
Sprawozdanie 3

Piotr Krześniak (303133)

1. Pozyskiwanie i wyświetlanie metadanych pliku

1. Proszę przejrzeć manual do funkcji z rodziny stat(2). Czym różnią się te funkcje?

Funkcje systemowe stat(2), fstat(2), lstat(2) zwracają informacje o podanym pliku. Do uzyskania tych informacji nie musimy mieć dostępu do pliku, lecz — w przypadku stat() i lstat() jedynie prawa wykonania (eXec).

2. Jakie są standardowe deskryptory otwierane dla procesów?

- stat() zwraca status pliku wskazywanego przez path
- lstat() jest identyczny ze stat(), lecz w przypadku gdy path jest dowiązaniem symbolicznym, to zwraca status tego dowiązania, a nie pliku, do którego się to dowiązanie odwołuje.
- fstat() jest identyczny ze stat(), jednak różni się tym, że informacje na temat pliku jest uzyskiwana przez deskryptor pliku.

3. Proszę przeczytać opis struktury stat, w szczególności proszę zwrócić uwagę na pole st_mode

```
struct old_stat {  
    unsigned short st_dev; /*numer urządzenia zawierający dany i-wezeł*/  
    unsigned short st_ino; /*numer i-wezła*/  
    unsigned short st_mode; /*16-bitowy tryb pliku*/  
    unsigned short st_nlink; /*licznik dowiązań*/  
    unsigned short st_uid; /*identyfikator właściciela pliku*/  
    unsigned short st_gid; /*identyfikator grupy*/  
    unsigned short st_rdev; /*numer urządzenia związane z plikiem specjalnym*/  
    unsigned long st_size; /*aktualna wielkość pliku*/  
    unsigned long st_atime; /*czas dostępu do pliku*/  
    unsigned long st_mtime; /*czas modyfikacji pliku*/  
    unsigned long st_ctime; /*czas zmiany stanu pliku*/  
}
```

4. Co reprezentuje flaga S_IFMT zdefiniowana dla pola st_mode?

Flaga *S_IFMT* oznacza maskę bitową dla pół bitowych typu pliku. Używając pola *st_mode*, który zawiera typ pliku i jego tryb nakładamy maskę *S_IFMT* używając operatora &. Dzięki zastosowaniu tej maski na polu *st_mode* uzyskujemy dostęp do pola zawierającego typ pliku (za pomocą switcha bądź ifów, tak jak w przykładzie z manuala)

```

printf("Typ pliku:                ");

switch (sb.st_mode & S_IFMT) {
case S_IFBLK: printf("urządzenie blokowe\n"); break;
case S_IFCHR: printf("urządzenie znakowe\n"); break;
case S_IFDIR: printf("katalog\n"); break;
case S_IFIFO: printf("FIFO/pipe\n"); break;
case S_IFLNK: printf("dowiązanie symboliczne\n"); break;
case S_IFREG: printf("zwykły plik\n"); break;
case S_IFSOCK: printf("gniazdo\n"); break;
default:      printf("typ nieznany\n"); break;
}

```

5. Zmienna sb jest wypełnioną strukturą typu struct stat. Czy można sprawdzić typ pliku (np. czy plik jest urządzeniem blokowym) w następujący sposób?

```
if ((sb.st_mode & S_IFBLK) == S_IFBLK)
```

Nie, nie można, gdyż nie użyliśmy odpowiedniej maski, aby sprawdzić czy plik jest urządzeniem blokowym powinniśmy użyć sposobu pokazanego wyżej.

Poniżej zamieszczam wyniki mojego programu z kilkoma testami

```
peterczak@peterczak-VirtualBox:~/S0/LAB3$ ./a.out test.txt
File type:                regular file
Name of the file:         test.txt
I-node number:            2126673
Mode:                     755
Permissions:              read: yes, write: yes, execute: yes
Link count:               1
Ownership:                peterczak(1000), peterczak(1000)
Preferred I/O block size: 4096 bytes
File size:                164 bytes
Blocks allocated:         8
Last status change:       2 hour(s) ago
Last file access:         1 hour(s) ago
Last file modification:    2 hour(s) ago
peterczak@peterczak-VirtualBox:~/S0/LAB3$ ./a.out
a.out      bigFile.txt  stat_info.c  test.txt
peterczak@peterczak-VirtualBox:~/S0/LAB3$ ./a.out bigFile.txt
File type:                regular file
Name of the file:         bigFile.txt
I-node number:            2127161
Mode:                     755
Permissions:              read: yes, write: yes, execute: yes
Link count:               1
Ownership:                peterczak(1000), peterczak(1000)
Preferred I/O block size: 4096 bytes
File size:                27 kB
Blocks allocated:         56
Last status change:       1 hour(s) ago
Last file access:         1 hour(s) ago
Last file modification:    1 hour(s) ago
peterczak@peterczak-VirtualBox:~/S0/LAB3$ touch fileTest.txt
peterczak@peterczak-VirtualBox:~/S0/LAB3$ ./a.out fileTest.txt
File type:                regular file
Name of the file:         fileTest.txt
I-node number:            2127429
Mode:                     644
Permissions:              read: yes, write: yes, execute: no
Link count:               1
Ownership:                peterczak(1000), peterczak(1000)
Preferred I/O block size: 4096 bytes
File size:                0 bytes
Blocks allocated:         0
Last status change:       7 sec(s) ago
Last file access:         7 sec(s) ago
Last file modification:    7 sec(s) ago
peterczak@peterczak-VirtualBox:~/S0/LAB3$
```