό πυρήνα
  - Ο στόχος ήταν ένα ΛΣ παραγωγής
    - Υψηλές επιδόσεις και κάλυψη πολλών επεξεργαστών
- Το Linux αναπτύχθηκε πολύ γρήγορα
  - Προσθήκη νέων δυνατοτήτων και συσκευών
  - Εκδόσεις για πολλές διαφορετικές μηχανές
  - Μεταφέρθηκαν πολλά προγράμματα για UNIX
  - Γράφτηκαν όμως και νέα (π.χ. KDE και GNOME)

Linux ή xBSD;

- Το Linux διανέμεται με την άδεια GPL
  - Μπορεί κανείς να διανείμει τροποποιήσεις
  - Πρέπει όμως να παρέχει τον πηγαίο κώδικα
- Η ανάπτυξη του Berkeley UNIX σταμάτησε το 1992
  - Το 4.4BSD είχε άδεια ανοιχτού λογισμικού (όχι GPL)
  - Δεν είχε καθόλου κώδικα της AT&T, αλλά μηνύθηκε!
- Το Linux κάλυψε το κενό του BSD
  - Γράφτηκε από το μηδέν χωρίς νομικά προβλήματα
  - Ήταν σύστημα παραγωγής σε αντίθεση με το MINIX

Γενικά για το Linux

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 8:** Το ΛΣ Linux

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

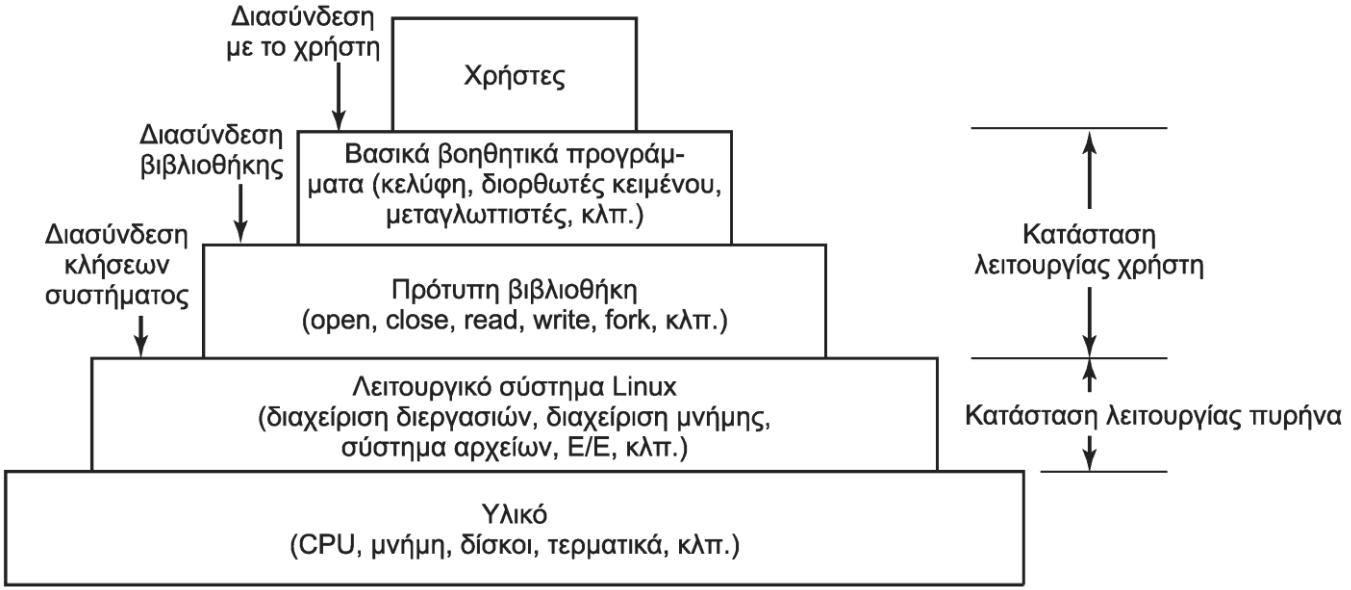
Στόχοι του Linux (1 από 2)

- Σύστημα αλληλεπίδρασης
- Πολλών χρηστών / πολλών διεργασιών
- Σχεδιάστηκε για προγραμματιστές
  - Πολλά εργαλεία ανάπτυξης λογισμικού
  - Υποστήριξη ανάπτυξης σε ομάδες
  - Διευκόλυνση έμπειρων χρηστών
- Απλό, κομψό και συνεπές
  - Ένα μόνο είδος αρχείων (ακολουθία byte)
  - ls A* έχει παρόμοια σημασία με rm A* (ομοιομορφία)

Στόχοι του Linux (2 από 2)

- Συγκεκριμένη φιλοσοφία προγραμμάτων
  - Κάθε πρόγραμμα κάνει μία απλή εργασία
  - Συνδυασμός για σύνθετες εργασίες
    - Ισχυρές μέθοδοι συνδυασμού προγραμμάτων
  - Σύντομα ονόματα προγραμμάτων (cp, όχι copy)
    - Λήψη πληροφοριών από γραμμή εντολών
  - Λίγες ερωτήσεις προς το χρήστη
    - Θεωρεί ότι ο χρήστης ξέρει τι κάνει

Διασυνδέσεις του Linux (1 από 2)



- Λειτουργικό: κλήσεις συστήματος
  - Ορίσματα σε καταχωρητές ή στοίβα
  - Εντολή TRAP για είσοδο στον πυρήνα
- Πρότυπη βιβλιοθήκη: υπερσύνολο κλήσεων POSIX

Διασυνδέσεις του Linux (2 από 2)

- Βοηθητικά προγράμματα: εν μέρει στο POSIX
  - Κέλυφος, μεταγλωττιστές, διορθωτές, χειρισμός αρχείων
- Γραφικές διεπαφές: αντί για γραμμή εντολών
  - Βασίζεται στο X Window System (ή X11 ή X)
    - Ο διακομιστής X ελέγχει τις συσκευές
    - Τα προγράμματα είναι οι πελάτες X
  - Το X σταδιακά αντικαθίσταται από το Wayland
  - Χρειάζεται και διαχειριστή παραθύρων
    - GNOME: GNU Network Object Model Environment
    - KDE: K Desktop Environment

Το κέλυφος (1 από 3)

- Το πρώτο κέλυφος του UNIX λεγόταν sh
  - Λέγεται και Bourne shell για να διακριθεί από τα άλλα
- Το κέλυφος του Linux λέγεται bash (Bourne again shell)
- Εμφανίζει προτροπή (prompt) και περιμένει είσοδο
- Ερμηνεύει την πρώτη λέξη ως όνομα εντολής
  - Οι επόμενες ερμηνεύονται ως παράμετροι
- Συνήθως με τον χαρακτήρα – δίνονται σημαίες
- Χρήση χαρακτήρων μπαλαντέρ *, ?, [-]
  - Παράδειγμα: ls *.c ή ls [ape]*

Το κέλυφος (2 από 3)

- Κάθε πρόγραμμα ξεκινά με τρία ανοιχτά αρχεία
  - Καθιερωμένη είσοδος, έξοδος και έξοδος σφαλμάτων
  - Αρχικά αντιστοιχούν σε πληκτρολόγιο και οθόνη
  - Τα φίλτρα (filters) διαβάζουν και γράφουν από εκεί
- Ανακατεύθυνση των αρχείων με < και >
  - Παράδειγμα: sort <in >out
  - Τα σφάλματα πηγαίνουν στην οθόνη
- Αγωγοί εντολών (pipelines) με |
  - Παράδειγμα: sort <in | head -30
  - Δεν χρειάζεται χρήση ενδιάμεσων αρχείων

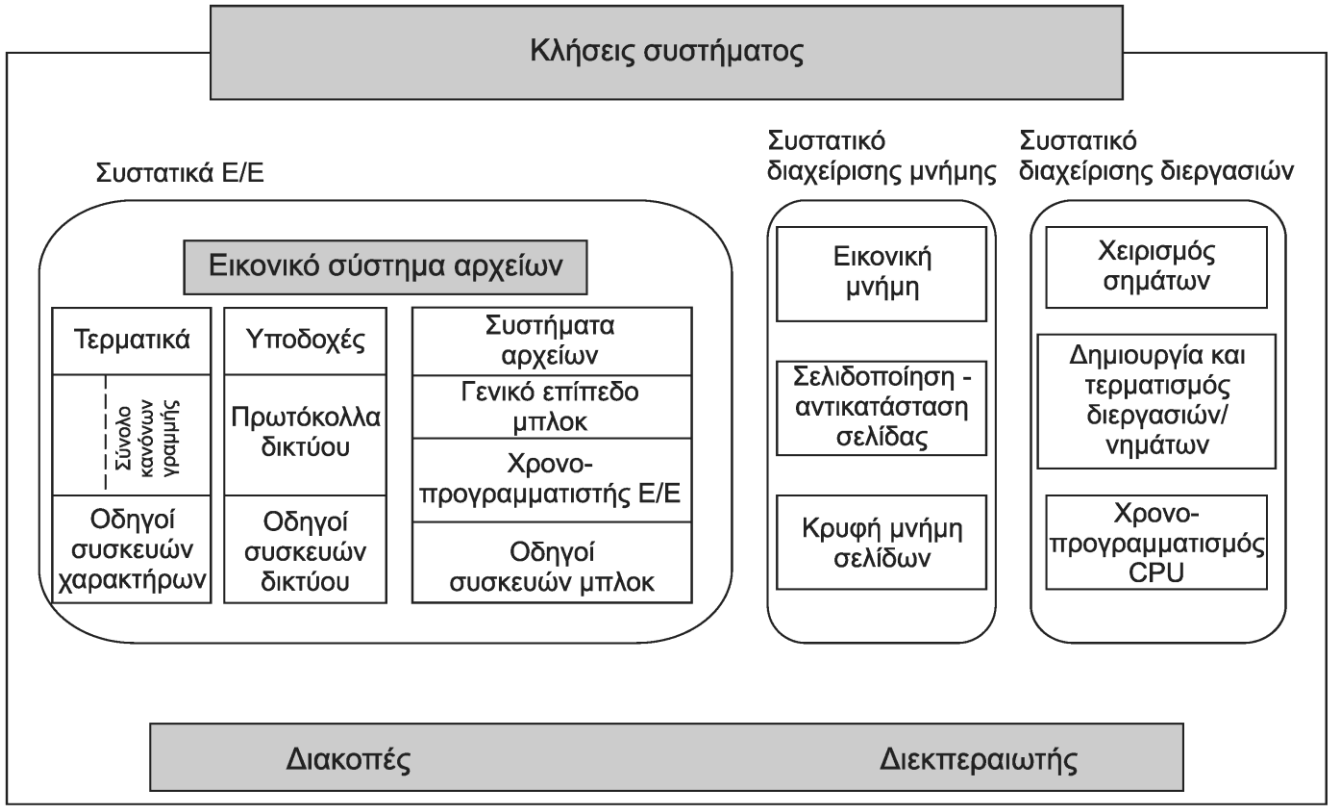
Το κέλυφος (3 από 3)

- Εκτέλεση στο παρασκήνιο με &
  - Παράδειγμα: wc –l <a >b &
  - Δίνει αμέσως τον έλεγχο πίσω στο χρήστη
    - Ο χρήστης μπορεί να χειριστεί το πρόγραμμα
- Σενάρια κελύφους (shell scripts)
  - Αρχεία με εντολές κελύφους προς εκτέλεση
    - Εκτελούνται από νέο κέλυφος
  - Μπορούν να παίρνουν παραμέτρους
  - Διαθέτουν μεταβλητές και δομές ελέγχου

Βοηθητικά προγράμματα

- Βοηθητικά προγράμματα του Linux
  - Το POSIX περιγράφει περίπου 100
    - Σύνταξη και σημασιολογία
  - Χειρισμός αρχείων και καταλόγων
  - Διαχείριση συστήματος
  - Ανάπτυξη προγραμμάτων
  - Επεξεργασίας κειμένου
  - Φίλτρα μετασχηματισμού δεδομένων

Δομή πυρήνα (1 από 3)



- Κατώτερο επίπεδο και τρία συστατικά
  - Διεργασίες, μνήμη, είσοδος / έξοδος

Δομή πυρήνα (2 από 3)

- Διακοπές και διεκπεραιωτής διεργασιών
  - Κατώτερο επίπεδο πυρήνα
  - Χειρισμός διακοπών (και) σε γλώσσα μηχανής
- Συστατικό διαχείρισης εισόδου-εξόδου
  - Εικονικό σύστημα αρχείων στο υψηλότερο επίπεδο
  - Οδηγοί συσκευών στο χαμηλότερο επίπεδο
  - Συσκευές χαρακτήρων: κανόνες γραμμής
  - Συσκευές δικτύου: πρωτόκολλα, υποδοχές
  - Συσκευές μπλοκ: συστήματα αρχείων

Δομή πυρήνα (3 από 3)

- Συστατικό διαχείρισης μνήμης
  - Απεικόνιση διευθύνσεων, κρυφή μνήμη σελίδων
- Συστατικό διαχείρισης διεργασιών
  - Δημιουργία/τερματισμός διεργασιών
  - Χρονοπρογραμματισμός, σήματα
- Αλληλεξαρτήσεις μεταξύ συστατικών
- Δυναμικά φορτώσιμες υπομονάδες
- Διασύνδεση κλήσεων συστήματος

Διεργασίες στο Linux

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 8:** Το ΛΣ Linux

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Έννοιες διεργασιών (1 από 5)

- Κάθε διεργασία εκτελεί ένα μόνο πρόγραμμα
- Αρχικά έχει ένα νήμα ελέγχου
  - Μπορεί στη συνέχεια να φτιάξει άλλα νήματα
- Πολλές ταυτόχρονες διεργασίες ανά χρήστη
- Διεργασίες δαίμονες
  - Παράδειγμα: το cron ξυπνάει κάθε 1 λεπτό
    - Ελέγχει αν υπάρχουν προγραμματισμένες εργασίες
    - Υπενθυμίσεις ή περιοδικές συντηρήσεις

Έννοιες διεργασιών (2 από 5)

- Δημιουργία νέων διεργασιών με fork
  - Η καλούσα διεργασία η γονική (parent)
  - Η νέα διεργασία είναι η θυγατρική (child)
  - Αρχικά έχουν τις ίδιες εικόνες στη μνήμη
    - Κώδικας και δεδομένα
    - Αλλαγές στη μία δεν επηρεάζουν την άλλη
  - Κοινά ανοιχτά αρχεία στις δύο διεργασίες
    - Μοιράζονται και τον δείκτη τρέχουσας θέσης

Έννοιες διεργασιών (3 από 5)

```
pid = fork();           /* αν η fork επιτύχει, ένα PID > 0 επιστρέφεται στη μητρική διεργασία */
if (pid < 0) {
    handle_error();     /* η fork απέτυχε (π.χ. λόγω έλλειψης μνήμης ή γιατί κάποιος */
                        /* πίνακας του συστήματος ήταν γεμάτος */

} else if (pid > 0) {
                        /* ο κώδικας της μητρικής διεργασίας τοποθετείται εδώ */
} else {
                        /* ο κώδικας της θυγατρικής διεργασίας τοποθετείται εδώ */
}
```

- Κάθε διεργασία έχει μία ταυτότητα (PID)
  - Η fork επιστρέφει το PID της θυγατρικής
    - Στη θυγατρική διεργασία επιστρέφει 0
  - Η διεργασία μαθαίνει το PID μέσω της getpid
  - Με το PID η διεργασία διακρίνει τα παιδιά της
    - Επιστρέφεται στη γονική και κατά τον τερματισμό

Έννοιες διεργασιών (4 από 5)

- Επικοινωνία διεργασιών με αγωγούς (pipes)
  - Μονόδρομη ροή byte ανάμεσα σε δύο διεργασίες
  - Μπλοκάρισμα συγγραφέα σε γεμάτο αγωγό
  - Μπλοκάρισμα αναγνώστη σε άδειο αγωγό
- Επικοινωνία διεργασιών μέσω σημάτων (signals)
  - Ουσιαστικά πρόκειται για διακοπές λογισμικού
    - Είτε από άλλες διεργασίες είτε από το σύστημα
  - Η διεργασία ορίζει τι θα κάνει σε κάθε σήμα
    - Τερματισμός, αγνόηση, χειρισμός με διαδικασία

