
Sprawozdanie 2

Piotr Krześniak (303133)

1. Podstawy obsługi plików w systemie UNIX

1. Co to są deskryptory plików?

Jest to identyfikator otwartego pliku (zazwyczaj), przydzielony do określonego procesu. Jest on liczbą całkowitą (typu int z C). Jest używany przez wywołania systemowe, takie jak `open()`, `write()`, `read()` czy `close()`.

2. Jakie są standardowe deskryptory otwierane dla procesów?

0: Standardowe wejście (`stdin`) - `STDIN`

1: Standardowe wyjście (`stdout`) - `STDOUT`

2: Standardowe wyjście diagnostyczne (`stderr`) - `STDERR`

3. Jakie flagi trzeba ustawić w funkcji `open` aby otrzymać funkcjonalność funkcji `creat`?

Potrzebujemy flag: **`O_WRONLY`, `O_CREAT`, `O_TRUNC`**

4. W wyniku wykonania polecenia `umask` otrzymano 0022. Jakie prawa dostępu będzie miał plik otwarty w następujący sposób?

`Umask` – polecenie interfejsu POSIX, które odpowiada za ograniczanie praw dostępu do plików.

- `S_IRWXU` (Read Write eXecute User) 00700 - wszystkie uprawnienia dla użytkownika
- `s_IRWXG` (Read Write eXecute Group) 00070 - wszystkie dla grupy
- `S_IRWXO` (Read Write eXecute Others) 0007 - wszystkie dla pozostałych

Domyślne prawa dostępu wylicza się ze wzoru: W naszym przypadku prawa przy tworzeniu pliku są ustawione na 777 oraz `umask` na 0022. W związku z tym $777 \& \sim 022$ (logiczny AND) czyli $111-111-111 \& 111-101-101 = 111-101-101 = \mathbf{755}$

5. Co oznaczają flagi: `O_WRONLY` | `O_CREAT` | `O_TRUNC`? Co oznacza flaga `O_APPEND`?

- **`O_WRONLY`** *write only*, tylko do zapisu,
- **`O_RDONLY`** *read only*, tylko do odczytu,
- **`O_RDWR`** do zapisu i do odczytu,
- **`O_APPEND`** połączenie `write`'a i funkcji `lseek`, idziemy na koniec pliku i dopisujemy treść

6. Co oznacza zapis: `S_IRUSR` | `S_IWUSR`?

- **`S_IRUSR`** - **Read USeR** - uprawnienie odczytu przez właściciela
- **`S_IWUSR`** - **Write USeR** - uprawnienie zapisu przez właściciela

2. Operacje pisania i czytania z pliku

1. Czy w momencie powrotu z funkcji write dane są już zapisane na urządzenie wyjściowe?

Nie, system operacyjny sam decyduje kiedy zapisze dane, może np. trzymać dane w buforze.

2. Co robi funkcja writeall? Jakiej sytuacji dotyczy wartość EINTR?

Funkcja ta zapisuje podaną ilość bajtów (nbate), jednak gdy zapis był powstrzymany przerwaniem jest kontynuowany dla następnych bajtów. Dzieje się tak za pomocą słowa kluczowego continue.

3. Wskaźnik pliku i sygnalizator O_APPEND

1. Dwa deskryptory: fd1 i fd2 użyto do otwarcia pliku podając tę samą ścieżkę dostępu do pliku. Wskaźnik pliku ustawiony jest na początku pliku. Następnie korzystając z deskryptora fd1 wykonano operację zapisania 100b do pliku. Następnie przy użyciu deskryptora fd2 wykonano operację czytania z pliku. Pytanie: Na jakiej pozycji jest wskaźnik pliku? Jakie dane odczytano przy użyciu fd2?

Dwa deskryptory mają swoje własne, osobne wskaźniki w pliku, które przemieszczają się w nim niezależnie od siebie. Z fd2 odczytano to co zapisano przy użyciu fd1

2. Do otwarcia pliku użyto jednego deskryptora fd3. Następnie wykonano kolejno operację pisania 100b i czytania 100b. Na jakiej pozycji jest wskaźnik pliku? Co zostało przeczytane?

Wskaźnik jest na pozycji 200, nic nie zostało przeczytane. Wskaźnik dla jednego deskryptora jest stały i jego położenie nie zeruje się przy wykonywaniu innej operacji.

3. Czy każdorazowe poprzedzenie operacji pisania ustawieniem wskaźnika pliku na końcu pliku za pomocą funkcji lseek daje taki sam rezultat jak otwarcie pliku w trybie z ustawioną flagą O_APPEND?

Tak, jeśli używamy flagi O_APPEND to korzysta z funkcji lseek do ustawienia sobie wskaźnika na koniec pliku przed każdą operacją.

4. Jak wygląda wywołanie funkcji lseek które:

- ustawia wskaźnik na zadanej pozycji:
`lseek(deskryptor, pozycja, SEEK_SET);`
- znajduje koniec pliku
`lseek(deskryptor, 0, SEEK_END);`
- zwraca bieżącą pozycję wskaźnika:
`lseek(deskryptor, 0, SEEK_CUR);`