

תרגול מס' 7

Unix File System

1 מבני נתונים במערכת ההפעלה

1.1 מבני נתונים בזיכרון לניהול מערכת הקבצים

מערכת ההפעלה מחזיקה בזיכרון שלושה סוגי טבלאות לניהול מערכת הקבצים :

1. טבלת הקבצים של תהליך.

2. טבלת קבצים כללית

3. טבלת Inodes של הקבצים הפתוחים

↔ כאשר מתבצעת פתיחה של קובץ, קוראת מערכת ההפעלה את ה-Inode של הקובץ לתוך הזיכרון. מבנה ה-Inode בזיכרון מכיל נתונים נוספים. לכל Inode יכול להיות עותק אחד בלבד בזיכרון של המחשב.

↔ בטבלת הקבצים הכללית כל כניסה מתארת פתיחה של קובץ¹. לכל כניסה מידע לגבי הגישה המבוקשת לקובץ (קריאה, כתיבה, ...), המיקום הנוכחי בקובץ, הצבעה ל-Inode המתאים לקובץ זה, ...

כל כניסה כוללת את מספר ה-file descriptors המצביעים עליה (reference counter).

↔ יכולות להיות שתי כניסות או יותר בטבלת הקבצים הכללית אשר מתארות את אותו קובץ ומצביעות לאותו Inode

2 System Calls לטיפול במערכת הקבצים

2.1 פעולות לטיפול בקובץ

2.1.1 Open/Creat

```
int open ( const char *path, int mode) ;
```

```
int creat ( const char *path, int mode) ;
```

פותח/יוצר קובץ. מחזיר file descriptor לשימוש בקריאות מערכת אחרות. אם הפעולה נכשלת מוחזר -1.

¹עבור פתיחות שונות (אפילו של אותו קובץ) יוקצו כניסות שונות בטבלה.

אוניברסיטת חיפה

החוג למדעי המחשב
מערכות הפעלה – תרגול

1. עבור open : חפש את מספר ה-Inode של הקובץ, קרא את ה-Inode לזיכרון ונעל אותו. אם הקובץ לא קיים - החזר שגיאה.
עבור creat : הקצה Inode חדש בדיסק, הקצה Inode בזיכרון ונעל אותו.
2. הקצה כניסה בטבלת הקבצים הכללית ואתחל אותה.
3. הקצה כניסה בטבלת הקבצים של התהליך ואתחל אותה.
4. עבור creat : אם הקובץ נועד לכתיבה, אתחל אותו ע"י שחרור כל הבלוקים שלו.
5. הורד נעילת Inode והחזר מספר כניסה בטבלת התהליך.

Close 2.1.2

`int close (int fildes) ;`

סוגר file descriptor, ומוחק אותו מהטבלה. מצליח - מחזיר 0, נכשל - 1.

1. שחרר את הכניסה בטבלת הקבצים של התהליך.
2. הורד 1 ממספר המתייחסים לכניסה בטבלת הקבצים הכללית.
3. אם אין יותר התייחסויות לכניסה בטבלת הקבצים :
 - 3.1. שחרר את הכניסה.
 - 3.2. שחרר Inode בזכרון.

Read 2.1.3

`int read (int fildes, void *buf, unsigned nbytes) ;`

קורא nbytes בתים מהקובץ המתואר ע"י fildes לתוך buf. מחזיר את מספר הבתים שנקראו בהצלחה (0 מוחזר כשמגיע לסוף הקובץ). אם הפעולה נכשלת, מוחזר -1

1. השג Inode מתוך טבלת הקבצים הכללית ונעל אותו.
2. קבע מיקום נוכחי בקובץ (מופיע בטבלת הקבצים הכללית).
3. כל עוד נותרו בתים לקריאה :
 - 3.1. חשב מספר בלוק בקובץ עפ"י מיקום נוכחי.
 - 3.2. תרגם למספר בלוק בדיסק.
4. אם סוף הקובץ עבור ל- (6).
5. קרא בלוק מהקובץ והעתק אותו למקום המבוקש.
6. הורד נעילת Inode ועדכן מיקום נוכחי בטבלת הקבצים הכללית.
7. החזר את מספר הבתים שנקראו.

אוניברסיטת חיפה

החוג למדעי המחשב
מערכות הפעלה – תרגול

Write 2.1.4

`int write (int fildes, const void *buf, unsigned nbytes) ;`

כותב nbytes בתים מהחוצץ המוצבע ע"י buf לקובץ המתואר ע"י fildes. מחזיר את מספר הבתים שנכתבו בהצלחה. אם הפעולה נכשלת מוחזר -1.