Έννοιες διεργασιών (5 από 5)

- Ομάδα διεργασιών (process group)
  - Οι απόγονοι μίας διεργασίας
  - Σήματα στέλνονται μόνο μέσα στην ομάδα
  - Δυνατότητα αποστολής σε όλη την ομάδα
- Σήματα POSIX
  - Το POSIX ορίζει σειρά τυποποιημένων σημάτων
  - Σήματα που στέλνονται από διεργασίες (SIGUSR)
  - Σήματα ελέγχου της διεργασίας (SIGHUP)
  - Σφάλματα προγράμματος (SIGILL, SIGSEGV, SIGFPE)
  - Σήματα που προέρχονται από το πρόγραμμα (SIGALRM)

Κλήσεις διεργασιών (1 από 4)

Κλήση	Περιγραφή
pid = fork()	Δημιουργία θυγατρικής διεργασίας, πανομοιότυπης με τη μητρική
pid = waitpid(pid, &statloc, opts)	Αναμονή μέχρι τον τερματισμό της θυγατρικής διεργασίας
s = execve(name, argv, envp)	Αντικατάσταση του ειδώλου πυρήνα μιας διεργασίας
exit(status)	Τερματισμός εκτέλεσης της διεργασίας και επιστροφή κωδικού κατάστασης
s = sigaction(sig, &act, &oldact)	Καθορισμός της ενέργειας του σήματος
s = sigreturn(&context)	Επιστροφή από ένα σήμα
s = sigprocmask(how, &set, &old)	Εξέταση ή αλλαγή της μάσκας του σήματος
s = sigpending(set)	Λήψη του συνόλου των μπλοκαρισμένων σημάτων
s = sigsuspend(sigmask)	Αντικατάσταση μάσκας σήματος και μπλοκάρισμα της διεργασίας
s = kill(pid, sig)	Αποστολή του σήματος σε μια διεργασία
residual = alarm(seconds)	Ανάθεση τιμής στο χρονόμετρο συναγερμού
s = pause()	Αναστολή καλούντος μέχρι το επόμενο σήμα

- Δημιουργία αντιγράφων διεργασιών με fork
- Αναμονή τερματισμού θυγατρικής με waitpid
- Εκτέλεση νέου προγράμματος με exec

Κλήσεις διεργασιών (2 από 4)

```
while (TRUE) {                                  /* επανάληψη συνεχώς */
    type_prompt();                              /* εκτύπωση του προτρεπτικού μηνύματος στην οθόνη */
    read_command(command, parameters);          /* ανάγνωση της εισόδου από το τερματικό */
    pid = fork();                               /* δημιουργία θυγατρικής διεργασίας */
    if (pid < 0) {
        printf("Unable to fork");               /* συνθήκη σφάλματος */
        continue;                               /* επανάληψη βρόχου */
    }
    if (pid!= 0) {
        waitpid(–1, &status, 0);                /* η μητρική περιμένει τη θυγατρική */
    } else {
        execve (command, parameters, 0);        /* η θυγατρική κάνει τη δουλειά */
    }
}
```

- Παράδειγμα: ένας απλός φλοιός
  - Εκτύπωση προτροπής και ανάγνωση εντολών
  - Δημιουργία θυγατρικής διεργασίας
  - Η θυγατρική εκτελεί την εντολή του χρήστη

Κλήσεις διεργασιών (3 από 4)

- Η exec παίρνει τρεις παραμέτρους
  - Όνομα προγράμματος προς εκτέλεση
  - Μεταβλητές προγράμματος: παράμετροι κλήσης
  - Μεταβλητές περιβάλλοντος: π.χ. τύπος τερματικού
  - Ο καλούμενος ξεκινάει με main(argc, argv, envp)
- Έξοδος από διεργασία με την exit
  - Παράμετρος που ορίζεται από το χρήστη (0-255)
    - Επιστρέφεται στο υψηλό byte της status στην waitpid
    - Στο χαμηλό byte επιστρέφεται ένας κωδικός συστήματος
- Μια διεργασία πρέπει να περιμένει τα παιδιά της

Κλήσεις διεργασιών (4 από 4)

- Διαχείριση σημάτων με την sigaction
  - Σήμα που θέλουμε να χειριστούμε
  - Δομή με δείκτη προς χειριστή σήματος και σημαίες
  - Επιστρέφεται παλιός χειριστής και σημαίες
- Αποστολή σημάτων με την kill
  - Αν δεν συλληφθεί, τερματίζει τον παραλήπτη
- Ρύθμιση “ξυπνητηριών” με την alarm
  - Ορίζουμε σε πόση ώρα θέλουμε το SIGALARM
- Αναμονή μέχρι να συμβεί κάτι με την pause

Διεργασίες και νήματα (1 από 9)

- Διεργασία: τμήμα χρήστη και τμήμα πυρήνα
  - Τμήμα χρήστη: εκτελεί το πρόγραμμα του χρήστη
  - Τμήμα πυρήνα: χρήση σε κλήσεις συστήματος
    - Στις κλήσεις συστήματος εκτελείται κώδικας πυρήνα
    - Ανεξάρτητη στοίβα και μετρητής προγράμματος
    - Επιτρέπει διακοπή της διεργασίας μέσα στον πυρήνα
- Στο Linux νήματα=διεργασίες=εργασίες (tasks)
  - Κάθε εργασία αντιστοιχεί σε μία δομή εργασίας
  - Μια πολυνηματική διεργασία έχει πολλές δομές
  - Και ο πυρήνας είναι πολυνηματικός

Διεργασίες και νήματα (2 από 9)

- Κάθε εργασία έχει μία δομή μόνιμα στη μνήμη
  - Οργανώνονται σε διπλά συνδεδεμένη λίστα
  - Κάθε εργασία έχει ένα PID και ένα TID
- Περιεχόμενα δομής εργασίας στο Linux
  - Παράμετροι χρονοπρογραμματισμού (χρόνος CPU)
  - Εικόνα μνήμης (δείκτες σε πίνακες σελίδων ή δίσκο)
  - Σήματα (μάσκες και κατάσταση σημάτων)
  - Καταχωρητές μηχανής (όταν η διεργασία σταματάει)

Διεργασίες και νήματα (3 από 9)

- Περιεχόμενα δομής εργασίας στο Linux
  - Κατάσταση κλήσης συστήματος (τρέχουσα κλήση)
  - Πίνακας περιγραφέων αρχείων (ανοιχτά αρχεία)
  - Στοίβα πυρήνα (για το τμήμα πυρήνα)
  - Διάφορα (κατάσταση, συμβάν που αναμένεται, PID)
- Μέρος της δομής μπορεί να πηγαίνει στο δίσκο
  - Τα σήματα πρέπει να είναι συνέχεια στη μνήμη
  - Οι περιγραφείς αρχείων μπορούν να είναι στο δίσκο

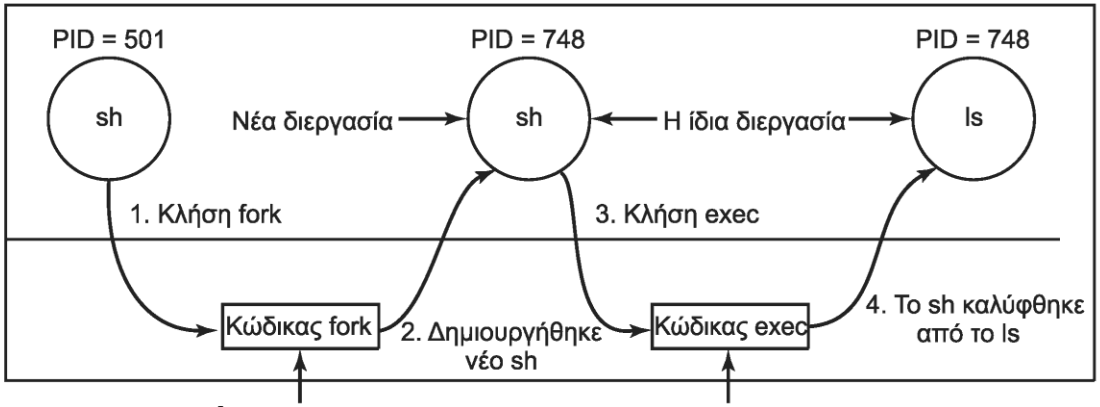
Διεργασίες και νήματα (4 από 9)

- Υλοποίηση κλήσης fork (πρώτο μέρος)
  - Δημιουργία νέας δομής εργασίας
  - Δημιουργία πληροφοριών νήματος
    - Περιέχει δείκτη προς τη δομή εργασίας
  - Δημιουργία στοίβας κατάστασης πυρήνα
  - Συμπλήρωση δομών από τη μητρική διεργασία
  - Εισαγωγή PID σε πίνακα κατακερματισμού
    - Μπορούμε από το PID να βρούμε άμεσα τη διεργασία
  - Σύνδεση δομής εργασίας στη λίστα των εργασιών

Διεργασίες και νήματα (5 από 9)

- Υλοποίηση κλήσης fork (δεύτερο μέρος)
  - Ορισμός τμήματος κώδικα ως κοινόχρηστου
  - Δημιουργία πίνακα σελίδων για τμήμα δεδομένων
  - Σημείωση σελίδων δεδομένων ανάγνωσης μόνο
    - Ανεξάρτητα από το αν είναι όντως ανάγνωσης μόνο
  - Όταν γραφτούν, δημιουργείται παγίδα
    - Η σελίδα αντιγράφεται σε νέο πλαίσιο
    - Οι σελίδες σημειώνονται ως ανάγνωσης και εγγραφής
    - Ουσιαστικά αντιγράφονται μόνο αν τροποποιηθούν

Διεργασίες και νήματα (6 από 9)



- Υλοποίηση κλήσης exec
  - Εντοπισμός και επαλήθευση του κώδικα προς εκτέλεση
  - Αντιγραφή ορισμάτων και περιβάλλοντος στον πυρήνα
  - Απελευθέρωση μνήμης και δημιουργία νέου χάρτη
    - Με απεικόνιση του αρχείου κώδικα στη μνήμη
  - Αντιγραφή ορισμάτων και περιβάλλοντος σε στοίβα
  - Μηδενισμός σημάτων και καταχωρητών

Διεργασίες και νήματα (7 από 9)

- Τα νήματα στο Linux
  - Τα νήματα περιπλέκουν το μοντέλο του UNIX
  - Πρέπει η fork να αντιγράφει τα νήματα;
  - Πρέπει να δέχονται τα νήματα τα σήματα;
- Η κλήση clone στο Linux
  - Η clone είναι μια εξελιγμένη μορφή της fork
  - pid=clone(function, stack_ptr, sharing_flags, arg);
    - Ανάλογα με τη sharing_flags έχουμε ή όχι κοινό χώρο μνήμης
  - Η νέα εργασία ξεκινά στη function με παράμετρο arg
  - Η στοίβα της νέας εργασίας ξεκινά στην stack_ptr

Διεργασίες και νήματα (8 από 9)

Σημαία	Σημασία της τιμής 1	Σημασία της τιμής 0
CLONE_VM	Δημιουργεί ένα νέο νήμα	Δημιουργεί μια νέα διεργασία
CLONE_FS	Κοινή χρήση umask, και βασικού και τρέχοντος καταλόγου	Όχι κοινή χρήση
CLONE_FILES	Κοινή χρήση περιγραφέων αρχείων	Αντιγραφή των περιγραφέων αρχείων
CLONE_SIGHAND	Κοινή χρήση πίνακα χειρισμού σημάτων	Αντιγραφή του πίνακα
CLONE_PARENT	Το νέο νήμα έχει τον ίδιο γονέα με τον καλούντα	Γονέας του νέου νήματος είναι ο καλών

- Η sharing_flags (χάρτης bit) δείχνει τι θα συμβεί
  - CLONE_VM: κοινή μνήμη ή αντίγραφο μνήμης
  - CLONE_FS: κοινοί κατάλογοι και μάσκα αρχείων ή όχι
  - CLONE_FILES: κοινά αρχεία ή αντίγραφα περιγραφέων
  - CLONE_SIGHAND: κοινός πίνακας σημάτων ή αντίγραφο
  - CLONE_PARENT: ίδιος γονέας ή ο καλών είναι ο γονέας

Διεργασίες και νήματα (9 από 9)

- Οι δομές εργασιών δείχνουν σε άλλες δομές
  - Για μνήμη, αρχεία, σήματα…
  - Έτσι έχουμε τόσα κοινά στοιχεία στην clone
  - Χάνουμε τη συμβατότητα με το κλασικό UNIX
- Ταυτότητα εργασίας (TID) - διεργασίας (PID)
  - Κάθε εργασία έχει διαφορετικό (μοναδικό) TID
  - Μπορεί να έχει ίδιο PID με άλλη για συμβατότητα
    - Όλα τα νήματα μίας διεργασίας έχουν το ίδιο PID

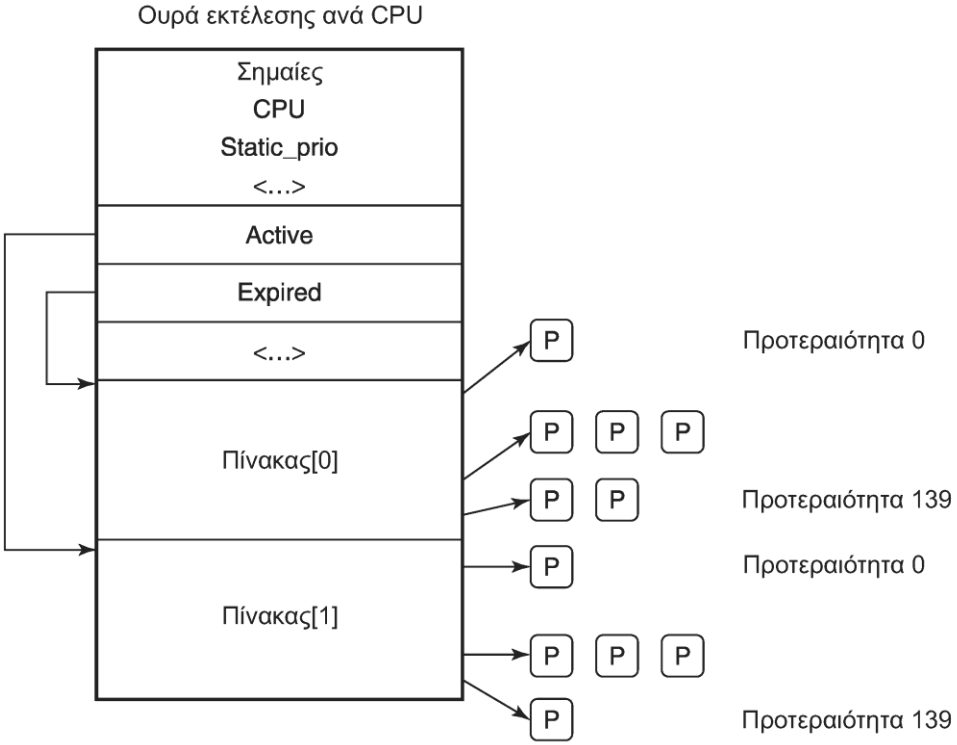
Χρονοπρογραμματισμός (1 από 8)

- Χρονοπρογραμματισμός σε επίπεδο εργασιών
  - Πραγματικού χρόνου FIFO
  - Πραγματικού χρόνου εκ περιτροπής
  - (απλού) Χρονομερισμού
- Οι πραγματικού χρόνου FIFO εκτελούνται πρώτα
  - Προεκτόπιση μόνο από εργασίες FIFO
- Μετά οι πραγματικού χρόνου εκ περιτροπής
  - Κάθε μία εκτελείται για ένα κβάντο
  - Τα κβάντα δίνονται σε χτύπους ρολογιού (jiffies)
    - Συχνότητες 1000, 500, 250, 1 Hz

Χρονοπρογραμματισμός (2 από 8)

