Laboratorium 5 – Operacje na plikach

Plik w systemie LINUX identyfikowany jest przez nazwę, przy czym podawanie nazwy pliku przy każdym odwołaniu do niego wymagałoby każdorazowego przeszukiwania odpowiednich katalogów w celu ostatecznego ustalenia jego lokalizacji. W celu uniknięcia czasochłonnego przeszukiwania katalogów podczas lokalizowania pliku przy każdej operacji, wprowadzona została funkcja systemowa **open**, której zadaniem jest zaalokowanie niezbędnych zasobów w jądrze, umożliwiających wykonywanie dalszych operacji na pliku bez potrzeby przeszukiwania katalogów. Funkcja **open** zwraca deskryptor pliku.

Jądro systemu operacyjnego dostarcza również mechanizm tworzenia plików. Mechanizm tworzenia plików zwykłych dostępny jest przez funkcję systemową **creat**, która tworzy plik o nazwie podanej jako parametr aktualny i otwiera utworzony plik w trybie do zapisu, również zwracając odpowiedni deskryptor.

Tworzenie, otwieranie i zamykanie plików realizowane jest za pomocą następujących funkcji:
- `open` - otwarcie pliku (uogólniona funkcja open umożliwia również utworzenie pliku),
- `creat` - utworzenie pliku i otwarcie do zapisu,
- `close` - zamknięcie deskryptora otwartego pliku.

Operacje na plikach realizowane są za pomocą funkcji:
- `read` - odczyt fragmentu pliku,
- `write` - zapis fragmentu pliku,
- `lseek` - przesunięcie wskaźnika bieżącej pozycji.

Powyższe funkcje zdefiniowane są w pliku **fcntl.h** i **unistd.h**

`int creat( const char *pathname, mode_t mode )`
Wartości zwracane:
- poprawne wykonanie funkcji: deskryptor otwartego pliku,
- zakończenie błędne: -1 (Kody błędów zwracane są przez zmienną `errno` w przypadku błędnego zakończenie funkcji.)

Argumenty funkcji:
- `pathname` – wskaźnik do napisu zawierającego nazwę ścieżki pliku, który ma być otwarty (nazwa bezwzględna lub względna),
- `mode` – prawa dostępu (np. 0640).

Funkcja `creat` tworzy plik, którego lokalizację wskazuje parametr `pathname`. Prawa dostępu do utworzonego pliku ustawiane są zgodnie z parametrem `mode`. Jeśli plik o takiej

nazwie już istnieje a proces wywołujący funkcję `creat` ma prawo do zapisu tego pliku, to jego zawartość jest usuwana (następuje obcięcie pliku). Plik wskazywany przez `pathname` otwierany jest w trybie do zapisu.

`int open( const char *pathname, int flags[, mode_t mode] )`

Wartości zwracane:

- poprawne wykonanie funkcji: deskryptor otwartego pliku
- zakończenie błędne: -1

Argumenty funkcji:

- `pathname` – wskaźnik do napisu zawierającego nazwę ścieżki pliku, który ma być otwarty (nazwa bezwzględna lub względna)
- `flags` – metoda dostępu do pliku
  - `O_RDONLY` – otwarcie w trybie tylko do odczytu,
  - `O_WRONLY` – otwarcie w trybie tylko do zapisu,
  - `O_RDWR` – otwarcie w trybie do odczytu i do zapisu.

  Argument flags może być połączony operatorem bitowym OR z jedną (lub więcej) z następujących wartości:
  - `O_CREAT` – utworzenie pliku, jeśli plik jeszcze nie istnieje,
  - `O_TRUNC` – obcięcie pliku, jeśli plik istnieje i otwierany jest w trybie `O_WRONLY` lub `O_RDWR`,
  - `O_EXCL` – powoduje zgłoszenie błędu, jeśli plik już istnieje i otwierany jest z flagą `O_CREAT`
  - `O_APPEND` – operacje pisania odbywają się na końcu pliku.
- `mode` – prawa dostępu, argument opcjonalny.

