

ДЗЗ. Му_ср.

2 октября 2021 г.

Написать аналог ср:

`ср file1 file2`

копирует первый файл во второй. Содержимое второго затирает. Если второго нет, создает.

Флаги:

1) `-h` (`-help`): вывести принтфом подробную справку (как `man`) по вашей проге и выйти из нее сразу же, ничего не скопировав.

2) `-v` (`-verbose`): после того как скопировали вывести в консоль, что сделали, например:

```
"2.txt> "3.txt"
```

это если было `ср 2.txt 3.txt`

3) `-i` (`-interactive`): если файл в который пишем уже существует, спросить перед тем как в него записать: "вы уверены?". Если юзер ответит `yes` (ответ считывает например `scanf` или `cin`), то продолжаем, если по то прогу завершаем, не копируя.

4) `-f` (`-force`): если у нас нет прав на запись, удалить файл `нахрен`, создать новый с тем же именем и в него записать. `foren` кстати если нет файла с таким именем еще создает новый как раз. Без флага `-f` если нет прав `ср` не должно записать, разумеется.

Проверять права с помощью переменной `errno`. Эта переменная лежит в библиотеке `<errno.h>`, значит в проге ее можно объявить так: `extern int errno`. В ней хранится код последней ошибки. Вам нужно проверить, что этот код связан именно с правами доступа (этот код - 6 букв большими буквами, если что, ищите=)). Зная, что это за код, флаг `force` пишется буквально в 4 короткие строчки: открыть файл, проверить ошибку, если ошибка доступа, то стереть файл `нахрен` (`remove("1.txt")`) и создать новый (`open("1.txt "w")`).

5) -p (-preserve): не только данные о файле скопировать, но и метаданные (st_mode, st_atime, st_mtime из стата, а также uid и gid). St_mode с помощью chmod, uid и gid - с помощью chown, время доступа и модификации: создать структуру

```
struct utimbuf Time;
```

соответствующим полям в ней присвоить соответствующие поля из стата копируемого файла.

Потом с помощью функции utime файлу в который копируем присвоить эту структуру.