- Δεν είναι βέβαια πραγματικού χρόνου
  - Απλά έχουν υψηλότερη προτεραιότητα!
  - Επίπεδα από 0 (υψηλότερο) έως 99 (χαμηλότερο)
- Τέλος εκτελούνται οι άλλες εργασίες
  - Έχουν προτεραιότητα 100 έως 139
  - Η βασική προτεραιότητα είναι 20 (δηλαδή 120)
  - Οι χρήστες μπορούν να προσθέσουν 1-19
    - Με την κλήση nice
  - Ο υπερχρήστης μπορεί να αφαιρέσει 1-20

Χρονοπρογραμματισμός (3 από 8)



- Κλασικός χρονοπρογραμματιστής
  - Μία ουρά εκτέλεσης ανά επεξεργαστή (runqueue)
  - Πίνακες ενεργών (active) / ληγμένων (expired) εργασιών

Χρονοπρογραμματισμός (4 από 8)

- Χρήση της ουράς εκτέλεσης
  - Επιλογή εργασιών από τον ενεργό πίνακα
    - Επιλέγεται αυτή με την υψηλότερη προτεραιότητα
  - Αν λήξει το κβάντο της πηγαίνει στον ληγμένο πίνακα
    - Αν μπλοκαριστεί, παραμένει στον ενεργό
    - Μπορεί να εκτελεστεί για το υπόλοιπο του κβάντου της
  - Όταν αδειάσει, ο ενεργός αλλάζει με τον ληγμένο
  - Έτσι όλες οι διεργασίες τελικά θα εκτελεστούν

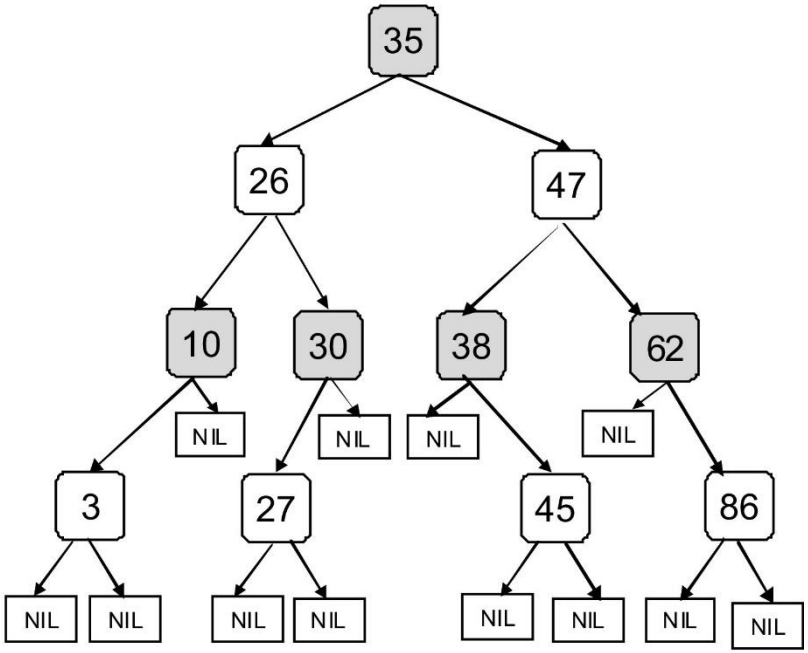
Χρονοπρογραμματισμός (5 από 8)

- Καθορισμός προτεραιοτήτων
  - Μικρότερο κβάντο σε χαμηλότερες προτεραιότητες
  - Κάθε εργασία έχει μία μεταβλητή sleep_avg (+/-5)
    - Αύξηση όσο κοιμάται, μείωση όσο τρέχει
  - Αλλαγή προτεραιότητας όταν αλλάζει πίνακα
- Μειονεκτήματα κλασικού χρονοπρογραμματιστή
  - Η αλληλεπιδραστικότητα προσδιορίζεται ευρετικά
  - Μέτρια απόδοση στις διεργασίες αλληλεπίδρασης

Χρονοπρογραμματισμός (6 από 8)

- Τελείως δίκαιος χρονοπρογραμματιστής (CFS)
  - Οργάνωση διεργασιών σε δένδρο κόκκινο-μαύρο
  - Κάθε διεργασία είναι εσωτερικός κόμβος
  - Ταξινόμηση με βάση το vruntime
    - Χρόνος εκτέλεσης στην ΚΜΕ
  - Εκτελείται πάντα ο αριστερότερος κόμβος
  - Περιοδικά αυξάνεται ο χρόνος του κόμβου
    - Σε κάποια στιγμή αναδιατάσσεται το δένδρο

Χρονοπρογραμματισμός (7 από 8)



- Το vruntime τρέχει ανάλογα με προτεραιότητα
  - Όσο χαμηλότερη, τόσο πιο γρήγορα αυξάνεται
  - Δεν χρειάζεται να έχουμε πολλές ουρές

Χρονοπρογραμματισμός (8 από 8)

- Χρονοπρογραμματισμός συγγένειας (affinity)
  - Προσπαθούμε η εργασία να μην αλλάζει επεξεργαστή
    - Αυτός είναι ο λόγος που έχουμε πολλές ουρές εκτέλεσης
    - Περιοδικά κάνουμε και εξισορρόπηση φόρτου
- Ουρές αναμονής (wait queues)
  - Μία ανά συμβάν όπου αναμένουν εργασίες
  - Εκεί βρίσκονται οι μπλοκαρισμένες εργασίες
  - Χρήση spinlock για παράλληλη πρόσβαση
    - Από πυρήνα και κώδικα διακοπών

Εκκίνηση (1 από 6)

- Εκκίνηση του πυρήνα
  - Το BIOS εκτελεί το διαγνωστικό εκκίνησης (POST)
  - Εντοπίζονται και διευθετούνται οι βασικές συσκευές
  - Διαβάζεται η κύρια εγγραφή εκκίνησης (MBR)
  - Η MBR διαβάζει το boot από συσκευή εκκίνησης
    - Παράδειγμα: GRUB (grand unified bootloader)
  - Το boot διαβάζει τον πυρήνα
  - Τέλος μεταβιβάζει τον έλεγχο στον πυρήνα

Εκκίνηση (2 από 6)

- Πρώτο μέρος εκκίνησης (σε assembly)
  - Εξαρτάται πολύ από τον επεξεργαστή
  - Δημιουργία στοίβας πυρήνα
  - Αναγνώριση επεξεργαστή
  - Υπολογισμός μεγέθους μνήμης
  - Ενεργοποίηση MMU
  - Απενεργοποίηση διακοπών
  - Κλήση της συνάρτησης main (στη C)

Εκκίνηση (3 από 6)

- Δεύτερο μέρος εκκίνησης (σε C)
  - Δέσμευση χώρου για μηνύματα εκκίνησης
  - Δέσμευση χώρου για δομές πυρήνα
  - Διερεύνηση και διευθέτηση συσκευών
  - Στατική φόρτωση οδηγών
    - Σε μεγάλες και ασφαλείς εγκαταστάσεις
  - Δυναμική φόρτωση οδηγών
    - Σε απλούστερες εγκαταστάσεις
  - Δημιουργία διεργασίας 0 και εκτέλεσή της

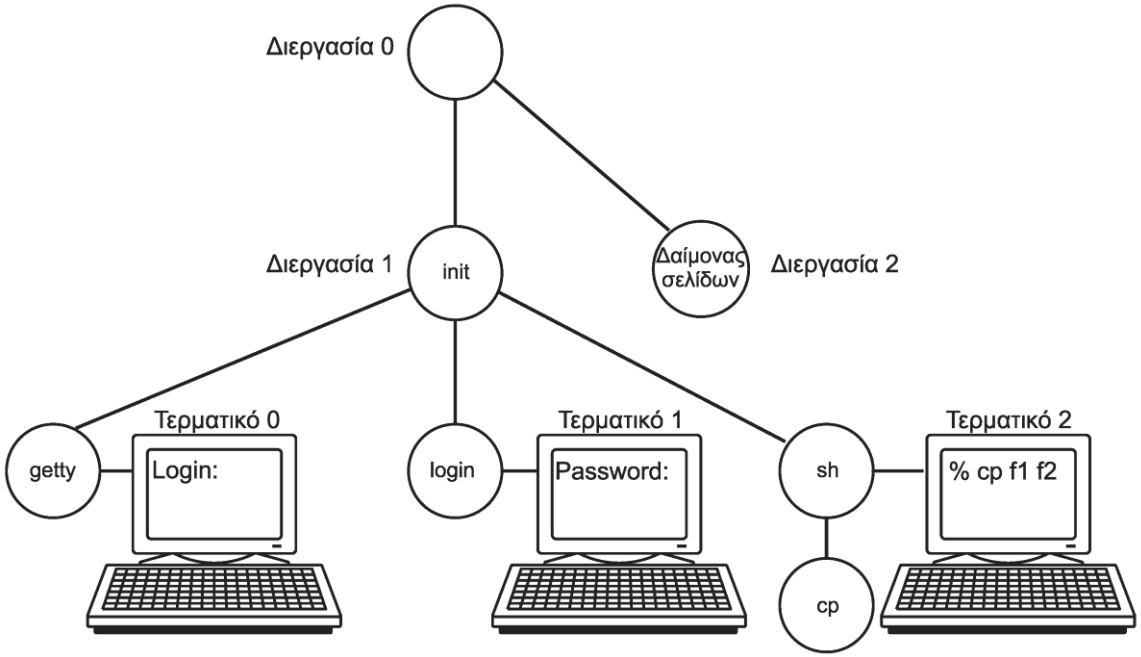
Εκκίνηση (4 από 6)

- Διεργασία 0
  - Ρύθμιση ρολογιού πραγματικού χρόνου
  - Ανάρτηση βασικού συστήματος αρχείων
  - Δημιουργία διεργασίας 1 (init) και 2 (page daemon)
- Διεργασία 1 (init)
  - Μονοχρηστικό: θυγατρική για κέλυφος
  - Πολυχρηστικό: θυγατρική για διάρθρωση
    - Εκτελεί το σενάριο /etc/rc για αρχικοποίηση
    - Ξεκινάει δαίμονες, αναρτά συστήματα αρχείων, κλπ

Εκκίνηση (5 από 6)

- Διεργασία 1 (init)
  - Διαβάζεται το /etc/ttys (στοιχεία τερματικών)
  - Για κάθε τερματικό εκτελείται η getty
  - Η getty ρυθμίζει το τερματικό και εμφανίζει login:
  - Όταν δώσουμε όνομα χρήστη εκτελείται η /bin/login
  - Η login διαβάζει και ελέγχει τον κωδικό πρόσβασης
  - Αν είναι σωστός, εκτελείται το κέλυφος του χρήστη
  - Το κέλυφος κάνει τη δική του αρχικοποίηση
  - Τελικά εμφανίζει το σύμβολο προτροπής στο χρήστη

Εκκίνηση (6 από 6)

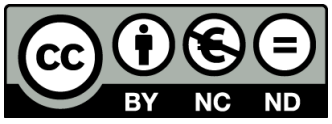


- Παράδειγμα δένδρου διεργασιών με 3 τερματικά
  - Οι getty/login/sh είναι παιδιά της init
    - Από εκεί ξεκινά η ομάδα διεργασιών ενός τερματικού

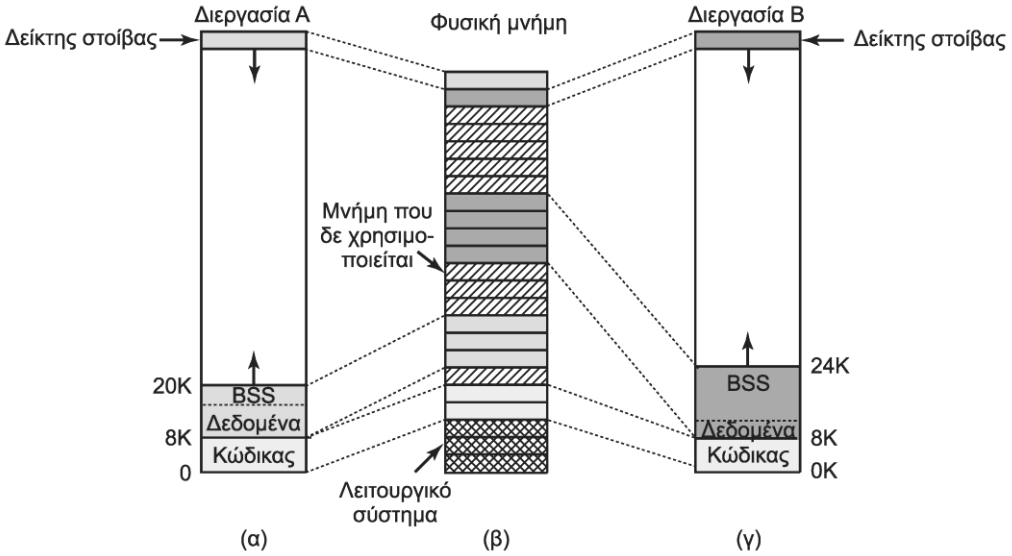
Διαχείριση μνήμης στο Linux

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 8:** Το ΛΣ Linux

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Έννοιες μνήμης (1 από 5)



- Ο χώρος διευθύνσεων χωρίζεται σε τρία τμήματα
  - Τμήμα κώδικα (text): εκτελέσιμος κώδικας
    - Σταθερό μέγεθος, δεν τροποποιείται
  - Τμήμα δεδομένων (data): δεδομένα προγράμματος
  - Τμήμα στοίβας (stack): στοίβα επιπέδου χρήστη

Έννοιες μνήμης (2 από 5)

- Το τμήμα δεδομένων χωρίζεται σε δύο μέρη
- Αρχικοποιημένα: μεταβλητές / σταθερές με τιμές
  - Αποθηκεύονται στο δίσκο οι αρχικές τιμές
- Μη αρχικοποιημένα (BSS): μηδενικές τιμές
  - Σημειώνεται ο χώρος που καταλαμβάνουν
  - Κατά τη φόρτωση παραχωρείται χώρος
    - Αρχικά εκχωρείται μία δεσμευμένη σελίδα
    - Η σελίδα είναι μηδενισμένη αλλά προστατευμένη
    - Στην πρώτη εγγραφή εκχωρείται κανονική σελίδα

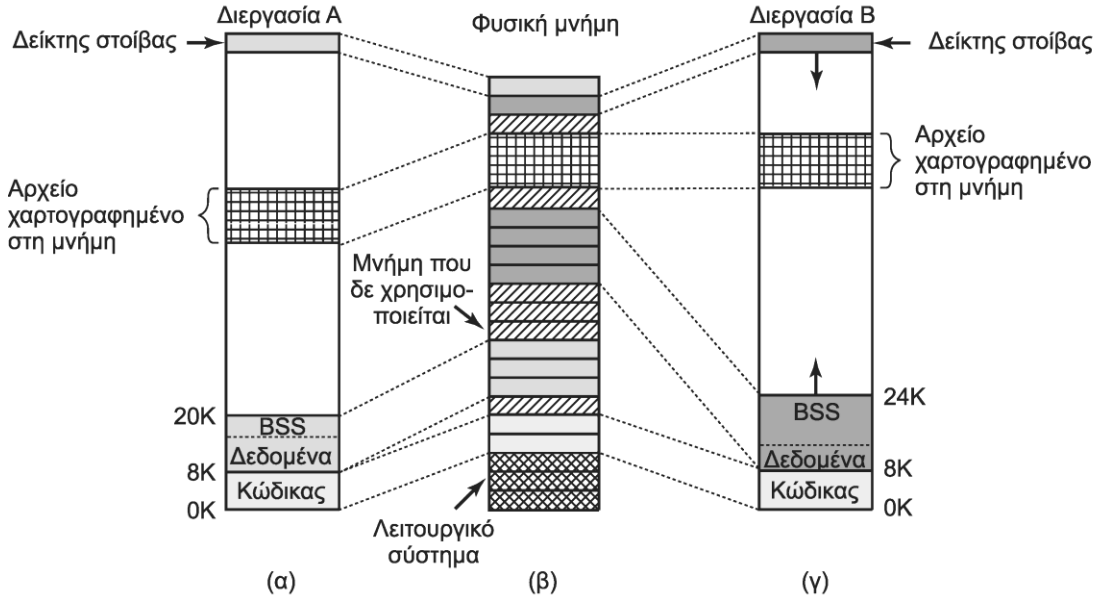
Έννοιες μνήμης (3 από 5)