האלגוריתם דומה מאוד לאלגוריתם read בשינוי של כתיבה במקום קריאה והקצאת בלוקים חדשים לקובץ על הדיסק במקרה הצורך.

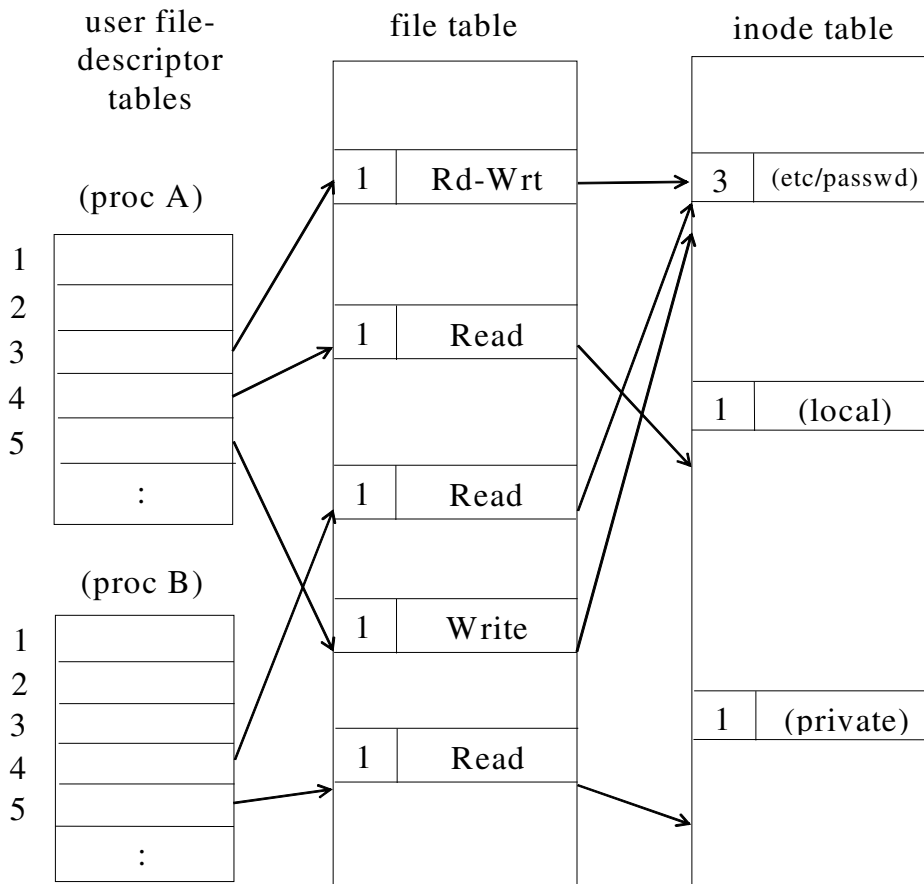
2.1.5 דוגמא

תהליך A ביצע :

```
open("/etc/passwd",O_RDWR);
open("local",O_RDONLY);
open("/etc/passwd",O_WRONLY);
```

תהליך B ביצע :

```
open("/etc/passwd",O_RDONLY);
open("private",O_RDONLY);
```



אוניברסיטת חיפה

החוג למדעי המחשב
מערכות הפעלה – תרגול

2.1.6 דוגמא – שימוש בקריאות מערכת לטיפול בקבצים

```
void main (int argc, char *argv[]) {
    int forkid, infile, outfile;
    if (argc!=3) exit(1);
    if ((infile=open(argv[1], O_RDONLY)) < 0) exit(2);
    if ((infile=creat(argv[2], 0666)) < 0) exit(3);
    if ((forkid=fork() == -1) exit (-1);
    dump(forkid, infile, outfile);
    exit(0);
}

void dump(int forkid, int infile, int outfile) {
    char ch;
    while(1) {
        if (read(infile, &ch, sizeof(char)) != sizeof(char)) {
            close(infile);
            close(outfile);
            return;
        }
        if ((forkid != 0)&&(islower(ch)) ch=toupper(ch);
        if ((forkid == 0)&&(isupper(ch)) ch=tolower(ch);
        write(outfile, &ch, 1);
    }
}
```

fork() יוצר תהליך חדש. הקבצים שהיו פתוחים לפני הפיצול נשארים משותפים. תוצאות הריצה אינן דטרמיניסטיות (עקב ה-context-switches). ה-case של התווים בקובץ פלט לא עקבי.

פלט אפשרי:

in.txt	out.txt
abcdefghijklmnopqrstuvwxyZ	ABCDe fghI JKLMnopQRstUVWxyz
ABCDEFGHIJKLMNOpqrstuVWXYZ	ABe fCDghi JKLMNOPqrstuvWXYZ