

Εργασία 3 – Δικτυακό σύστημα αρχείων

Υλοποιήστε ένα (απλό) δικτυακό σύστημα αρχείων που επιτρέπει σε διεργασίες να δημιουργούν και να προσπελάζουν απομακρυσμένα αρχεία σε ένα άλλο μηχάνημα, με διαφανή τρόπο. Η επιθυμητή διαφάνεια πρέπει να επιτευχθεί μέσω ενδιάμεσου λογισμικού που παρέχει στην εφαρμογή λειτουργίες παρόμοιες με αυτές του τοπικού συστήματος αρχείων:

int mynfs_init(char *ipaddr, int port, int cacheblocks, int blocksize, int freshT);	Αρχικοποίηση στοιχείων επαφής του εξυπηρετητή, και προσδιορισμός του αριθμού και του μεγέθους των μπλοκ στην κρυφή μνήμη και της διάρκειας φρεσκάδας τους (σε δευτερόλεπτα).
int mynfs_open(char *fname, int flags);	Άνοιγμα/δημιουργία αρχείου ανάλογα με τα flags (O_CREAT, O_EXCL, O_TRUNC, O_RDWR, O_RDONLY, O_WRONLY).
int mynfs_read(int fd, void *buf, size_t n);	Διάβασμα δεδομένων από το αρχείο.
int mynfs_write(int fd, void *buf, size_t n);	Γράψιμο δεδομένων στο αρχείο.
int mynfs_seek(int fd, off_t pos, int whence);	Μετακίνηση θέσης ανάγνωσης/εγγραφής.
int mynfs_ftruncate(int fd, off_t len);	Κόψιμο αρχείου.
int mynfs_close(int fd);	Κλείσιμο αρχείου.

Βασική λειτουργικότητα λογισμικού για την πλευρά του πελάτη. Η διεπαφή εφαρμογής που προσφέρει το λογισμικό πελάτη πρέπει να υλοποιηθεί στο επίπεδο του ενδιάμεσου λογισμικού μέσω ενός πρωτοκόλλου request-reply που υποστηρίζει αποκλειστικά τις εξής απομακρυσμένες λειτουργίες: (1) εντοπισμός/δημιουργία ενός αρχείου και λήψη του «εσωτερικού» αναγνωριστικού, (2) διάβασμα δεδομένων από αρχείο με βάση το αναγνωριστικό του, (3) γράψιμο δεδομένων σε αρχείο με βάση το αναγνωριστικό του. Οι παραπάνω αλληλεπιδράσεις πρέπει να υλοποιηθούν πάνω από UDP/IP με σημασιολογία at-least-once για να μην χρειάζεται ο εξυπηρετητής να κρατά ξεχωριστή κατάσταση για κάθε πελάτη και να μπορεί να επανέρχεται με διαφανή τρόπο μετά από βλάβη χωρίς την χρήση σταθερής μνήμης/αποθήκης.

Βασική λειτουργικότητα λογισμικού για την πλευρά του εξυπηρετητή. Ο εξυπηρετητής δεν παρέχει κάποια διεπαφή εφαρμογής. Προσφέρει στους πελάτες πρόσβαση στο τοπικό σύστημα αρχείων, με ρίζα έναν κατάλογο που δίνεται ως όρισμα κατά την εκκίνηση του. Ο εξυπηρετητής αντιστοιχίζει κάθε αρχείο που προσπελάζεται από πελάτες σε ένα «εσωτερικό» αριθμητικό αναγνωριστικό που χρησιμοποιείται ως αναφορά για τις υπόλοιπες αλληλεπιδράσεις πελάτη-εξυπηρετητή που αφορούν το συγκεκριμένο αρχείο, έτσι ώστε να μην ξαναστέλνεται το όνομα του αρχείου κάθε φορά πάνω από το δίκτυο. Σημείωση: Σκεφτείτε τι πρόβλημα δημιουργεί αυτό και πώς μπορεί να λυθεί με πρακτικό τρόπο.

Κρυφή μνήμη. Για να επιταχυνθεί η ανάγνωση, ο πελάτης ζητά από τον εξυπηρετητή και κρατά σε κρυφή μνήμη ολόκληρα μπλοκ του αρχείου, το μέγεθος των μπλοκ είναι σταθερό και ανεξάρτητο από το πόσα bytes διαβάζει/γράφει η εφαρμογή. Ο έλεγχος εγκυρότητας των μπλοκ πρέπει να υλοποιηθεί στο πνεύμα του NFS (στην πλευρά του εξυπηρετητή, με βάση την χρονοσφραγίδα αλλαγής που διατηρεί το σύστημα αρχείων). Ο αριθμός μπλοκ στην κρυφή μνήμη, το μέγεθος τους και η διάρκεια φρεσκάδας καθορίζεται από την εφαρμογή μέσω της mynfs_init(). Όταν εξαντληθεί ο χώρος της κρυφής μνήμης, πρέπει να εκδιώκονται ένα ή περισσότερα μπλοκ με βάση μια πολιτική που θα αποφασίσετε εσείς. Οι αλλαγές που κάνει η εφαρμογή μεταδίδονται άμεσα από τον πελάτη στον εξυπηρετητή, ο οποίος με την σειρά του τις περνάει άμεσα στον δίσκο.

Δοκιμές / Μετρήσεις. Κάντε κατάλληλες δοκιμές για να επιβεβαιώσετε την ορθή λειτουργία της υλοποίησης σας. Επίσης, μετρήστε τόσο τον αριθμό αλληλεπιδράσεων request-reply ανάμεσα στον πελάτη και στον εξυπηρετητή όσο και τον όγκο δεδομένων που στέλνονται πάνω από το δίκτυο για διάφορα σενάρια ανάγνωσης/γραψίματος ενός αρχείου, ως συνάρτηση του μεγέθους της κρυφής μνήμης και της διάρκειας φρεσκάδας σε σχέση με το να μην υποστηρίζεται καθόλου κρυφή μνήμη.

Η πλευρά του εξυπηρετητή πρέπει να υλοποιηθεί σε C με system calls πάνω από το σύστημα αρχείων του λειτουργικού. Για την πλευρά του πελάτη μπορείτε να χρησιμοποιήσετε όποια γλώσσα προγραμματισμού επιθυμείτε (χωρίς να κάνετε καμία υπόθεση για τα δεδομένα που γράφει/διαβάζει η εφαρμογή στα/από τα αρχεία). Ακολουθήστε τις οδηγίες παράδοσης (στο eclass). Εκτός από τον κώδικα, πρέπει να ετοιμάσετε μια παρουσίαση που θα περιγράφει την υλοποίηση, τα πειράματά σας και τα αποτελέσματα.

Φροντιστήριο/συζήτηση: Πέμπτη 14 Απριλίου 2022 Ημερομηνία παράδοσης: Σάββατο 14 Μαΐου 2022, 22:00
--