- Δυναμική εκχώρηση μνήμης
  - Το τμήμα δεδομένων μεγαλώνει με την brk
    - Η malloc την χρησιμοποιεί για δυναμικές δομές
  - Η δυναμική μνήμη ονομάζεται σωρός (heap)
- Το τμήμα στοίβας επίσης αλλάζει μέγεθος
  - Συνήθως αρχίζει από το άνω άκρο
  - Όταν γεμίσει η τρέχουσα σελίδα κατανέμεται άλλη
  - Η στοίβα αρχικά περιέχει παραμέτρους κλήσης
    - Μεταβλητές περιβάλλοντος και ορίσματα γραμμής εντολών

Έννοιες μνήμης (4 από 5)

- Κοινόχρηστα τμήματα κώδικα
  - Πολλές διεργασίες με τον ίδιο κώδικα
    - Ένα τμήμα που απεικονίζεται στις ίδιες σελίδες μνήμης
  - Τα δεδομένα και η στοίβα δεν είναι κοινόχρηστα
    - Μετά τη fork φαίνεται ότι έχουμε κοινές σελίδες
    - Όταν τροποποιηθούν αντιγράφονται σε χωριστές σελίδες
- Χωριστοί χώροι κώδικα / δεδομένων
  - Υποστηρίζονται σε μηχανές με τέτοια διάκριση
  - Οι προσπελάσεις πάνε στον κατάλληλο χώρο

Έννοιες μνήμης (5 από 5)



- Αρχεία χαρτογραφημένα στη μνήμη
  - Πολλές διεργασίες βλέπουν ταυτόχρονα το αρχείο
  - Κάθε διεργασία μπορεί να το έχει αλλού
  - Πολύ γρήγορη μέθοδος ανταλλαγής δεδομένων

Κλήσεις μνήμης

Κλήση συστήματος	Περιγραφή
s = brk(addr)	Αλλαγή μεγέθους τμήματος δεδομένων
a = mmap(addr, len, prot, flags, fd, offset)	Χαρτογράφηση αρχείου στη μνήμη
s = munmap(addr, len)	Αποχαρτογράφηση ενός αρχείου

- Το POSIX δεν ορίζει κλήσεις διαχείρισης μνήμης
  - Προτείνει τη χρήση των συναρτήσεων της C (malloc)
- brk: ορίζει πού σταματάει το τμήμα δεδομένων
  - Χρήση και για αύξηση και για μείωση του μεγέθους του
- mmap: χαρτογράφηση αρχείου σε θέση μνήμης
  - Χαρτογράφηση σε αρχή σελίδας, ακέραιο πλήθος σελίδων
  - Μπορούμε να χαρτογραφήσουμε μέρος του
- unmap: κατάργηση χαρτογράφησης αρχείου
  - Μπορεί να αφορά μέρος μόνο του αρχείου

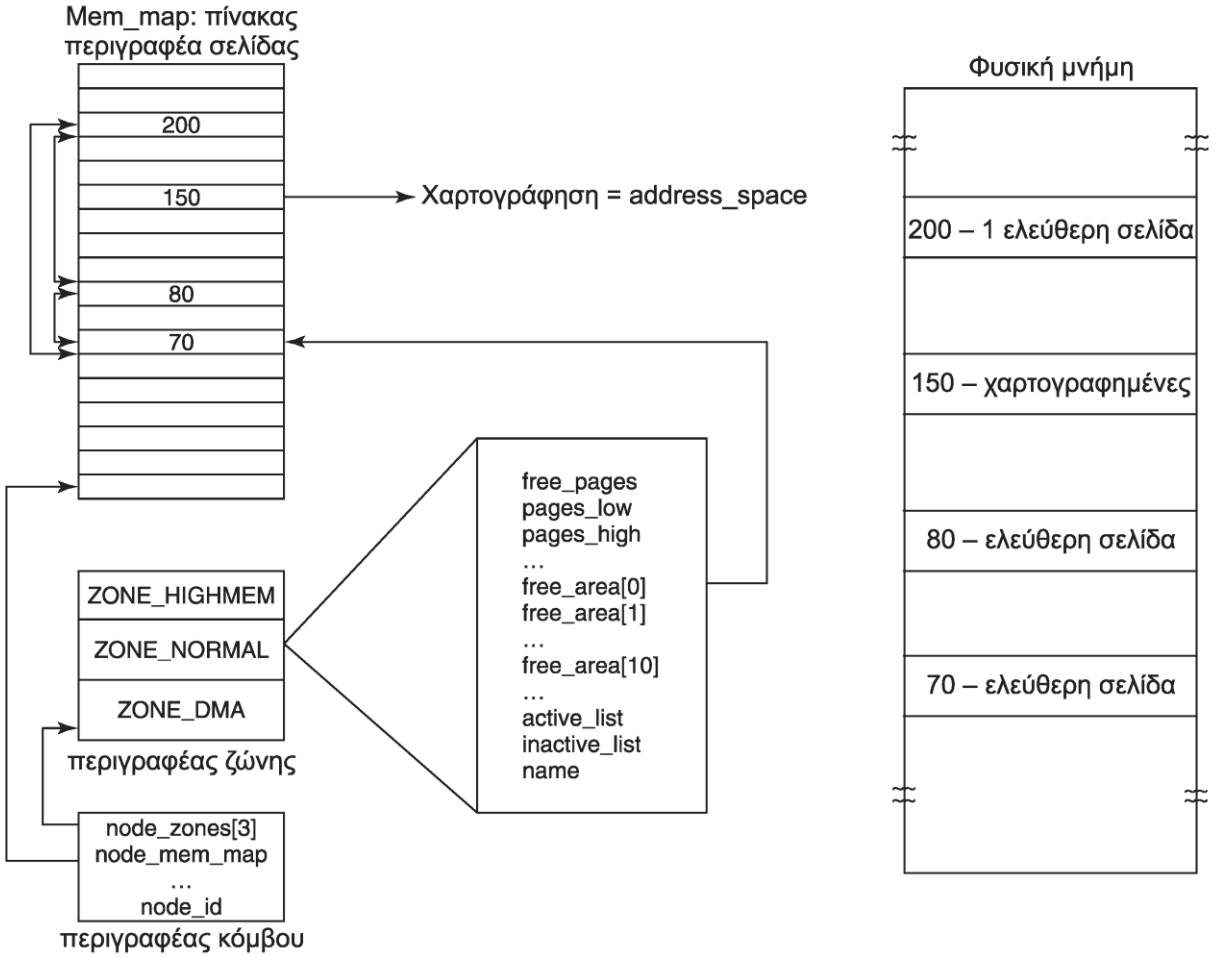
Υλοποίηση μνήμης (1 από 10)

- Στις μηχανές 32 bit ο χώρος διευθύνσεων είναι 3 GB
  - Το άνω 1 GB περιέχει πυρήνα και πίνακες σελίδων
  - Ο χώρος διευθύνσεων δημιουργείται με την clone
    - Αντικαθίσταται όταν καλείται η exec
- Το Linux διακρίνει τρεις ζώνες στη φυσική μνήμη
  - ZONE_DMA: σελίδες κατάλληλες για DMA
  - ZONE_NORMAL: κανονικές σελίδες
  - ZONE_HIGHMEM: σελίδες στην υψηλή μνήμη
    - Δεν υπάρχει σε μηχανές 64 bit, υπάρχει ZONE_DMA32
  - Ανεξάρτητη διαχείριση κάθε ζώνης της μνήμης

Υλοποίηση μνήμης (2 από 10)

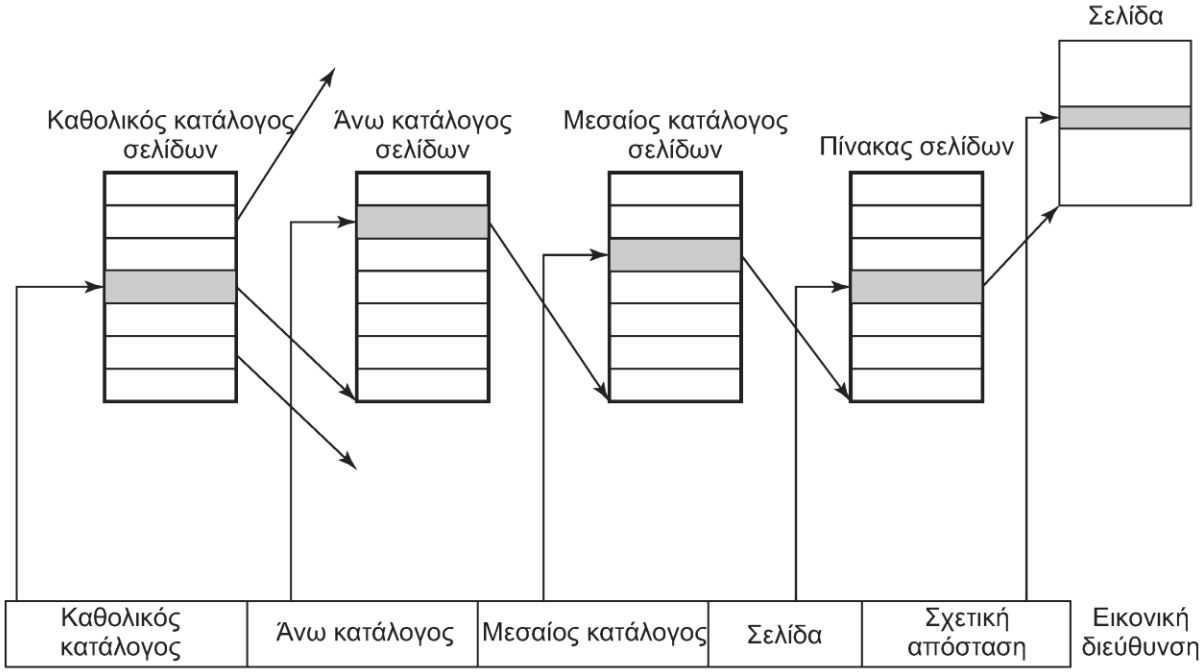
- Πυρήνας και χάρτης μνήμης καρφιτσώνονται (pinned)
- Πίνακας περιγραφέων σελίδων (συστήματος)
  - Κάθε καταχώρηση περιγράφει μία φυσική σελίδα
    - Δείκτες για διπλή σύνδεση ελεύθερων πλαισίων
    - Δείκτης σε χώρο διευθύνσεων για κατειλημμένες σελίδες
- Για κάθε ζώνη έχουμε ένα περιγραφέα ζώνης
  - Πλήθος ενεργών και ελεύθερων σελίδων
  - Υψηλό και χαμηλό υδατογράφημα
  - Δείκτες προς λίστες ελεύθερων πλαισίων
- Περιγραφέας κόμβου: περιγράφει μνήμη ενός κόμβου

Υλοποίηση μνήμης (3 από 10)



Δομές διαχείρισης μνήμης

Υλοποίηση μνήμης (4 από 10)

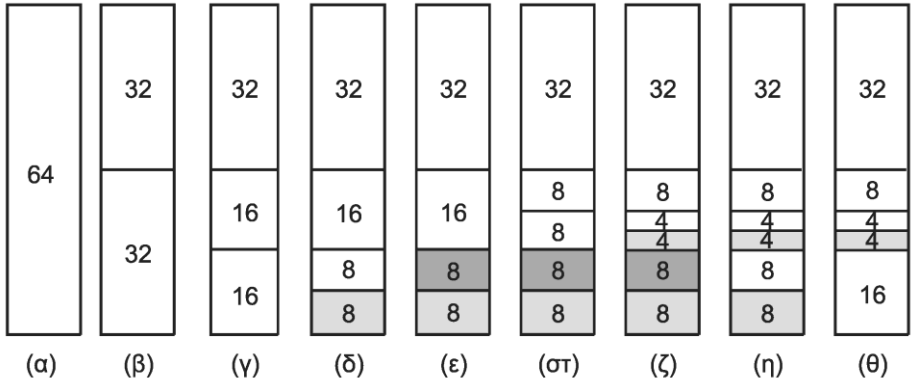


- Μηχανισμός σελιδοποίησης τεσσάρων επιπέδων
  - Η διεύθυνση χωρίζεται σε πέντε πεδία
  - Μπορεί να χρησιμοποιηθεί και με λιγότερα επίπεδα
    - Χρήση πινάκων με ένα μόνο στοιχείο

Υλοποίηση μνήμης (5 από 10)

- Η περισσότερη μνήμη κατανέμεται δυναμικά
  - Καρφιτσωμένες σελίδες (πυρήνας)
  - Σελίδες χρηστών (ενεργές)
  - Κρυφή μνήμη μπλοκ: πρόσφατα μπλοκ αρχείων
  - Κρυφή μνήμη σελιδοποίησης: υποψήφιες για αφαίρεση
  - Μία ενιαία κρυφή μνήμη για όλες τις σελίδες
- Δυναμική φόρτωση υπομονάδων
  - Συνήθως οδηγοί συσκευών
  - Απαιτούν συνεχόμενη μνήμη στον πυρήνα

Υλοποίηση μνήμης (6 από 10)



- Χρήση του συστήματος φίλων (buddy system)
  - Οι ελεύθερες περιοχές οργανώνονται σε λίστες
  - Κάθε λίστα έχει περιοχές με 1, 2, 4, … πλαίσια
- Κάθε αίτηση στρογγυλεύεται σε δύναμη του 2
  - Δεσμεύεται μία περιοχή από την αντίστοιχη λίστα
  - Αν δεν υπάρχει, σπάει μία μεγαλύτερη περιοχή
  - Το ελεύθερο κομμάτι πάει στην κατάλληλη λίστα

Υλοποίηση μνήμης (7 από 10)

- Το σύστημα φίλων οδηγεί σε εσωτερική κατάτμηση
  - Ζητάμε 65 σελίδες αλλά δεσμεύουμε 128!
- Ο κατανεμητής πλακιδίων κατανέμει τα υπόλοιπα
  - Χρησιμοποιεί τα υπόλοιπα των κατανομών μνήμης
  - Ο πυρήνας χρησιμοποιεί κρυφές μνήμες αντικειμένων
    - Δείκτες προς πλακίδια με όμοια αντικείμενα
    - Το πλακίδιο μπορεί να είναι άδειο, μισογεμάτο ή γεμάτο
    - Ψάχνουμε αρχικά για μισογεμάτο πλακίδιο
    - Αν δεν υπάρχει χρησιμοποιούμε άδειο πλακίδιο
- Κατανομή συνεχόμενης εικονικής μνήμης με vmalloc

Υλοποίηση μνήμης (8 από 10)

- Ο χώρος διευθύνσεων χωρίζεται σε περιοχές
  - Συνεχόμενες σελίδες με την ίδια προστασία
  - Αναφορές μεταξύ περιοχών δίνουν μοιραίο σφάλμα
  - Οι σελίδες ήταν 4 KB στον Pentium, τώρα έχει και 4 MB
    - Με επέκταση φυσικής διεύθυνσης (PAE) είναι 2 MB
- Κάθε περιοχή περιγράφεται με δομή vm_area_struct
  - Bit προστασίας, καρφιτσωμένη, προς τα πού μεγαλώνει
  - Οι περιοχές μίας διεργασίας συνδέονται μεταξύ τους

Υλοποίηση μνήμης (9 από 10)

- Υλοποίηση αντιγραφής με την εγγραφή (copy on write)
  - Μετά το fork οι διεργασίες δείχνουν στις ίδιες σελίδες
  - Οι σελίδες σημειώνονται μόνο για ανάγνωση
  - Οι περιοχές σημειώνονται για ανάγνωση/εγγραφή
  - Σε απόπειρα εγγραφής ο πυρήνας αντιγράφει τη σελίδα
- Η vm_area_struct δείχνει και στο χώρο στο δίσκο
  - Ο εκτελέσιμος κώδικας δείχνει στο εκτελέσιμο αρχείο
  - Τα χαρτογραφημένα αρχεία δείχνουν στο αρχείο
  - Στοίβα και σωρός δείχνουν στην περιοχή ανταλλαγής

Υλοποίηση μνήμης (10 από 10)