Parametr wejściowy `pathname` jest nazwą pliku, parametr wejściowy `flags` oznacza tryb otwarcia pliku i może mieć następujące wartości: `O_RDONLY`, `O_WRONLY`, `O_RDWR` . Dodatkowo w trybie zapisu możliwe jest użycie flagi `O_APPEND`, która jest sumowana bitowo z `O_WRONLY` lub `O_RDWR` i powoduje, że zapis wykonywany jest zawsze na końcu pliku.

`int close( int fd )`

Wartości zwracane:

- poprawne wykonanie funkcji: 0
- zakończenie błędne: -1

Argumenty funkcji:

- `fd` – deskryptor zamykanego pliku

Funkcja `close` zamyka deskryptor pliku przekazany przez parametr `fd`. Po zamknięciu pliku zwalniana jest pozycja w tablicy deskryptorów i może ona zostać ponownie wykorzystana przy otwarciu kolejnego pliku, czyli nowo otwarty plik może otrzymać ten sam deskryptor, który miał plik wcześniej zamknięty.

`int read( inf fd, void *buf, size_t count )`

Wartości zwracane:

- poprawne wykonanie funkcji: *rzeczywista liczba bajtów, jaką udało się odczytać*
- zakończenie błędne: -1

Argumenty funkcji:

- `fd` – deskryptor pliku, z którego mają zostać odczytane dane,
- `buf` – adres bufora, do którego zostaną przekazane dane odczytane z pliku,
- `count` – ilość bajtów do odczytania

Funkcja `read` powoduje odczyt `count` bajtów z otwartego pliku, identyfikowanego przez deskryptor `fd`, począwszy od bieżącej pozycji wskaźnika do pliku i umieszczenie ich pod adresem `buf`. Funkcja zwraca liczbę bajtów na której udało się wykonać operację (**zero oznacza koniec pliku**). Odczyt powoduje zmianę wskaźnika bieżącej pozycji w pliku. Po otwarciu pliku wskaźnik ten ustawiony jest na 0, czyli na początek pliku, a po kolejnych operacjach przesuwa się w kierunku końca pliku o liczbę bajtów, które udało się odczytać.

`int write( inf fd, void *buf, size_t count )`

Wartości zwracane:

- poprawne wykonanie funkcji: *rzeczywista liczba bajtów, jaką udało się zapisać,*
- zakończenie błędne: -1.

Argumenty funkcji:

- `fd` – deskryptor pliku, do którego mają zostać zapisane dane,
- `buf` – adres bufora, z którego zostaną pobrane dane,
- `count` – ilość bajtów do zapisania

Funkcja `write` powoduje zapis `count` bajtów do otwartego pliku, identyfikowanego przez deskryptor `fd`, począwszy od bieżącej pozycji wskaźnika do pliku i umieszczenie ich pod adresem `buf`. Funkcja zwraca liczbę bajtów na której udało się wykonać operację. Zapis również powoduje zmianę wskaźnika bieżącej pozycji w pliku.

`long lseek( inf fd, off_t offset, int whence )`

Wartości zwracane:

- poprawne wykonanie funkcji: *nowa bieżąca pozycja wskaźnika*,
- zakończenie błędne: -1.

Argumenty funkcji:

- `fd` – deskryptor otwartego pliku,
- `offset` – liczba bajtów, o jaką należy przesunąć wskaźnik,
- `whence` – parametr określający pozycję względem której jest przesuwany wskaźnik.

Funkcja `lseek` powoduje zmianę wskaźnika bieżącej pozycji w otwartym pliku o deskryptorze `fd`. Nowa pozycja jest bajtem o numerze `offset` liczonym odpowiednio względem:

- początku pliku, gdy `whence` = SEEK_SET,
- końca pliku, gdy `whence` = SEEK_END,
- bieżącej pozycji, gdy `whence` = SEEK_CUR.

Wartość parametru `offset` < 0 oznacza przesunięcie w kierunku początku pliku (niższych indeksów), a wartość `offset` > 0 oznacza przesunięcie w kierunku końca pliku (wyższych indeksów). Funkcja zwraca aktualną wartość wskaźnika bieżącej pozycji (po przesunięciu) liczoną względem początku pliku.