YAUHENIYA PADBIAROZSKAYA

308295

Lab 2

1. Podstawy obsługi plików w systemie UNIX
   - Co to są deskryptory plików?

     To identyfikator używany przez system operacyjny do obsługi operacji wejścia/wyjścia, w standardzie POSIX deksryptor pliku jest liczbą całkowitą a tablica deskryptorów plików jest odrębna dla każdego procesu.
   - Jakie są standardowe deskryptory otwierane dla procesów?
     - 0 – STDIN
     - 1 – STDOUT
     - 2 – STDERR
   - Jakie flagi trzeba ustawić w funkcji open aby otrzymać funkcjonalność funkcji creat?

```
int creat(const char *pathname, mode_t mode);
  int open(const char *pathname, int flags, mode_t mode);
```

int flags – określa sposób otwierania pliku – ustawiamy na O_CREAT | O_WRONLY | O_TRUNC
O_CREAT : jesli plik nie istnieje bedzie stworzony
O_WRONGLY: otwórz tylko dla zapisu
O_TRUNC: jesli plik juz istnieje obetnij go

   - W wyniku wykonania polecenia umask otrzymano 0022. Jakie prawa dostępu będzie miał plik otwarty w następujący sposób: open(pathname, O_RDWR | O_CREAT, S_IRWXU | S_IRWXG | S_IRWXO)

     Przy tworzeniu plik będzie miał prawa mode NAND umask:
mode: S_IRWXU | S_IRWXG | S_IRWXO (wszystkie uprawnienia dla użytkownika | grupy | reszty) – 0777
umask: 0022
prawa = mode – umask = 0755 (user ma wszystkie uprawnienia, grupa – odczyt i wykonywanie, pozostali – odzczyt i wykonywanie)
   - opisane w punkcie wyżej
   - O_APPEND -- poczatkowo oraz przed kazdym pisaniem wskaznik do pliku bedzie ustawiany na jego koncu
   - S_IRUSR | S_IWUSR -- uprawnienie odczytu przez właściciela i uprawnienie zapisu przez właściciela
2. Operacje pisania i czytania z pliku
   - Czy w momencie powrotu z funkcji write dane są już zapisane na urządzenie wyjściowe?

     Nie, system operacyjny sam decyduje kiedy zapisze dane.
   - Co robi funkcja writeall() ? Jakiej sytuacji dotyczy wartość EINTR?

     EINTR -- przerwanie wywołania funkcji. Funkcja writeall() przepisuje liczbę *nbyte* z bufora do pliku, jeśli funkcja zostaje przerwana, kontunuuje zapis z miejsca zatrzymania.
3. Wskaźnik pliku i sygnalizator O_APPEND
   - Na jakiej pozycji jest wskaźnik pliku? Jakie dane odczytano przy użyciu fd2?

Deskryptory fd1 i fd2 mają swoje własne wskaźniki w pliku. Przemieszczenie wskaźnika 1. nie ma wpływu na pozycję wskaźnika 2. Zatem przy użyciu fd2 odczytamy dane zapisane za pomocą fd1.

- 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?

W tym przypadku mamy do czynienia z jednym deskryptorem, a znaczy z jednym wskaźnikiem. Kolejna operacja zaczyna się od miejsca, w którym skończyła działanie pierwsza. A więc wskaźnik znajduje się na pozycji 200, nic nie zostanie przeczytane.

- 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, bo O_APPEND używa lseek do ustawienia wskaźnika na koniec pliku przed każdą operacją.

- Jak wygląda wywołanie funkcji lseek które:

i.ustawia wskaźnik na zadanej pozycji?

```
lseek(fd, pozycja, SEEK_SET)    SEEK_SET- początek pliku
```

ii.znajduje koniec pliku?

```
lseek(fd, 0, SEEK_END)
```

iii.zwraca bieżącą pozycję wskaźnika?

```
lseek(fd, 0, SEEK_CUR)
```

4. Buforowanie operacji I/O

1)F funkcje (np. fread) wskaźnik na strukturę typu FILE mają bardziej skomplikowaną implementację, są trochę wolniejsze od funkcji systemowych. Ale są bardziej bezpieczne, zawsze zapisują do buforowanego strumienia, to funkcje ze standardowej biblioteki C.

2)Przy kopiowaniu za pomocą bufora, rozmiar którego jest znacznie mniejszy niż rozmiar pliku, operacja zajmuje znacznie więcej czasu. Więc gdy zmnieniamy rozmiar bufora z 1 na 512, widać poważne zmiany w czasie wykonania operacji kopiowania.