Λειτουργικά Συστήματα

Εργασία 2

Ιωάννης Χουστουλάκης

1115202300296

Αύξηση μέγιστου μεγέθους αρχείου στο xv6

Τροποποίησα το struct inode στο file.h ώστε ο πίνακας addrs να έχει δυο παραπάνω διευθύνσεις εκτός των direct inodes για να μπορεί να καταχωρηθεί το double indirect inode

struct inode {

[…]

uint addrs[NDIRECT+2];

Τροποποίησα τη συνάρτηση bmap στο fs.c ώστε για περισσότερα blocks από αυτά που υποστηρίζονται στο single indirect block να καταχωρούνται στο double indirect block, του οποίου η διεύθυνση βρίσκεται addrs[NDIRECT+1], καταχωρώντας τη διεύθυνση του επιμέρους single indirect blocks στη διεύθυνση block\_number/NINDIRECT, βρίσκοντας στη συνέχεια τη διεύθυνση του data block καθεαυτού μέσα στο single indirect block στη θέση block\_number %NINDIRECT. Η θέση block\_number μετατοπίζεται κατά NINDIRECT θέσεις πριν γίνει αυτός ο υπολογισμός καθώς οι θέσεις μέσα στο double indirect block αριθμούνται ξεκινώντας από το μηδέν, όμως ως προς το input block number της bmap είναι οι θέσεις από την τιμή NINDIRECT+ NDIRECT και μετά.

Το μέγιστο μέγεθος αρχείων που υποστηρίζεται με αυτό τον τρόπο είναι NDIRECT + NINDIRECT + DOUBLEINDIRECT = 11 +256 +256\*256 = 65803 blocks, όπως ορίστηκε στο fs.h.

Προσπάθησα να φτιάξω τον κώδικα για το triple indirect block, που περιλαμβάνεται σε μορφή σχολίου στον κώδικα, καθώς δεν έτρεχε σωστά.

Συμπλήρωσα στη συνάρτηση itrunc τη λειτουργικότητα να ελευθερώνονται και τα blocks που περιέχονται στο double indirect block.

Αρχεία που επεξεργάστηκα για την αύξηση μέγιστου μεγέθους αρχείου:

Kernel\file.h

Kernel\fs.c

Kernel\fs.h

**Δημιουργία Symbolic Link**

Πρόθεσα στο fcntl.h την εντολή

#define O\_NOFOLLOW 0x800

Ώστε να οριστεί το flag για το αν θα ανοίξουμε το περιεχόμενο του link χωρίς να χρησιμοποιεί ίδιο bit με τα άλλα flags.

Όρισα στο stat.h: #define T\_SYMLINK 4 ως τύπο inode.

Δημιούργησα το πρωτότυπο της συνάρτησης sys\_symlink στο syscall.c και όρισα το call number της ως 22 στο syscall.h

Στην sysfile.c συμπλήρωσα στη συνάρτηση create να επιστρέφει το inode και για τον τύπο symbolic link. Υλοποίησα τη συνάρτηση sys\_symlink, στην οποία λαμβάνω τις παραμέτρους από την κλήση της με argstr. Δημιουργώ με την create το inode και γράφω σε αυτό το μήκος του target και στη συνέχεια το target. Στη συνάρτηση sys\_open πρόσθεσα τη λειτουργικότητα να ανοίγει τα symbolic links. Διαβάζω με read το target και το μήκος του. Συμπεριλαμβάνεται δικλείδα ασφαλείας για την περίπτωση που είναι κυκλική η πορεία των symbolic links. Κάθε φορά που ακολουθεί ένα symbolic link αυξάνει μια μεταβλητή κατά 1. Εάν ανοίγοντας το ίδιο αρχικό inode ξεπεράσουμε τα 10 symbolic links θεωρείται κυκλική διαδρομή και επιστρέφει σφάλμα. Ελέγχεται το flag no\_follow (με bitwise operation) στην περίπτωση που θέλουμε να ανοίξουμε το αρχικό symbolic link καθεαυτό σαν inode και όχι το target inode στο οποίο δείχνει ο σύνδεσμος.

Προστέθηκε η user function symlink στο user.h και στο usys.pl.

Αρχεία που επεξεργάστηκα για τα symbolic links:

Kernel\ fcntl.h

Kernel\ stat.h

Kernel\ syscall.c

Kernel\ syscall.h

User\user.h

user\usys.pl