POLITECHNIKA
OPOLSKA

PROGRAMOWANIE SYSTEMOWE

Podstawowe funkcje systemowe do obsługi plików.

Ćwiczenie Nr 1

Proszę napisać program wykorzystujący elementarne operacje Uniksa dostępu do plików realizujący następujące zadania:

1. Program musi być uruchamiany z dwoma argumentami z linii komend. W przeciwnym wypadku powinien poinformować o błędzie i zakończyć się.
2. Sprawdzić, czy istnieje plik o nazwie „**tmp.txt**” i czy jest on możliwy do odczytania. Jeżeli plik nie istnieje lub proces nie ma dostępu do odczytu, wówczas program powinien o tym poinformować i zakończyć się.
3. Wprowadzony pierwszy argument należy potraktować jako nazwę pliku. Należy sprawdzić czy plik o takiej nazwie istnieje:
   - jeśli istnieje:
   należy sprawdzić prawo zapisu do pliku. Jeśli proces nie ma prawa zapisu do tego pliku wówczas poinformować o tym i zakończyć program. Jeśli proces ma prawo zapisu do tego pliku wówczas należy go nadpisać.
   - jeśli nie istnieje:
   należy stworzyć plik z prawami zapisu i odczytu przez wszystkich użytkowników (0666)
4. Zapisać do pliku w pierwszej linii: „zadanie 1:”
   Odczytać 3, 5, 9 linię tekstu z pliku „tmp.txt” i wypisać je na ekranie oraz przepisać je do nowego pliku.
5. Zapisać do pliku w nowej linii: „zadanie 2:”
   Odczytać 2, 4, 7 linię tekstu z pliku „tmp.txt”, zamienić wszystkie małe litery na duże litery oraz przepisać je do nowego pliku.
6. Zapisać do pliku w nowej linii: „zadanie 3:”
   Przepisać wszystkie cyfry występujące w pliku „tmp.txt” do nowego pliku.
7. Zapisać do pliku w nowej linii: „zadanie 4:”
   Porównać tekst z pliku „tmp.txt” z drugim argumentem i zapisać do nowego pliku położenie identycznych ciągów znaków.
8. Wypisać na ekranie wielkość pliku.
9. Zmienić prawa do pliku na (0600).
10. Zamknąć otwarte pliki.

przydatne funkcje: **https://linux.die.net/**

```
#include <fcntl.h>
int open(const char *pathname, int flags, mode_t mode);
int creat(const char *pathname, mode_t mode);

#include <unistd.h>
int access(const char *pathname, int mode);
int close(int fd);
int unlink(const char *pathname);
ssize_t read(int fd, void *buf, size_t count);
ssize_t write(int fd, const void *buf, size_t count);


#include <stdio.h>
void perror(const char *s);


#include <ctype.h>
int toupper(int c);
int tolower(int c);
int isdigit(int c);


#include <sys/stat.h>
int chmod(const char *path, mode_t mode);
mode_t umask(mode_t mask);
int stat(const char *pathname, struct stat *statbuf);


#include <string.h>
size_t strlen(const char *s);
int strcmp(const char *s1, const char *s2);
int strncmp(const char *s1, const char *s2, size_t n);
char *strcpy(char *dest, const char *src);
```