LAB3

Maryna Lukachyk

Zaawansowane operacje wejścia-wyjścia dla plików

1. Pozyskiwanie i wyświetlanie metadanych pliku

- funkcje systemowe stat(2), fstat(2), lstat(2)

Funkcje te zwracają informacje o podanym pliku. Do uzyskania tej informacji nie są wymagane prawa dostępu do samego pliku, lecz — w przypadku **stat**() i **lstat**() — konieczne są prawa wykonywania (przeszukiwania) do wszystkich katalogów na prowadzącej do pliku ścieżce path.

- **stat**() zwraca status pliku wskazywanego przez path, ładując go do argumentu buf.
- **lstat**() jest identyczny z **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 z **stat**(), z tym wyjątkiem, że plik, którego status ma zwrócić, jest określony przez deskryptor pliku fd.

- struktura stat:

```
struct stat {
    dev_t     st_dev;      /* ID urządzenia zawierającego plik */
    ino_t     st_ino;      /* numer i-węzła (inode) */
    umode_t   st_mode;     /* ochrona */
    nlink_t   st_nlink;    /* liczba dowiązań stałych (hardlinks) */
    uid_t     st_uid;      /* ID użytkownika właściciela */
    gid_t     st_gid;      /* ID grupy właściciela */
    dev_t     st_rdev;     /* ID urządzenia (jeśli plik specjalny) */
    off_t     st_size;     /* całkowity rozmiar w bajtach */
    blksize_t st_blksize;  /* wielkość bloku dla I/O systemu plików */
    blkcnt_t  st_blocks;   /* liczba zaalokowanych bloków 512-bajtowych */
    time_t    st_atime;    /* czas ostatniego dostępu */
    time_t    st_mtime;    /* czas ostatniej modyfikacji */
    time_t    st_ctime;    /* czas ostatniej zmiany */
};
```

- funkcje biblioteczne getpwuid(3), getgrgid(3), getlogin(3)

  Funkcja **getpwuid**() zwraca wskaźnik do struktury, zawierającej pola powstałe z rozłożenia tego rekordu bazy danych haseł, który odpowiada użytkownikowi o identyfikatorze *uid*.

  Funkcja **getgrgid**() zwraca wskaźnik do struktury, zawierającej pola powstałe z rozłożenia tego rekordu bazy danych o grupach, który odpowiada grupie o identyfikatorze *gid*.

  getlogin() zwraca wskaźnik do napisu zawierającego nazwę użytkownika zalogowanego na terminalu sterującym procesu lub wskaźnik null, jeśli nie można tej informacji określić.

- funkcje manipulujące czasem: time(2), strftime(3), difftime(3)

  time() zwraca czas od 1 stycznia 1970r 00:00:00 GMT, mierzony w sekundach. Jeśli t nie jest równe NULL, to zwracana wartość jest również zapisywana w pamięci wskazywanej przez t.

  strftime() zwraca ilość bajtów w char* s, w której znajduje się wskaźnik char* s na czas systemowy w sformatowanym c-stringu, formatujemy tak jak w większości języków np. "%H:%m" oznacza format HH:mm.

  difftime() zwraca liczbę sekund, które upłynęły pomiędzy czasem *time1* a *time0*, reprezentowanymi jako wartości typu *double*. Oba czasy są podane w czasie kalendarzowym, co oznacza, że jego wartość jest mierzona (w sekundach) względem epoki tj. daty 1970-01-01 00:00:00 +0000 (UTC).

- funkcja readlink(2)

  readlink() - wypisuje cele dowiązań symbolicznych lub kanoniczne nazwy plików.

Ćwiczenia:

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

Stat zwraca status pliku wskazywanego przez path, lstat w przypadku gdy path jest dowiązaniem symbolicznym zwraca status dowiązania symbolicznego, a nie pliku do którego to dowiązanie odwołuje się, w przypadku fstat plik jest opisany przez deskryptor.

2. Proszę przeczytać opis stuktury stat, w szczególności proszę zwrócić uwagę na pole st_mode:

- Co reprezentuje flaga S_IFMT zdefiniowana dla pola st_mode?

  S_IFMT 0170000 maska bitowa dla pól bitowych typu pliku

- 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) {/* plik jest urządzeniem blokowym */}
```

S_IFBLK 0060000 urządzenie blokowe

NIE możemy w taki sposób sprawdzić typ pliku.

Typ pliku można sprawdzić w następujący sposób, gdzie flaga S_IFMT sprawdzi typ pliku i jego tryb i potem porównujemy z typem który nas interesuje:

```
if ((sb.st_mode & S_IFMT) == S_IFBLK) {/* plik jest urządzeniem blokowym */}
```

```
S_IFMT     0170000   maska bitowa dla pól bitowych typu pliku
S_IFSOCK   0140000   gniazdo
S_IFLNK    0120000   dowiązanie symboliczne (symbolic link)
S_IFREG    0100000   plik regularny
S_IFBLK    0060000   urządzenie blokowe
S_IFDIR    0040000   katalog
S_IFCHR    0020000   urządzenie znakowe
S_IFIFO    0010000   kolejka FIFO
S_ISUID    0004000   bit "set-used-ID"
S_ISGID    0002000   bit "set-group-ID" (patrz niżej)
S_ISVTX    0001000   bit "sticky" (patrz niżej)
S_IRWXU    00700     maska praw dostępu właściciela pliku
S_IRUSR    00400     właściciel ma prawa odczytu
S_IWUSR    00200     właściciel ma prawa zapisu
S_IXUSR    00100     właściciel ma prawa wykonania
S_IRWXG    00070     maska praw dostępu dla grupy
S_IRGRP    00040     grupa ma prawa odczytu
S_IWGRP    00020     grupa ma prawa zapisu
S_IXGRP    00010     grupa ma prawa wykonania
S_IRWXO    00007     maska uprawnień dla innych (poza grupą)
S_IROTH    00004     inni mają prawa odczytu
S_IWOTH    00002     inni mają prawa zapisu
S_IXOTH    00001     inni mają prawa wykonania
```

2. Wejście/wyjście asynchroniczne

- funkcje systemowe: open(2)

open - otwarcie i utworzenie pliku lub urządzenia

- Funkce biblioteczne :

```
aio_read(3)      Enqueue a read request.  This is the asynchronous
                 analog of read(2).

aio_write(3)     Enqueue a write request.  This is the asynchronous
                 analog of write(2).

aio_fsync(3)     Enqueue a sync request for the I/O operations on a
                 file descriptor.  This is the asynchronous analog of
                 fsync(2) and fdatasync(2).

aio_error(3)     Obtain the error status of an enqueued I/O request.

aio_return(3)    Obtain the return status of a completed I/O request.

aio_suspend(3)   Suspend the caller until one or more of a specified
                 set of I/O requests completes.

aio_cancel(3)    Attempt to cancel outstanding I/O requests on a
                 specified file descriptor.

lio_listio(3)    Enqueue multiple I/O requests using a single function
                 call.
```

- struktura struct aiocb

- int aio_fildes - deskryptor pliku, którego dotyczy żądanie.
- off_t aio_offset - pozycja w pliku, od której ma się rozpocząć operacja.
- size_t aio_nbytes - liczba bajtów do odczytania/zapisania.
- void *aio_buf - wskaźnik do bufora zawierającego dane do zapisania lub gdzie zostaną umieszczone odczytane dane.
- struct sigevent aio_sigevent - struktura określająca sposób powiadomienia o zakończeniu żądania (**aio_sigevent.sigev_notify**)