- Για κάθε χώρο διευθύνσεων μία δομή mm_struct
  - Πληροφορίες για τα τμήματα εικονικής μνήμης
  - Χρήστες που μοιράζονται το χώρο διευθύνσεων
  - Δείκτες προς τις δομές vm_area_struct
  - Οι δομές οργανώνονται σε διπλά συνδεδεμένη λίστα
    - Κατάλληλη για σάρωση όλου του χώρου διευθύνσεων
  - Επιπλέον οργανώνονται σε δένδρο κόκκινο-μαύρο
    - Κατάλληλη για άμεσο εντοπισμό περιοχής μνήμης
  - Κάθε λειτουργία χρησιμοποιεί την κατάλληλη μέθοδο

Σελιδοποίηση (1 από 6)

- Διαχείριση μνήμης μόνο σε επίπεδο σελίδων
  - Οι σελίδες προσκομίζονται όταν χρειάζονται
  - Δεν γίνεται προκαταβολική προσκόμιση σελίδων
  - Αρκεί η δομή χρήστη και ο πίνακας σελίδων στη μνήμη
- Υλοποίηση: πυρήνας και δαίμονας σελιδοποίησης
  - Η διεργασία 2 είναι ο page daemon
- Εναλλαγή σελίδων στο δίσκο
  - Κώδικας και χαρτογραφημένα αρχεία: αντίστοιχα αρχεία
  - Δεδομένα: περιοχή εναλλαγής (swap area)
    - Διαμέρισμα δίσκου ή/και αρχεία σταθερού μήκους

Σελιδοποίηση (2 από 6)

- Το διαμέρισμα δίσκου είναι πιο αποδοτικό
  - Δεν χρειάζεται μετάφραση διευθύνσεων
  - Χρησιμοποιεί οποιοδήποτε μέγεθος μπλοκ
  - Οι σελίδες είναι συνεχόμενες στο δίσκο
  - Έχει υψηλότερη προτεραιότητα από τα αρχεία
- Κατανομή σελίδων στην περιοχή εναλλαγής
  - Χρήση χάρτη bit για τις ελεύθερες σελίδες
  - Όποτε χρειάζεται εναλλαγή, δεσμεύεται σελίδα
  - Η σελίδα γράφεται στο δίσκο
  - Ενημερώνεται ο πίνακας σελίδων

Σελιδοποίηση (3 από 6)

- Αλγόριθμος επαναδιεκδίκησης πλαισίων (PFRA)
  - Προσπαθεί να διατηρεί μερικές σελίδες ελεύθερες
    - Όταν χρειαστούν σελίδες θα είναι άμεσα διαθέσιμες
  - Καλείται περιοδικά για να απελευθερώσει σελίδες
- Οι σελίδες διακρίνονται σε τέσσερις κατηγορίες
  - Μη διεκδικήσιμες: δεν αφαιρούνται από τη μνήμη
    - Δεσμευμένες και κλειδωμένες, μέρη του πυρήνα
  - Εναλλάξιμες: πρέπει να γραφτούν σε περιοχή εναλλαγής
  - Συγχρονίσιμες: πρέπει να γραφτούν αν είναι λερωμένες
  - Απορρίψιμες: μπορούν να διατεθούν αμέσως

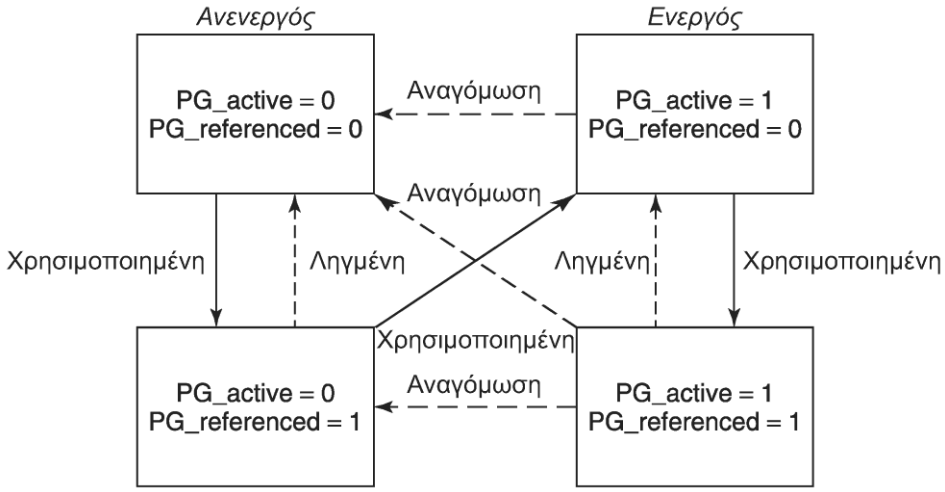
Σελιδοποίηση (4 από 6)

- Η init ξεκινάει μία kswapd ανά κόμβο μνήμης
  - Ξυπνάει περιοδικά ή όταν έχουμε λίγες σελίδες
  - Ο έλεγχος μνήμης γίνεται ανά ζώνη
  - Αν υπάρχει διαθέσιμη μνήμη μπλοκάρεται ξανά
  - Αν η μνήμη πέσει κάτω από όριο, αρχίζει ο PFRA
  - Ο PFRA προσπαθεί να ελευθερώσει 32 σελίδες
    - Το πλήθος μπορεί να ρυθμιστεί
  - Προσπάθεια να αποφύγουμε την υπερβολική Ε/Ε

Σελιδοποίηση (5 από 6)

- Ο PFRA ξεκινάει με τις πιο εύκολες σελίδες
  - Απορρίψιμες, χωρίς αναφορά
  - Με εφεδρική αποθήκευση χωρίς πρόσφατες αναφορές
  - Κοινόχρηστες σελίδες χωρίς πρόσφατες αναφορές
    - Αν ελευθερωθούν αλλάζουν πολλοί πίνακες σελίδων
  - Σελίδες που πρέπει να γραφτούν σε περιοχή εναλλαγής
  - Οι άκυρες, απούσες, κλειδωμένες παραλείπονται
- Ο PFRA χρησιμοποιεί έναν αλγόριθμο τύπου ρολογιού
  - Προσπαθεί ανάλογα με την κατάσταση επείγοντος
  - Οι σελίδες είναι σε ενεργές και ανενεργές λίστες

Σελιδοποίηση (6 από 6)

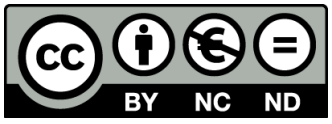


- Ενεργός/ανενεργός και αναφορά/όχι αναφορά
  - Οι ανενεργές και μη αναφερθείσες είναι οι καλύτερες
  - Μπορούν να επιλεγούν και άλλες σελίδες (αναγόμωση)
- Ο pdflush γράφει λερωμένες σελίδες στο δίσκο
  - Πολλά νήματα για επικάλυψη των εγγραφών στο δίσκο
  - Εκτελείται περιοδικά ή όταν υπάρχει έλλειψη μνήμης

Είσοδος / έξοδος στο Linux

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 8:** Το ΛΣ Linux

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



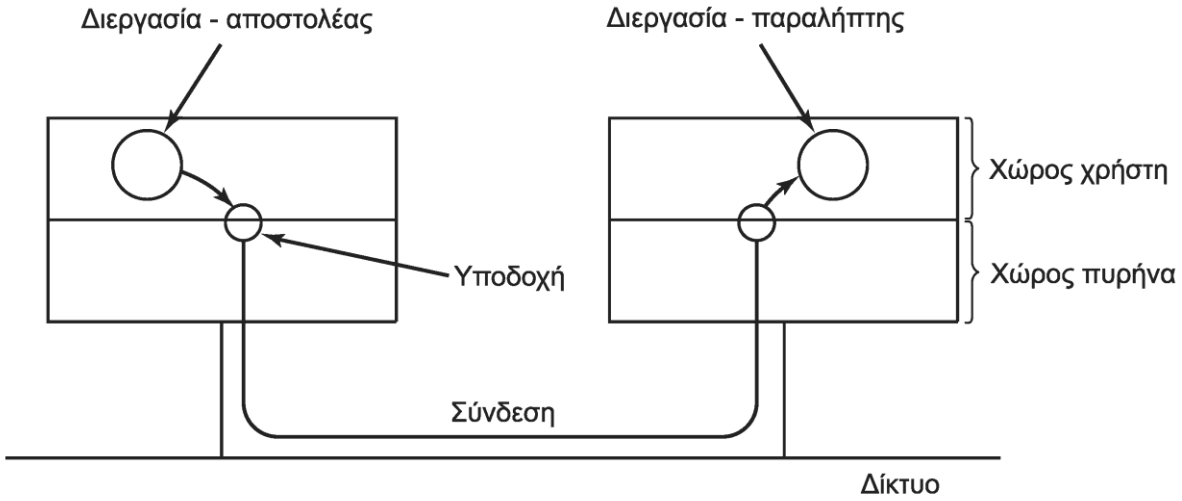
Έννοιες Ε/Ε (1 από 2)

- Οι συσκευές αντιστοιχούν σε ειδικά αρχεία
  - Συνήθως βρίσκονται στον κατάλογο /dev
    - Παράδειγμα: /dev/sda (δίσκος a), /dev/lp (εκτυπωτής)
  - Χρήση ίδιων κλήσεων και εντολών με τα αρχεία
    - Παράδειγμα: cp file /dev/lp (εκτύπωση file)
- Κάθε συσκευή χαρακτηρίζεται από δύο αριθμός
  - Μείζων (major): επιλέγει οδηγό συσκευής
  - Ελάσσων (minor): επιλέγει συσκευή του οδηγού
  - Οδηγοί για πολλαπλές συσκευές (π.χ. τερματικό)

Έννοιες Ε/Ε (2 από 2)

- Ειδικά αρχεία μπλοκ (block special files)
  - Αποτελούνται από ακολουθία αριθμημένων μπλοκ
  - Διαβάζουμε/γράφουμε κάθε μπλοκ ξεχωριστά
- Ειδικά αρχεία χαρακτήρων (character special files)
  - Παράγουν ή καταναλώνουν ρεύματα χαρακτήρων
  - Δεν μπορούμε να αναζητήσουμε τυχαίες θέσεις
- Έλεγχος ειδικών αρχείων χαρακτήρων
  - Το τερματικό αναγνωρίζει ειδικούς χαρακτήρες
  - Ο χρήστης μπορεί να επιλέξει τους δικούς του
    - Χρήση ειδικής κλήσης συστήματος

Δικτύωση (1 από 3)



- Βασική αφαίρεση: υποδοχή επικοινωνίας
  - Προέρχεται από το Berkeley UNIX
  - Η υποδοχή είναι ένα άκρο επικοινωνίας
  - Οι διεργασίες επικοινωνούν μέσω υποδοχών
  - Η δημιουργία τους επιστρέφει περιγραφέα αρχείου

Δικτύωση (2 από 3)

- Αξιόπιστο συνδεσμοστρεφές ρεύμα byte
  - Παρόμοιο με αγωγούς αλλά μεταξύ μηχανών
  - Συνήθως βασίζεται στο TCP
- Αξιόπιστο συνδεσμοστρεφές ρεύμα πακέτων
  - Παρόμοια με ρεύμα byte αλλά με όρια πακέτων
  - Μπορεί να υλοποιηθεί με το SCTP
- Αναξιόπιστη μετάδοση πακέτων
  - Απευθείας μετάδοση πακέτων στο δίκτυο
  - Η εφαρμογή ασχολείται με την αξιοπιστία
  - Συνήθως βασίζεται σε UDP

Δικτύωση (3 από 3)

- Κάθε υποδοχή πρέπει να έχει μία διεύθυνση
  - Συνήθως: τομέας διευθύνσεων Διαδικτύου
  - Αριθμοί 32 bit για τα άκρα (διευθύνσεις IP)
  - Αριθμοί 16 bit για την εφαρμογή (θύρες TCP ή UDP)
  - Απόδοση διεύθυνση με την κλήση bind
  - Αναμονή για αίτηση με listen
  - Αίτηση σύνδεσης με connect
  - Αποδοχή αίτησης με accept
  - Τερματισμός σύνδεσης με close

Κλήσεις Ε/Ε

Κλήση συνάρτησης	Περιγραφή
s = cfsetospeed(&termios, speed)	Καθορισμός ταχύτητας εξόδου
s = cfsetispeed(&termios, speed)	Καθορισμός ταχύτητας εισόδου
s = cfgetospeed(&termios, speed)	Λήψη ταχύτητας εξόδου
s = cfgetispeed(&termios, speed)	Λήψη ταχύτητας εισόδου
s = tcsetattr(fd, opt, &termios)	Καθορισμός χαρακτηριστικών
s = tcgetattr(fd, &termios)	Λήψη χαρακτηριστικών

- Ίδιες κλήσεις με αυτές των αρχείων
- Επιπλέον κλήσεις ελέγχου για ειδικές λειτουργίες
  - Παλιότερα κάθε συσκευή δεχόταν κλήσεις ioctl
  - Οι κλήσεις αυτές διέφεραν από σύστημα σε σύστημα
  - Στο POSIX ορίστηκαν γενικές κλήσεις βιβλιοθήκης
    - Ορισμός/ανάγνωση ταχύτητας τερματικού
    - Ορισμός/ανάγνωση παραμέτρων τερματικών

Υλοποίηση Ε/Ε (1 από 5)

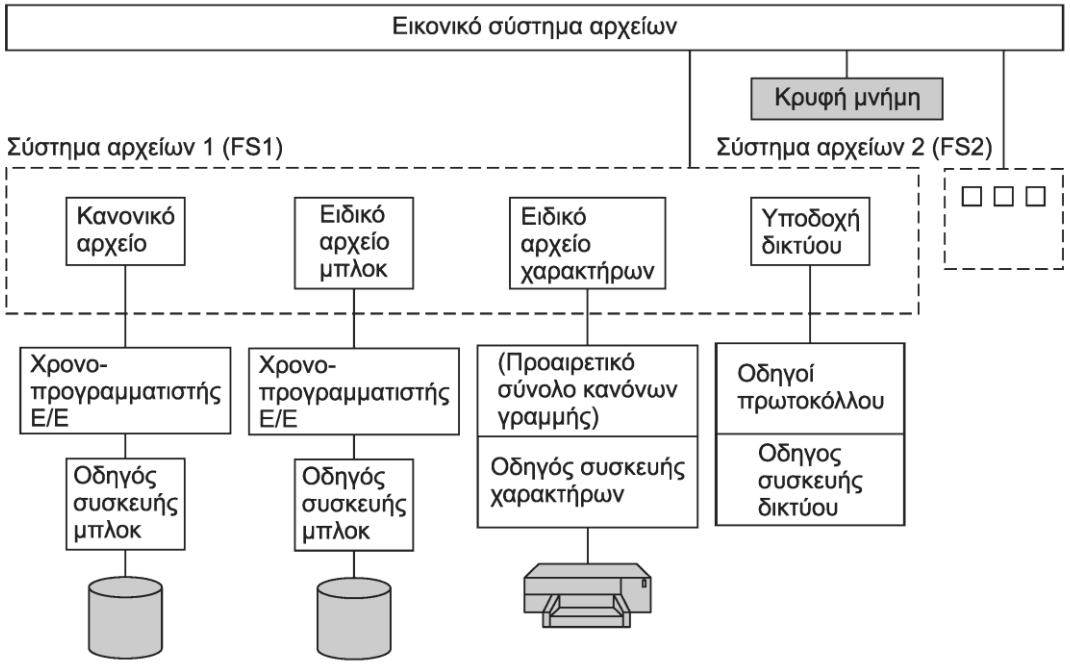
Συσκευή	Open	Close	Read	Write	Ioctl	Άλλη
Null	null	null	null	null	null	...
Μνήμη	null	null	mem_read	mem_write	null	...
Πληκτρολόγιο	k_open	k_close	k_read	error	k_ioctl	...
Τερματικό (tty)	tty_open	tty_close	tty_read	tty_write	tty_ioctl	...
Εκτυπωτής	lp_open	lp_close	error	lp_write	lp_ioctl	...

