

Εργασία 3 – Δικτυακό σύστημα αρχείων

Υλοποιήστε ένα απλό δικτυακό σύστημα αρχείων που επιτρέπει σε διεργασίες πελάτες να προσπελάζουν απομακρυσμένα αρχεία που βρίσκονται σε ένα άλλο μηχάνημα εξυπηρετητή. Η προσπέλαση ενός αρχείου πρέπει να γίνεται με διαφανή τρόπο, μέσω ενδιάμεσου λογισμικού που παρέχει στην εφαρμογή κατάλληλες λειτουργίες στο πνεύμα του παρακάτω API:

<code>int mynfs_set_srv_addr(char *ipaddr, int port);</code>	Στοιχεία επικοινωνίας εξυπηρετητή.
<code>int mynfs_set_cache_validity(int secs);</code>	Προσδιορισμός χρονικής διάρκειας εγκυρότητας δεδομένων στην κρυφή μνήμη.
<code>O_CREAT, O_EXCL, O_TRUNC, O_RDWR, O_RDONLY, O_WRONLY</code>	Επιλογές για τα flags που δίνονται στην <code>open</code> .
<code>int mynfs_open(char *fname, int flags);</code>	Άνοιγμα/δημιουργία αρχείου για διάβασμα/γράψιμο, ανάλογα με τα flags.
<code>int mynfs_read(int fd, void *buf, size_t n);</code>	Διάβασμα δεδομένων από το αρχείο.
<code>int mynfs_write(int fd, void *buf, size_t n);</code>	Γράψιμο δεδομένων στο αρχείο.
<code>int mynfs_seek(int fd, off_t pos, int whence);</code>	Μετακίνηση θέσης ανάγνωσης/εγγραφής.
<code>int mynfs_close(int fd);</code>	Κλείσιμο αρχείου.

Το λογισμικό σας πρέπει να υποστηρίζει, στο δικό του επίπεδο, τις εξής αλληλεπιδράσεις ανάμεσα στον πελάτη και τον εξυπηρετητή: (1) εντοπισμός/δημιουργία αρχείου, (2) ανάγνωση bytes από το αρχείο, και (3) εγγραφή bytes στο αρχείο. Οι αλληλεπιδράσεις πρέπει να σχεδιαστούν με σημασιολογία at-least-once, και έτσι ώστε να μην χρειάζεται ο εξυπηρετητής να κρατά ξεχωριστή κατάσταση για κάθε πελάτη. Η υλοποίηση πρέπει να γίνει πάνω από UDP/IP.

Κάθε φορά που η εφαρμογή ανοίγει ένα αρχείο, το λογισμικό πελάτη επιστρέφει έναν νέο file descriptor (ακόμα και αν αυτός εσωτερικά αντιστοιχίζεται στο ίδιο αρχείο), που η εφαρμογή χρησιμοποιεί ως παράμετρο στις υπόλοιπες λειτουργίες. Το λογισμικό πελάτη πρέπει να κάνει τους απαραίτητους ελέγχους με βάση και τα flags που χρησιμοποιήθηκαν στην `open()`, και να προχωρεί στις αντίστοιχες αλληλεπιδράσεις με τον εξυπηρετητή, με διαφανή τρόπο για την εφαρμογή. Ο εξυπηρετητής διαχειρίζεται έναν (για απλότητα, επίπεδο) χώρο ονομάτων στο πλαίσιο του τοπικού συστήματος αρχείων, με ρίζα έναν κατάλογο που δίνεται ως όρισμα του προγράμματος. Εσωτερικά, ο εξυπηρετητής αντιστοιχίζει κάθε αρχείο που προσπελάζεται σε ένα αριθμητικό αναγνωριστικό, το οποίο χρησιμοποιείται ως αναφορά στο αρχείο στο πλαίσιο των υπόλοιπων αλληλεπιδράσεων με τους πελάτες έτσι ώστε να μην ξαναστέλνεται το όνομα του αρχείου πάνω από το δίκτυο. Σκεφτείτε τι πρόβλημα δημιουργεί η χρήση τέτοιων αναγνωριστικών, και πώς αυτό μπορεί να λυθεί με πρακτικό τρόπο.

Για να επιταχυνθεί η προσπέλαση στα απομακρυσμένα αρχεία, ο πελάτης κρατά σε κρυφή μνήμη τμήματα του αρχείου ώστε αυτά να είναι άμεσα διαθέσιμα αν τυχόν τα ξαναζητήσει η εφαρμογή. Ο έλεγχος εγκυρότητας των δεδομένων της κρυφής μνήμης πρέπει να υλοποιηθεί στο πνεύμα του NFS. Το χρονικό διάστημα για το οποίο τα δεδομένα της κρυφής μνήμης θεωρούνται έγκυρα καθορίζεται από την εφαρμογή, μέσω της `set_cache_validity()`. Για ευκολία διαχείρισης της κρυφής μνήμης, ο πελάτης λαμβάνει/στέλνει τα περιεχόμενα των αρχείων σε τμήματα σταθερού μεγέθους, ανεξάρτητα από το πόσα bytes ζητά να διαβάσει/γράψει η εφαρμογή. Όταν εξαντληθεί ο χώρος της κρυφής μνήμης, πρέπει να εκδιώκονται ένα ή περισσότερα τμήματα με βάση μια πολιτική που θα αποφασίσετε εσείς. Τέλος, ο πελάτης πρέπει να μεταδίδει άμεσα στον εξυπηρετητή τις αλλαγές που κάνει η εφαρμογή στα δεδομένα του αρχείου.

Κάντε κατάλληλες δοκιμές για να επιβεβαιώσετε την ορθή λειτουργία της υλοποίησής σας, π.χ., μέσα από μια εφαρμογή που δημιουργεί ένα τοπικό και ένα απομακρυσμένο αντίγραφο ενός απομακρυσμένου αρχείου και στην συνέχεια συγκρίνει τα περιεχόμενα τους. Επίσης, αξιολογήστε την βελτίωση της απόδοσης που επιφέρει η χρήση της κρυφής μνήμης για διαφορετικά σενάρια ανάγνωσης/γραψίματος ενός αρχείου και μεγέθους κρυφής μνήμης.

Χρησιμοποιήστε όποια γλώσσα προγραμματισμού επιθυμείτε. Ο κώδικας σας πρέπει να μεταφράζεται και να εκτελείται κανονικά στο περιβάλλον Linux του εργαστηρίου. Εναλλακτικά, μπορείτε να επιδείξετε την υλοποίηση σε δικά σας laptop που θα πρέπει να συνδέονται στο ενσύρματο δίκτυο του εργαστηρίου. Σιγουρευτείτε ότι η επίδειξή σας δουλεύει σωστά. Δεν μπορείτε να υποθέσετε ότι τα περιεχόμενα των αρχείων που διαβάζει/γράφει η εφαρμογή είναι ASCII.

Ημερομηνία παράδοσης: **Σάββατο 5 Μαΐου 2018, 22:00.**

Παραδοτέα: (α) κώδικας με οδηγίες εγκατάστασης/χρήσης, (β) παρουσίαση του πρωτοκόλλου και της υλοποίησης, καθώς και των μετρήσεων και των συμπερασμάτων/παρατηρήσεων σας.