

AM:1115202000234

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΚΡΑΤΗΜΕΝΟΣ ΧΡΗΣΤΟΣ

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ : ΕΡΓΑΣΙΑ 2

Διαφορές στη Συνάρτηση bmap

στην νέα έκδοση μου, έχει προστεθεί ένα επιπλέον σταδίο για τα double-indirect blocks. Συγκεκριμένα, ο πίνακας `ip->addrs` έχει μέγεθος `NDIRECT+2` αντί για `NDIRECT+1`. Το επιπλέον στοιχείο οδηγεί σε ακόμα περισσότερα block. Αφού εξαντληθούν τα direct (`NDIRECT`) και το single-indirect (`ip->addrs[NDIRECT]`), η `bmap` κοιτάζει αν υπάρχει double-indirect (`ip->addrs[NDIRECT+1]`). Αν δεν υπάρχει, το δεσμεύει με `ballocc()` και κατόπιν φορτώνει το αντίστοιχο buffer (`bp`) για το πρώτο επίπεδο του double-indirect. Τέλος, προχωρά σε δεύτερο buffer (`bp2`), μέσα στο οποίο αναζητά το συγκεκριμένο block index, για αυτό και ορίσα `MAXFILE` το $(NDIRECT + NINDIRECT + NINDIRECT * NINDIRECT)$ αντί $(NDIRECT + NINDIRECT)$.

Επιπλέον αντί για `log_write(bp)` χρησιμοποιούμε `bwrite(bp)` για να αλλάξουμε τον τρόπο των ενημερώσεων για ένα νέο block

SYMLINK

Εχουμε προσθέσει έναν νέο τύπο αρχείου (`T_SYMLINK`) στα αρχεία `stat.h`, `syscall.h`,

Στο `sys_open()`, εντοπίζεται η λογική που χειρίζεται τα symlinks: πρώτα γίνεται έλεγχος αν υπάρχει η σημαία `O_NOFOLLOW`, που αν ρυθμιστεί εμποδίζει αυτόματα την περαιτέρω παρακολούθηση (follow) του συνδέσμου. Διαφορετικά, η συνάρτηση ακολουθεί αναδρομικά τους συνδέσμους μέχρι να βρεθεί ένα πραγματικό αρχείο (`T_FILE`, `T_DIR`) ή μέχρι να φτάσει σε κάποιο όριο για να αποφύγει πιθανή κυκλική αναφορά (infinite loop)

Στο αρχείο `fs.h`, ορίζουμε ως `NDIRECT = 11` αντί `12` και `NDIRECT+2` για το `inode`. Επίσης το `MAXFILE` είναι μεγαλύτερο, αφού συμπεριλαμβάνεται και το γινόμενο `NINDIRECT*NINDIRECT`.

Τέλος, όσον αφορά την `itrunc()`, η διαχείριση των block επεκτείνεται και σε double-indirect pointer, οπότε πρέπει να καθαριστούν (free) περισσότερα επίπεδα αν το αρχείο συρρικνωθεί ή διαγραφεί. Για αυτό έχω προσθέσει ένα βρόχο εμφωλευμένων pointers για το double-indirect. (χρηάζεται για τα μεγάλα αρχεία)

ΕΚΤΕΛΕΣΗ ΚΩΔΙΚΑ

Εφόσον βρισκομαστε στο `/xv6-project-2024` directory, μπορούμε να τρεξουμε `make` και έπειτα `make qemu` έτσι ώστε να ξεκινήσει το xv6 kernel. Στα πλαίσια της εργασίας μπορούμε

να τρεξουμε τις εξης εντολες, για τις οποιες ακολουθουν αντιστοιχα παραδειγματα:
symlinktest ,usertests -q,bigfile

ΕΝΔΕΙΚΤΙΚΑ ΠΑΡΑΔΕΙΓΜΑΤΑ:

bigfile:

```
xv6 kernel is booting
init: starting sh
$ bigfile
.....
.....
wrote 65883 blocks
reading bigfile
.....
.....
bigfile done; ok
$
```

symlinktest

```
$ symlinktest
Start: test symlinks
test symlinks: ok
Start: test concurrent symlinks
test concurrent symlinks: ok
$
```