- Προσπέλαση συσκευών μέσω ειδικών αρχείων
  - Διαβάζονται ο μείζων και ο ελάσσων αριθμός συσκευής
  - Διακρίνουμε αν η συσκευή είναι μπλοκ ή χαρακτήρων
  - Ο μείζων αριθμός δείχνει (κατακερματισμός) σε μία δομή
    - Η δομή δείχνει στις μεθόδους ανοίγματος, ανάγνωσης, εγγραφής
    - Οι διαδικασίες καταχωρούνται με την φόρτωση του οδηγού
  - Ο ελάσσων αριθμός χρησιμοποιείται ως παράμετρος

Υλοποίηση Ε/Ε (2 από 5)

- Οι οδηγοί αποτελούνται από δύο τμήματα
  - Και τα δύο εκτελούνται σε κατάσταση πυρήνα
  - Το άνω μέρος εκτελείται για λογαριασμό του καλούντα
  - Το κάτω μέρος αλληλεπιδρά με τη συσκευή
- Διασύνδεση οδηγού-πυρήνα
  - Ο πυρήνας διαθέτει σύνολο κλήσεων στις συσκευές
  - Κατανομή μνήμης, έλεγχος DMA, χρονόμετρα κλπ
- Υποσύστημα χειρισμού ειδικών αρχείων μπλοκ
  - Αξιοποιεί κρυφή μνήμη (cache) για μπλοκ και σελίδες
  - Πάνω τους βρίσκεται το γενικό (generic) επίπεδο μπλοκ

Υλοποίηση Ε/Ε (3 από 5)



- Υποσύστημα χειρισμού ειδικών αρχείων μπλοκ
  - Πρώτα ελέγχεται αν το μπλοκ είναι στην κρυφή μνήμη
  - Αλλιώς διαβάζεται και προστίθεται στην κρυφή μνήμη
    - Αντικατάσταση μπλοκ με τον αλγόριθμο PFRA

Υλοποίηση Ε/Ε (4 από 5)

- Υποσύστημα χειρισμού ειδικών αρχείων μπλοκ
  - Και οι εγγραφές αποθηκεύονται στην κρυφή μνήμη
  - Ο δαίμονας pdflush γράφει μπλοκ περιοδικά στο δίσκο
  - Χρονοπρογραμματιστής ελαχιστοποίησης αναζητήσεων
    - Στη βασική του μορφή είναι αλγόριθμος ανελκυστήρα
    - Αρχικά ταξινομούνται οι αιτήσεις με τη σειρά
    - Μετά συγχωνεύονται οι γειτονικές αιτήσεις
  - Χρήση δύο πρόσθετων λιστών για αποφυγή λιμοκτονίας
    - Αναγνώσεις και εγγραφές ταξινομημένες κατά προθεσμία
    - Μετά από κάποιο όριο καθυστέρησης, εξυπηρετούνται πρώτες

Υλοποίηση Ε/Ε (5 από 5)

- Ανεπεξέργαστα αρχεία μπλοκ (raw block files)
  - Άμεση προσπέλαση χωρίς σύστημα αρχείων
  - Χρησιμοποιούνται για σελιδοποίηση και συντήρηση
- Υποσύστημα χειρισμού ειδικών αρχείων χαρακτήρων
  - Δεν χρησιμοποιείται κρυφή μνήμη (δεν έχει νόημα)
  - Χρήση προαιρετικών κανόνων γραμμής (line disciplines)
- Υποσύστημα χειρισμού συσκευών δικτύου
  - Παρόμοιες (αλλά όχι ίδιες) με συσκευές χαρακτήρων
  - Παράγουν και καταναλώνουν πακέτα με κεφαλίδες
  - Χρήση δομών προσωρινής μνήμης υποδοχής (skbuff)

Υπομονάδες

- Στο UNIX οι οδηγοί συσκευών συνδέονται στατικά
- Το Linux κάνει (και) για προσωπικούς υπολογιστές
- Φορτώσιμες υπομονάδες (loadable modules)
  - Κώδικας που φορτώνεται κατά την εκτέλεση στον πυρήνα
    - Οδηγοί συσκευών, συστήματα αρχείων, πρωτόκολλα δικτύου
  - Κατά τη φόρτωση οι υπομονάδες επανατοποθετούνται
  - Ελέγχεται η διαθεσιμότητα των πόρων
  - Ενεργοποιούνται τα κατάλληλα διανύσματα διακοπών
  - Ενημερώνεται ο κατάλληλος πίνακας οδηγών
  - Τέλος, ο οδηγός προετοιμάζει τη συσκευή

Σύστημα αρχείων στο Linux

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 8:** Το ΛΣ Linux

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Έννοιες αρχείων (1 από 6)

- Στο Linux έχουμε πολλά συστήματα αρχείων
  - Αρχικά χρησιμοποιήθηκε το σύστημα του MINIX
    - Παρόμοιο με UNIX Version 7 (14 χαρακτήρες, 64 MB)
  - Το σύστημα ext επέτρεπε 255 χαρακτήρες, 2 GB
    - Ήταν όμως πιο αργό από το σύστημα του MINIX
  - Το σύστημα ext2 ήταν το πρώτο «καλό» σύστημα
    - Εξελίχθηκε σε ext3/ext4
  - Υποστηρίζονται και πολλά άλλα συστήματα
    - UFS, NFS, FAT, NTFS, ISO9660, …

Έννοιες αρχείων (2 από 6)

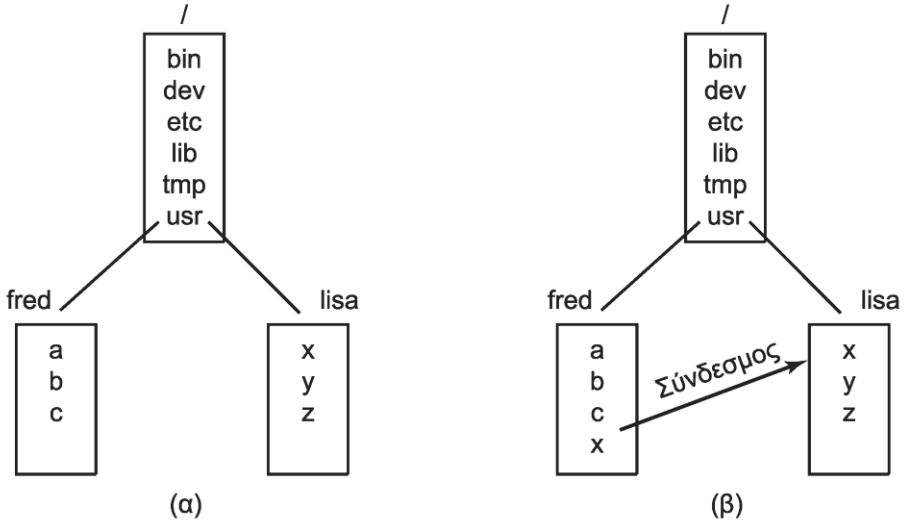
- Βασική δομή αρχείων στο Linux
  - Τα αρχεία είναι απλώς ακολουθίες από byte
  - Ονόματα έως 255 χαρακτήρες
    - Επιτρέπονται όλοι οι χαρακτήρες εκτός του NUL
    - Μπορεί να χρησιμοποιηθεί και η τελεία
  - Το σύστημα δεν αναγνωρίζει προεκτάσεις
    - Πολλά προγράμματα όμως τις αναμένουν
- Τα αρχεία ομαδοποιούνται σε καταλόγους
  - Και οι κατάλογοι αποθηκεύονται σαν αρχεία

Έννοιες αρχείων (3 από 6)

Κατάλογος	Περιεχόμενα
bin	Δυαδικά (εκτελέσιμα) προγράμματα
dev	Ειδικά αρχεία για συσκευές Ε/Ε
etc	Διάφορα αρχεία συστήματος
lib	Βιβλιοθήκες
usr	Κατάλογοι χρηστών

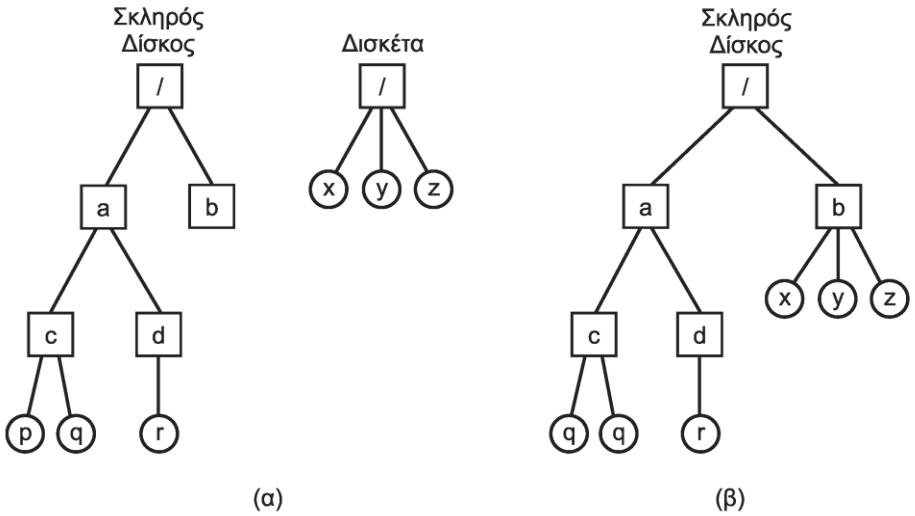
- Η ρίζα του συστήματος καταλόγων είναι το /
  - Το / χρησιμοποιείται και για διαχωρισμό καταλόγων
  - Οι βασικοί κατάλογοι είναι (σχετικά) τυποποιημένοι
  - Κάθε διεργασία έχει τρέχοντα κατάλογο εργασίας
  - Απόλυτες και σχετικές διαδρομές
  - Οι κατάλογοι περιέχουν ειδικές καταχωρίσεις . και ..

Έννοιες αρχείων (4 από 6)



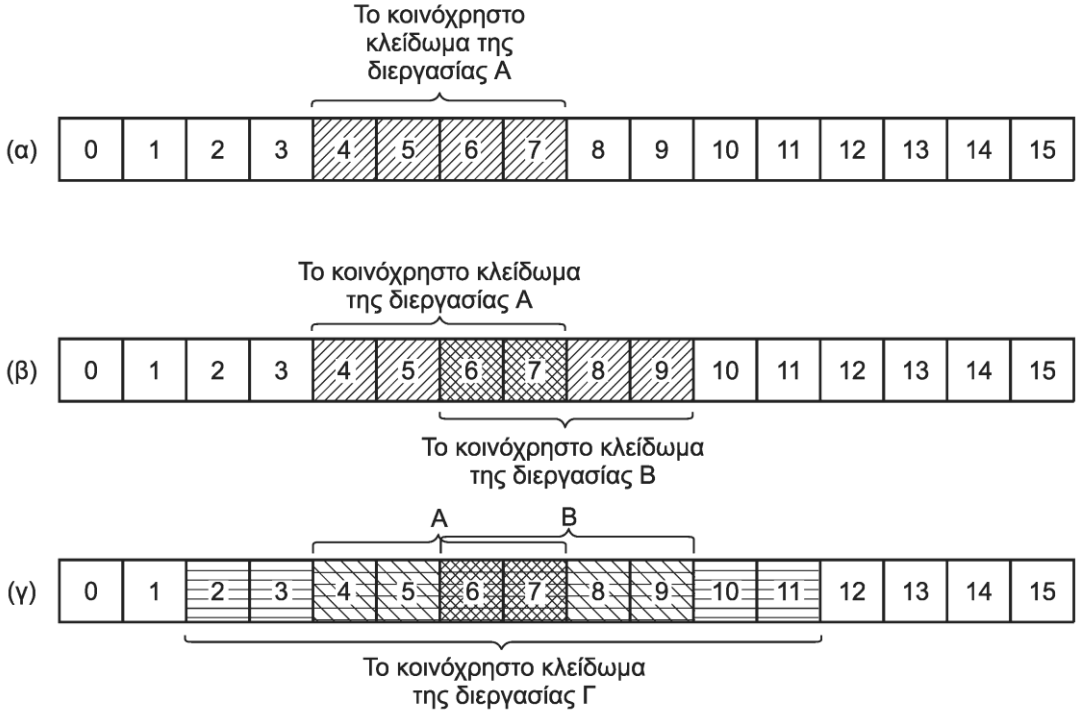
- Σύνδεσμοι (links) προς αρχεία/καταλόγους
  - Δείκτες προς άλλα αρχεία/καταλόγους
- Ενοποίηση συστημάτων αρχείων
  - Μπορεί να έχουμε πολλά διαμερίσματα και δίσκους
  - Σε κάθε διαμέρισμα υπάρχει χωριστό σύστημα αρχείων

Έννοιες αρχείων (5 από 6)



- Ενοποίηση συστημάτων αρχείων
  - Στο MS-DOS κάθε σύστημα είναι ανεξάρτητο
    - Παράδειγμα: a:\command.com ή c:\command.com
  - Στο Linux όλα τα συστήματα γίνονται ένα δένδρο
    - Η ρίζα του ενός αναρτάται σε έναν κατάλογο του άλλου

Έννοιες αρχείων (6 από 6)



- Κλείδωμα (locking)
  - Καθορίζεται το αρχείο, το πρώτο byte και το πλήθος byte
  - Κοινόχρηστα κλειδώματα και αποκλειστικά κλειδώματα

Κλήσεις αρχείων (1 από 5)

Κλήση συστήματος	Περιγραφή
fd = creat(name, mode)	Ένας τρόπος δημιουργίας ενός νέου αρχείου
fd = open(file, how, ...)	Ανοίγει ένα αρχείο για ανάγνωση, εγγραφή, ή και τα δύο
s = close(fd)	Κλείνει ένα ανοιχτό αρχείο
n = read(fd, buffer, nbytes)	Διαβάζει δεδομένα από ένα αρχείο και τα τοποθετεί σε προσωρινή μνήμη
n = write(fd, buffer, nbytes)	Διαβάζει δεδομένα από μια προσωρινή μνήμη και τα αποθηκεύει σε ένα αρχείο
position = lseek(fd, offset, whence)	Μετακινεί το δείκτη αρχείου
s = stat(name, &buf)	Διαβάζει τις πληροφορίες κατάστασης ενός αρχείου
s = fstat(fd, &buf)	Διαβάζει τις πληροφορίες κατάστασης ενός αρχείου
s = pipe(&fd[0])	Δημιουργεί έναν αγωγό (pipe)
s = fcntl(fd, cmd, ...)	Κλείδωμα αρχείου και άλλες λειτουργίες

- Κλήσεις που αφορούν μεμονωμένα αρχεία
  - Δημιουργία και άνοιγμα αρχείου για εγγραφή (creat)
  - Άνοιγμα αρχείου για ανάγνωση ή/και εγγραφή (open)
    - Επιστρέφουν περιγραφέα αρχείου για τις επόμενες κλήσεις
  - Οι περιγραφείς 0, 1 και 2 είναι ανοιχτοί από την αρχή

Κλήσεις αρχείων (2 από 5)

- Κλήσεις που αφορούν μεμονωμένα αρχεία
  - Κλείσιμο του αρχείου όταν δεν χρειάζεται (close)
  - Ανάγνωση (read) και εγγραφή δεδομένων (write)
    - Δίνεται περιοχή μνήμης για τα δεδομένα και μήκος
    - Κάθε αρχείο έχει έναν δείκτη τρέχουσα θέσης
    - Ο δείκτης αλλάζει μετά από κάθε ανάγνωση / εγγραφή
  - Ρητή μετακίνηση δείκτη τρέχουσας θέσης (lseek)
    - Ως προς την αρχή, το τέλος ή την τρέχουσα θέση
  - Κλείδωμα αρχείων και λειτουργίες ελέγχου (fcntl)

Κλήσεις αρχείων (3 από 5)

- Δημιουργία αγωγών (pipe)
  - Οι αγωγοί μοιάζουν με προσωρινά αρχεία
  - Επιστρέφονται δύο περιγραφείς
    - Ένας κατάλληλος για εγγραφή στον αγωγό
    - Ένας κατάλληλος για ανάγνωση από τον αγωγό
- Μεταδεδομένα αρχείων
  - Για κάθε αρχείο το σύστημα καταγράφει πληροφορίες
  - stat: διαβάζει πληροφορίες αρχείου με βάση το όνομα
  - fstat: διαβάζει πληροφορίες ανοιχτού αρχείου (μέσω fd)

