

Laboratorium 2

Programy do wykorzystania: `lab2.tar.gz`

Zadanie 1. Prawa dostępu

Napisz funkcję, która tłumaczy prawa dostępu do pliku zapisane w postaci binarnej na prawa zapisane w postaci symbolicznej. Prawa do pliku pobierz za pomocą funkcji `stat()`.

Zadanie 2. Informacja o pliku

- Zapoznaj się z programem `fileinfo.c` z pliku `lab3.tar.gz`. Uzupełnij program o podawanie nazwy pliku i wszystkich czasów związanych z plikiem.
- Zmodyfikuj program tak, aby informacja podawana była w bardziej przyjazny sposób:

```
File Name:
File Type: (np. zwykły, katalog, ...)
Mode: rw-rw-r-- (ósemkowo)
Links:
UID: ania (UID liczbowy)
GID: studenci (GID liczbowy)
Size:
Accessed: (data ostatniego dostępu)
Modified: (data ostatniej modyfikacji)
Changed: (data ostatniej modyfikacji i-węzła)
```

Przykładowe rozwiązanie: opracuj funkcje:

- `void mode_to_type(int mode, char str[])` - funkcja pobiera wartość pola `mode` i zwraca w tablicy `str` opis typu pliku (`regular`, `directory`, `character special`, `block special`, `symbolic link`, `FIFO`, `Unix domain socket`)
- `void mode_to_perm(int mode, char str[])` - funkcja pobiera wartość pola `mode` i zwraca w tablicy `str` prawa dostępu zapisanych symbolicznie
- `char *uid_to_name(uid_t uid)` - funkcja zwraca wskaźnik do nazwy użytkownika
- `char *gid_to_name(gid_t gid)`

Zadanie 3. Dostęp do pliku

Napisz program, który sprawdza, czy proces może czytać, zapisywać lub uruchamiać plik w zależności od podanego w wywołaniu parametru ('r', 'w', 'x').

Uwaga: uwzględnić fakt, że prawa dostępu są definiowane dla trzech kategorii: `user`, `group`, `other`.

Zadanie 4. Informacja o pliku

- Sprawdź jak działa polecenie `ls`. Zapoznaj się ze wstępną wersją tego polecenia `ls1` z pliku `lab3.tar.gz`. Jakie są podstawowe różnice między tymi dwoma wykazami?
- Wykorzystaj funkcje z poprzednich zadań, tak aby twój program wyświetlał pełny opis plików, jeśli zostanie podana opcja `-l` (podobnie jak opcja `-l` w poleceniu `ls`).

Zadanie 5. Kopiowanie plików

- a) Sprawdź jak działa polecenie `cp`.
- b) Zapoznaj się ze uproszczoną wersją polecenia (plik `cp1.c`).
- c) Jak sprawdzić czy utworzony plik jest kopią pliku źródłowego? Jakie prawa dostępu otrzymał utworzony plik? Czy program dobrze obsługuje błędy? Spróbuj na przykład skopiować nieistniejący plik lub skopiować plik do katalogu. Jakie inne błędy mogłyby wystąpić? Zapoznaj się z opisem użytych w programie funkcji systemowych i spróbuj wygenerować inne błędy
- d) Program w obecnej postaci nadpisuje istniejący plik. Uzupełnij program o obsługę opcji `-i` pozwalającej użytkownikowi podjąć decyzję, czy chce napisać istniejący plik.

Zadanie 6. Ścieżka dostępu do polecenia

Napisz własną wersję polecenia `which`. Polecenie to działa w oparciu o zmienną `PATH`. Pobiera nazwę programu do wykonania i wyświetla jego pełną nazwę ścieżkową, o ile zostanie znaleziona w jednym z katalogów zmiennej `PATH` (zobacz działanie polecenia `which`). Aby informacja ta była wyświetlona, dodatkowo musi być spełniony warunek, że użytkownik, który wydał polecenie `which` ma prawo do wykonywania tego programu.

Wskazówka: funkcje do wykorzystania `getenv()`, `geteuid()`, `getegid()`, `stat()`.

Zadania domowe

Zadanie 7

Polecenie `tee` czyta ze standardowego wejścia do napotkania końca pliku, zapisuje kopię wejścia na standardowe wyjście i do pliku podanego w wierszu wywołania. Domyślnie polecenie `tee` nadpisuje ten plik, jeśli istnieje. Napisz program, który

- a) działa zgodnie z domyślnym zachowaniem polecenia `tee`
- b) obsługuje opcję `-a` plik, która powoduje dopisanie wczytywanych danych do końca pliku, jeśli ten istnieje.

Zadanie 8

Napisz program, który jest uproszczoną wersją polecenia `du`:

- a) działa tylko we wskazanym katalogu,
- b) wyświetla albo wielkości poszczególnych plików, albo ich sumę
- c) wielkości mogą być wyświetlane w różnych jednostkach (patrz polecenie `du`) w zależności od podanej opcji

Zadanie 9

Napisz program, który przegląda wybrany katalog wraz z podkatalogami i wyświetla informację o liczbie i procentowym udziale plików różnych typów (zwykle, katalogi, dowiązania symboliczne, itp.). Do przeglądania katalogu użyj funkcji bibliotecznej `nftw()`.

Zadanie 10

Napisz program, który monitoruje i rejestruje zmiany wielkości pliku przez jedną godzinę. Po zakończeniu monitorowania program powinien pokazać prosty histogram pokazujący wszystkie zmiany w tym czasie.