Κλήσεις αρχείων (4 από 5)

Η συσκευή στην οποία βρίσκεται το αρχείο
Αριθμός κόμβου i (ποιο αρχείο της συσκευής)
Κατάσταση αρχείου (περιλαμβάνει πληροφορίες προστασίας)
Αριθμός συνδέσμων προς το αρχείο
Ταυτότητα του ιδιοκτήτη του αρχείου
Ομάδα στην οποία ανήκει ο ιδιοκτήτης
Μέγεθος αρχείου (σε byte)
Χρόνος δημιουργίας
Χρόνος τελευταίας προσπέλασης
Χρόνος τελευταίας τροποποίησης

- Μεταδεδομένα αρχείων
  - Πληροφορίες που επιστρέφουν οι stat/fstat
    - Στοιχεία του αρχείου, προστασία
    - Στοιχεία χρήστη / ομάδας, χρόνοι

Κλήσεις αρχείων (5 από 5)

Κλήση συστήματος	Περιγραφή
s = mkdir(path, mode)	Δημιουργεί ένα νέο κατάλογο
s = rmdir(path)	Διαγράφει έναν κατάλογο
s = link(oldpath, newpath)	Δημιουργεί ένα σύνδεσμο προς ένα υπάρχον αρχείο
s = unlink(path)	Διαγράφει ένα αρχείο από έναν κατάλογο
s = chdir(path)	Αλλάζει τον κατάλογο εργασίας
dir = opendir(path)	Ανοίγει έναν κατάλογο για ανάγνωση
s = closedir(dir)	Κλείνει έναν κατάλογο
dirent = readdir(dir)	Διαβάζει μια καταχώριση καταλόγου
rewinddir(dir)	Επιστρέφει στην αρχή ενός καταλόγου για να τον ξαναδιαβάσει

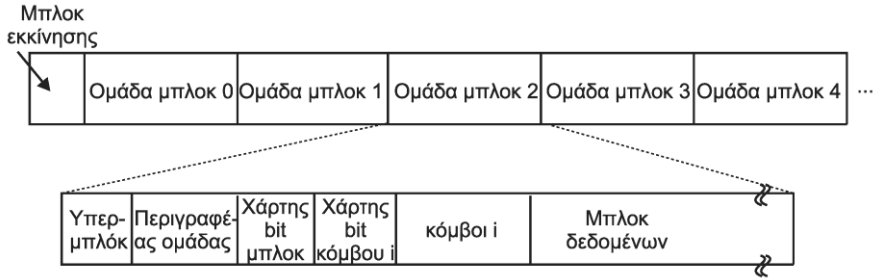
- Κλήσεις που αφορούν καταλόγους
  - Δημιουργία (mkdir) και διαγραφή καταλόγων (rmdir)
  - Δημιουργία (link) και διαγραφή συνδέσμων (unlink)
  - Αλλαγή τρέχοντος καταλόγου (chdir)
  - Άνοιγμα (opendir), κλείσιμο (closedir), αρχή (rewinddir)
  - Διάβασμα επόμενης εγγραφής καταλόγου (readdir)

Υλοποίηση αρχείων (1 από 13)

Αντικείμενο	Περιγραφή	Λειτουργία
Υπερμπλόκ (Superblock)	συγκεκριμένο σύστημα αρχείων	read_inode, sync_fs
Dentry	καταχώριση καταλόγου, ένα μοναδικό συστατικό μιας διαδρομής	create, link
Κόμβος i (i-node)	ειδικό αρχείο	d_compare, d_delete
Αρχείο	ανοιχτό αρχείο συσχετισμένο με μια διεργασία	read, write

- Εικονικό Σύστημα Αρχείων (VFS)
  - Βασικές αφαιρέσεις για όλα τα συστήματα αρχείων
    - Υπερμπλόκ (superblock): βασικές πληροφορίες συστήματος
    - Κόμβοι i (i-nodes): στοιχεία ενός αρχείου ή (συσκευής)
    - Καταχώριση καταλόγου (dentry): στοιχεία ενός καταλόγου
    - Δομή αρχείου (file): στοιχεία ενός ανοιχτού αρχείου
    - Το σύστημα αρχείων μεταφράζει τις δομές του σε αυτές του VFS

Υλοποίηση αρχείων (2 από 13)

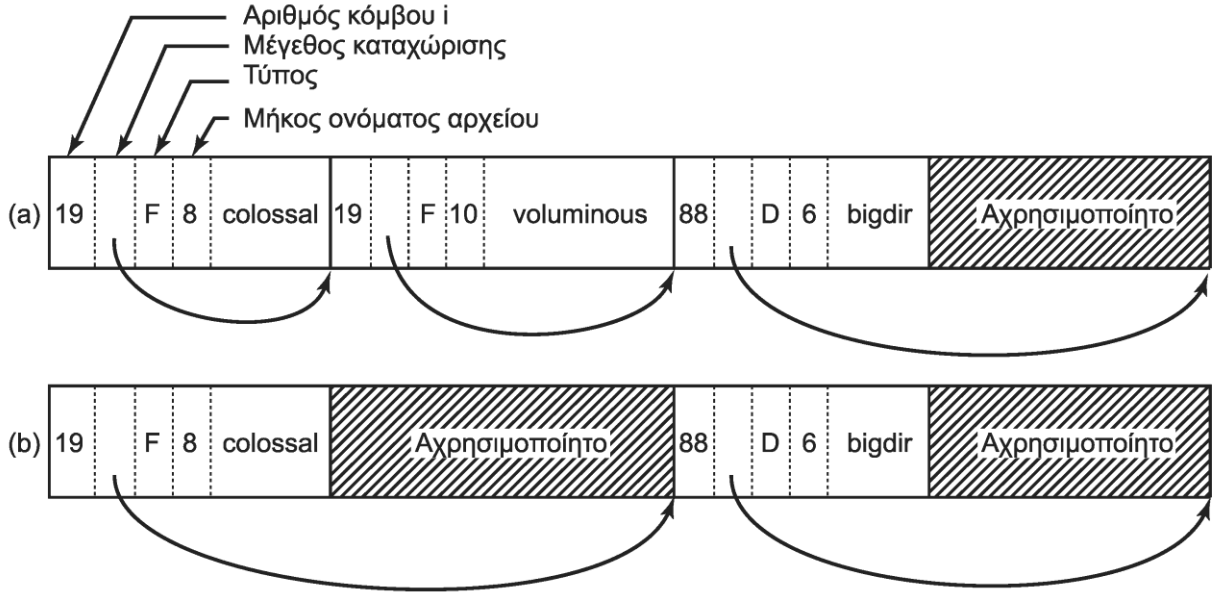


- Το σύστημα αρχείων ext2
  - Το μπλοκ 0 (boot) δεν χρησιμοποιείται από το ext2
  - Τα υπόλοιπα διαιρούνται σε ανεξάρτητες ομάδες
  - Κάθε ομάδα περιέχει μία σειρά από δομές
  - Υπερμπλόκ: βασικές πληροφορίες όλης της διαμέρισης
  - Περιγραφέας ομάδας: βασικές πληροφορίες ομάδας
  - Χάρτες bit για ελεύθερα μπλοκ και κόμβους i
  - Κόμβοι i και μπλοκ δεδομένων

Υλοποίηση αρχείων (3 από 13)

- Τοποθετούμε γειτονικά αρχεία στην ίδια ομάδα
  - Αρχεία στην ίδια ομάδα με τον γονικό τους κατάλογο
  - Μπλοκ αρχείων στην ίδια ομάδα με τον κόμβο i
  - Προκατανομή 8 μπλοκ όποτε μεγαλώνει ένα αρχείο
- Άνοιγμα αρχείων
  - Αρχίζουμε από κόμβο i τρέχοντα καταλόγου ή ρίζας
    - Κόμβος i ρίζας: σε προκαθορισμένο σημείο στο δίσκο
    - Κόμβος i τρέχοντα καταλόγου: σε περιγραφέα διεργασίας
  - Από εκεί διαβάζουμε τα δεδομένα του καταλόγου
  - Αυτά δείχνουν σε έναν νέο κόμβο i, και ούτω καθεξής

Υλοποίηση αρχείων (4 από 13)



- Δομή αρχείων καταλόγων
  - Ακέραιο πλήθος μπλοκ δίσκου (συμπλήρωση με NUL)
  - Συνεχόμενες αλλά όχι ταξινομημένες εγγραφές
  - Αριθμός κόμβου i, μήκος, τύπος, μήκος ονόματος, όνομα

Υλοποίηση αρχείων (5 από 13)

Πεδίο	Byte	Περιγραφή
Mode	2	Τύπος αρχείου, bit προστασίας, setuid, setgid
Nlinks	2	Αριθμός καταχωρίσεων καταλόγων που δείχνουν σε αυτόν τον κόμβο i
Uid	2	Το UID του ιδιοκτήτη του αρχείου
Gid	2	Το GID του ιδιοκτήτη του αρχείου
Size	4	Το μέγεθος του αρχείου σε byte
Addr	60	Η διεύθυνση των πρώτων 12 μπλοκ δίσκου, και στη συνέχεια των 3 έμμεσων μπλοκ
Gen	1	Αριθμός δημιουργίας (αυξάνεται κατά 1 κάθε φορά που επαναχρησιμοποιείται ο κόμβος i)
Atime	4	Χρόνος τελευταίας προσπέλασης του αρχείου
Mtime	4	Χρόνος τελευταίας τροποποίησης του αρχείου
Ctime	4	Χρόνος τελευταίας τροποποίησης του κόμβου i (δεν αφορά τις δύο προηγούμενες περιπτώσεις)

- Δομή αρχείων καταλόγων
  - Η αναζήτηση μπορεί να πάρει αρκετό χρόνο
  - Το Linux χρησιμοποιεί κρυφή μνήμη καταλόγων
- Δομή κόμβων i
  - Περιέχουν τουλάχιστον τα πεδία των stat/fstat

Υλοποίηση αρχείων (6 από 13)

- Δομή κόμβων i
  - Περιέχουν επιπλέον δείκτες προς τα μπλοκ του αρχείου
  - Πρόσθετα πεδία για καταλόγους και ειδικά αρχεία
- Ο πυρήνας διατηρεί έναν πίνακα ανοιχτών κόμβων i
- Υλοποίηση των πράξεων ανάγνωσης/εγγραφής
  - Οι κλήσεις περιέχουν τον περιγραφέα του αρχείου
  - Τι περιέχει ο πίνακας περιγραφέων της διεργασίας;
  - Έστω ότι περιέχει έναν δείκτη στον πίνακα κόμβων i
    - Αν ο δείκτης τρέχουσας θέσης περιέχεται στον πίνακα κόμβων i;
    - Τότε δύο διεργασίες δεν μπορούν να διαβάζουν ανεξάρτητα

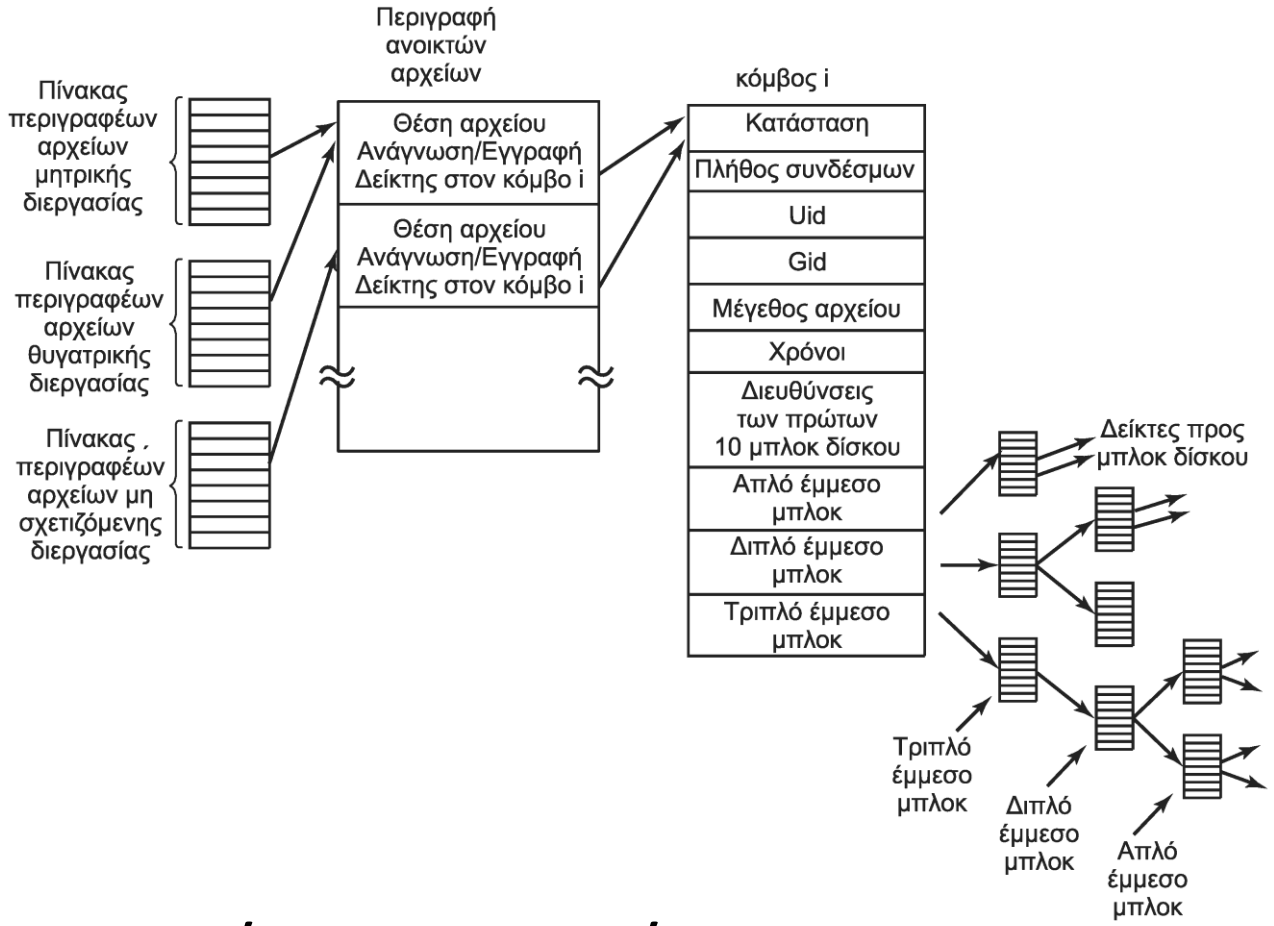
Υλοποίηση αρχείων (7 από 13)

- Υλοποίηση πράξεων ανάγνωσης/εγγραφής
  - Έστω ότι ο δείκτης είναι σε πίνακα περιγραφέων
    - Δεν μπορεί να συγχρονιστεί η θυγατρική με τη μητρική
    - Έστω ένα σενάριο s με τις εντολές p1 και p2
    - Έστω ότι εκτελούμε s > x
    - Η p2 πρέπει να βάλει την έξοδό της μετά από την p1
    - Άρα πρέπει να μοιράζονται το δείκτη θέσης!
  - Χρήση πίνακα περιγραφής ανοιχτών αρχείων

Υλοποίηση αρχείων (8 από 13)

- Πίνακας περιγραφής ανοιχτών αρχείων
  - Ο πίνακας περιγραφέων δείχνει σε αυτόν
  - Εκεί αποθηκεύεται ο δείκτης θέσης του αρχείου
  - Ο πίνακας αυτός δείχνει στον πίνακα κόμβων i
  - Έστω ότι δημιουργούμε τις p1 και p2 με fork
    - Ο πίνακας περιγραφέων αντιγράφεται
    - Δείχνουν στην ίδια θέση του πίνακα ανοικτών αρχείων
    - Δύο τυχαίες διεργασίες δείχνουν διαφορετικές θέσεις

Υλοποίηση αρχείων (9 από 13)



Πίνακες αρχείων στο Linux

Υλοποίηση αρχείων (10 από 13)

- Εντοπισμός των μπλοκ ενός αρχείου
  - Οι 12 πρώτοι δείκτες δείχνουν σε μπλοκ του αρχείου
  - Ο δείκτης 13 δείχνει σε ένα απλό έμμεσο μπλοκ
    - Περιέχει δείκτες προς τα επόμενα μπλοκ του αρχείου
  - Ο δείκτης 14 δείχνει σε ένα διπλό έμμεσο μπλοκ
    - Περιέχει πολλούς δείκτες προς απλά έμμεσα μπλοκ
  - Ο δείκτης 15 δείχνει σε ένα τριπλό έμμεσο μπλοκ
    - Περιέχει πολλούς δείκτες προς διπλά έμμεσα μπλοκ
  - Μπορούν έτσι να υποστηριχθούν τεράστια αρχεία

Υλοποίηση αρχείων (11 από 13)

- Το σύστημα αρχείων ext4
  - Το ext2 γράφει τα λερωμένα μπλοκ με καθυστέρηση
    - Υπάρχει κίνδυνος απώλειας δεδομένων
  - Το ext4 αντιμετωπίζει το πρόβλημα με ημερολόγιο
    - Οι λειτουργίες γράφονται άμεσα σε εγγραφές ημερολογίου
    - Στη συνέχεια γράφονται οι πραγματικές αλλαγές στο δίσκο
    - Στο τέλος διαγράφονται οι εγγραφές ημερολογίου
  - Το ext4 είναι απόλυτα συμβατό με το ext2
    - Παρόμοια διάταξη και δομές, συν το ημερολόγιο
    - Προσθήκη extents: συνεχόμενες περιοχές μπλοκ

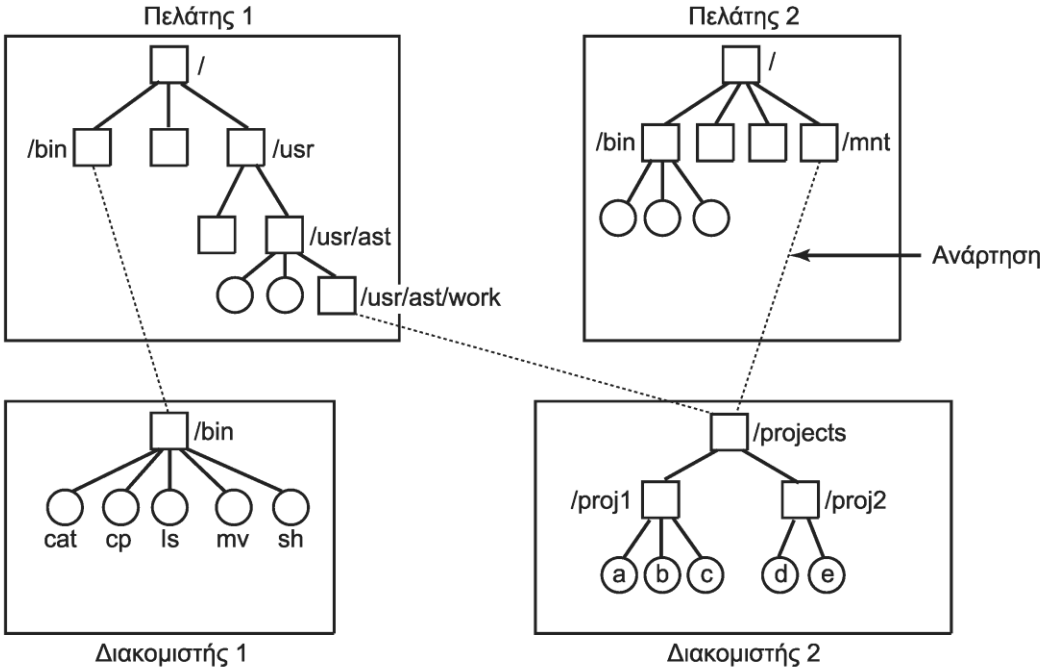
Υλοποίηση αρχείων (12 από 13)

- Συσκευή μπλοκ ημερολογίου (JBD)
  - Η συσκευή JBD υποστηρίζει τρεις κύριες δομές
  - Εγγραφή καταγραφής (log record)
    - Μία λειτουργία που αλλάζει κάποιο μπλοκ στο δίσκο
  - Χειριστήριο αδιαίρετης λειτουργίας (atomic op handle)
    - Ομαδοποιεί μία σειρά σχετικών λειτουργιών
    - Παράδειγμα: ένα write μπορεί να αλλάζει πολλά μπλοκ
  - Συναλλαγή (transaction)
    - Ομαδοποιεί πολλές αδιαίρετες λειτουργίες
    - Όταν εκτελεστεί η συναλλαγή, διαγράφεται από τη JBD

Υλοποίηση αρχείων (13 από 13)

- Το σύστημα αρχείων /proc
  - Κάθε διεργασία έχει κατάλογο στο σύστημα /proc
    - Παράδειγμα: /proc/619 για τη διεργασία με PID 619
  - Περιέχει αρχεία με στοιχεία για τη διεργασία
    - Παράδειγμα: γραμμή διαταγής, περιβάλλον
  - Οι πληροφορίες επιστρέφονται από τον πυρήνα
    - Οι αναγνώσεις ανακατευθύνονται στον πυρηνα
  - Αρχεία με άλλες πληροφορίες για το σύστημα

Δικτυακή αποθήκευση (1 από 4)



- Αρχιτεκτονική του NFS
  - Κάθε μηχανή μπορεί να είναι διακομιστής ή/και πελάτης
  - Διακομιστές: εξάγουν καταλόγους προς πρόσβαση
  - Πελάτες: αναρτούν καταλόγους στο τοπικό τους δένδρο

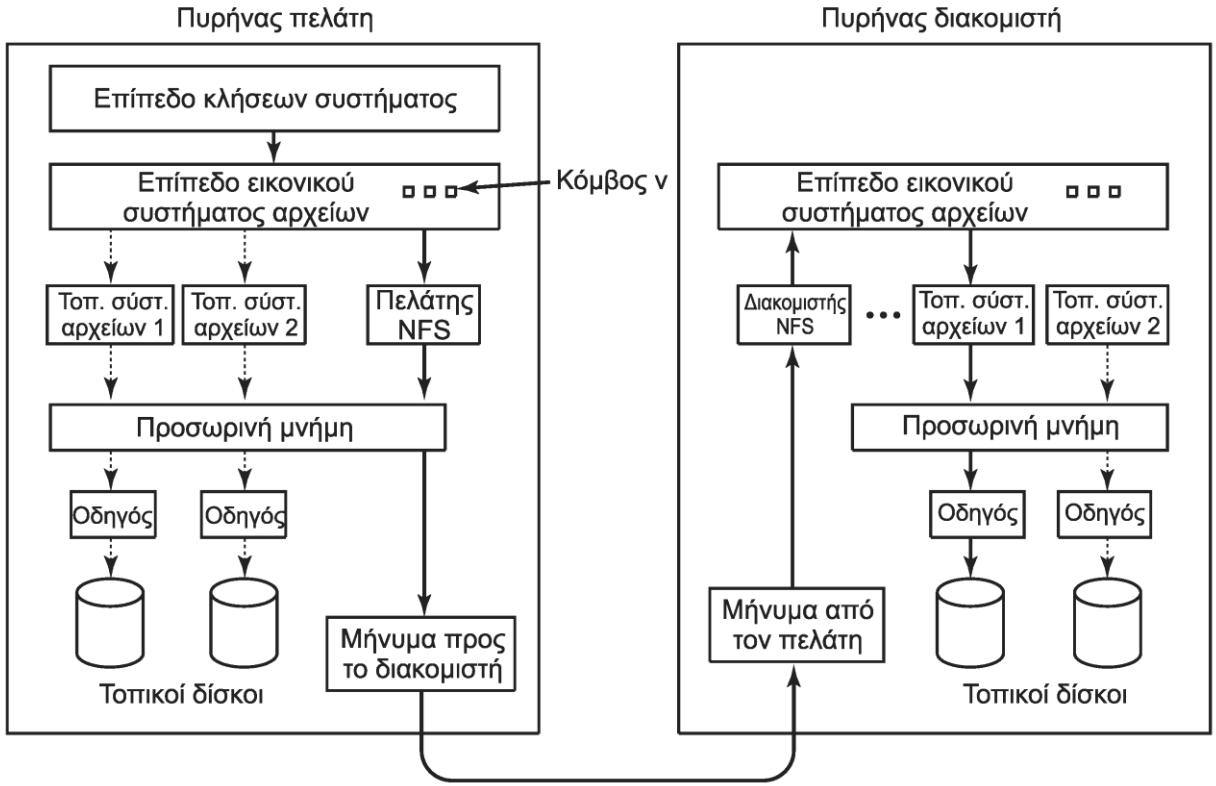
Δικτυακή αποθήκευση (2 από 4)

- Πρωτόκολλα του NFS
  - Διασφαλίζουν συμβατότητα πελατών και διακομιστών
- Πρωτόκολλο ανάρτησης καταλόγων
  - Ο πελάτης στέλνει ένα μήνυμα ζητώντας έναν κατάλογο
  - Ο διακομιστής επιστρέφει ένα χειριστήριο (file handle)
    - Το χειριστήριο προσδιορίζει τον κατάλογο στο διακομιστή
  - Ανάρτηση καταλόγων κατά την εκκίνηση
  - Ανάρτηση καταλόγων όταν χρειαστούν
    - Μπορούμε να έχουμε πολλούς διακομιστές
    - Ο πρώτος που θα απαντήσει επιλέγεται για χρήση

Δικτυακή αποθήκευση (3 από 4)

- Πρωτόκολλο προσπέλασης
  - Μηνύματα διαχείρισης καταλόγων και αρχείων
  - Δεν υπάρχουν μηνύματα ανοίγματος / κλεισίματος
    - Αντί για άνοιγμα αρχείου έχουμε μήνυμα lookup
    - Επιστρέφει ένα χειριστήριο για το αρχείο
  - Στα επόμενα μηνύματα ο πελάτης στέλνει χειριστήριο
  - Επιπλέον αναφέρει από πού θέλει να διαβάσει/γράψει
- Υλοποίηση του NFS
  - Οι κόμβοι v του VFS δείχνουν σε κόμβους r του NFS
  - Οι κόμβοι r περιέχουν χειριστήρια και δείκτες αρχείων

Δικτυακή αποθήκευση (4 από 4)



- Πελάτης: μηνύματα σε διακομιστή για δεδομένα
- Διακομιστής: επικοινωνεί με σύστημα αρχείων

Ασφάλεια στο Linux

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 8:** Το ΛΣ Linux

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

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

ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Θεμελιώδεις έννοιες ασφάλειας (1 από 4)

Δυαδικό	Συμβολισμός	Επιτρεπόμενη πρόσβαση στο αρχείο
111000000	rwx------	Ο ιδιοκτήτης μπορεί να διαβάσει, να γράψει, και να εκτελέσει
111111000	rwxrwx---	Ο ιδιοκτήτης και η ομάδα του μπορούν να διαβάσουν, να γράψουν, και να εκτελέσουν
110100000	rw-r-----	Ο ιδιοκτήτης μπορεί να διαβάσει και να γράψει, η ομάδα μπορεί να διαβάσει
110100100	rw-r--r--	Ο ιδιοκτήτης μπορεί να διαβάσει και να γράψει, όλοι οι άλλοι μπορούν να διαβάσουν
000000000	---------	Κανένας δεν έχει το παραμικρό δικαίωμα πρόσβασης
000000111	------rwx	Μόνον οι άλλοι έχουν πρόσβαση (περίεργο, αλλά νόμιμο)

- Κάθε χρήστης έχει μία ταυτότητα χρήστη (UID)
- Ομάδες χρηστών με ταυτότητα ομάδας (GID)
  - Τα GID και UID είναι ακέραιοι (συνήθως 16 bit)
- Κάθε διεργασία διαθέτει το UID / GID του χρήστη της
- Κάθε αρχείο αναφέρει το UID / GID του δημιουργού του
  - Δίνονται δικαιώματα χρήστη, ομάδας και άλλων
  - Ανάγνωση/εγγραφή/εκτέλεση (rwx)

Θεμελιώδεις έννοιες ασφάλειας (2 από 4)

- Χρήστης με UID 0: υπερχρήστης (superuser)
  - Ονομάζεται και βασικός χρήστης (root)
  - Μπορεί να διαβάσει και να γράψει όλα τα αρχεία
  - Μπορεί να εκτελεί προνομιούχες κλήσεις
- Ίδιο μοντέλο δικαιωμάτων για καταλόγους
  - Το δικαίωμα x σημαίνει αναζήτηση στον κατάλογο
- Ίδια δικαιώματα για ειδικά αρχεία συσκευών

Θεμελιώδεις έννοιες ασφάλειας (3 από 4)

- Το bit προστασίας SETUID
  - Η διεργασία έχει πραγματικό και ενεργό UID
  - Αρχικά έχουν ίδια τιμή (το UID του χρήστη)
  - Έστω εκτελέσιμο πρόγραμμα με ενεργό SETUID
    - Με την εκτέλεση αλλάζει το UID της διεργασίας
    - Η διεργασία παίρνει προσωρινά το UID του αρχείου
  - Ένα πρόγραμμα μπορεί να ελέγχει και τα δύο UID

Θεμελιώδεις έννοιες ασφάλειας (4 από 4)

- Το bit προστασίας SETUID
  - Παράδειγμα: το passwd αλλάζει το συνθηματικό μας
    - Το αρχείο με τα συνθηματικά ανήκει στον υπερχρήστη
    - Το πρόγραμμα passwd έχει UID 0 και είναι SETUID
    - Ο χρήστης προσωρινά γίνεται υπερχρήστης
    - Εκτελεί όμως μόνο τον κώδικα του passwd
- Υπάρχει και bit προστασίας SETGID
  - Δουλεύει όπως το SETUID για αλλαγή του GID
  - Χρησιμοποιείται σπάνια όμως

Κλήσεις ασφάλειας (1 από 2)

Κλήση συστήματος	Περιγραφή
s = chmod(path, mode)	Αλλάζει τον τρόπο προστασίας ενός αρχείου
s = access(path, mode)	Έλεγχος πρόσβασης με τα πραγματικά UID και GID
uid = getuid()	Παίρνει το πραγματικό UID
uid = geteuid()	Παίρνει το ενεργό UID
gid = getgid()	Παίρνει το πραγματικό GID
gid = getegid()	Παίρνει το ενεργό GID
s = chown(path, owner, group)	Αλλάζει ιδιοκτήτη και ομάδα
s = setuid(uid)	Αναθέτει τιμή στο UID
s = setgid(gid)	Αναθέτει τιμή στο GID

Κλήσεις ασφάλειας (2 από 2)

- Η chmod αλλάζει τα δικαιώματα προσπέλασης
- Η access ελέγχει αν επιτρέπεται η πρόσβαση
  - Χρησιμοποιεί το πραγματικό UID/GID
- Κλήσεις ελέγχου πραγματικού/ενεργού UID/GID
- Η chown αλλάζει το UID και το GID ενός αρχείου
  - Μόνο ο υπερχρήστης τις χρησιμοποιεί
- Οι setuid/setgid αλλάζουν UID/GID διεργασίας
  - Μόνο ο υπερχρήστης τις χρησιμοποιεί

Υλοποίηση ασφάλειας

- Όταν συνδέεται ένας χρήστης εκτελείται η login
  - Της μεταβιβάζεται το όνομα του χρήστη από την getty
  - Διαβάζει και κατακερματίζει τον κωδικό πρόσβασης
  - Είναι UID/GID root οπότε διαβάζει το αρχείο κωδικών
  - Αν ταιριάζει ο κωδικός, διαβάζει UID/GID/κέλυφος χρήστη
  - Η login αλλάζει στα UID / GID του χρήστη με setuid/setgid
  - Εκτελεί με exec το κέλυφος του χρήστη
- Έλεγχος κατά το άνοιγμα/δημιουργία αρχείων
  - Ελέγχεται αν το UID / GID επιτρέπουν πρόσβαση
  - Στη συνέχεια δεν γίνεται ξανά έλεγχος

Τέλος Ενότητας #8

Μάθημα: Λειτουργικά Συστήματα, **Ενότητα # 8:** Το ΛΣ Linux

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης