Adam Stec

Systemy operacyjne

Systemy plików – Laboratorium 06

1. Treść zadania
   Zaprojektować system plików, który będzie zapisywał zadane słowa bitowe do plików i manipulował nimi w odpowiedni sposób, wykorzystując funkcje określone przez prowadzącego. System ma również działać w systemie minix.

2. Koncepcja wykonania zadania – od strony teoretycznej
   a) Od strony sprzętowej:
   - Wzoruję się na systemie FAT;
   - Zakładam maksymalny rozmiar pliku 4096 bajtów, a pojedynczego bloku 64 bajty;
   - Kolejne pliki będą alokowane jako ciąg bitów;
   - Plik będzie opisany deskryptorem (DTF) zawierającym następujące elementy:
     - nazwę pliku;
     - blok pamięci, na którym się zaczyna;
     - całkowity rozmiar pliku.
   - Nad zapisem pliku w wielu blokach czuwa tablica FAT;
   - Superblok zajmujący 64 bajty,

   b) Będę potrzebował funkcji:
   - Tworzącej nowy plik o rozmiarze 0 bajtów;
   - Zapisującej do pliku pewne słowo bitowe. W przypadku nie istnienia zadanego pliku, tworzy go, a następnie zapisuje dane. Gdy rozmiar danych jest większy od dostępnego miejsca w pliku, wyrzucić wyjątek i przerwać działanie;
   - Czytającej dane z pliku;
   - Usuwającej plik o zadanej nazwie;
   - Kopiującej plik z dysku na minixa
   - Kopiującej plik z minixa do dysku

3. Opis zastosowanych funkcji

- void initValues(void); - służy do inicjowania wartości początkowy w superbloku, DTF oraz FAT;
- int createDisc(void); - tworzy nowy, pusty dysk. Zwraca ERROR_FILE_CREATING gdy tworzenie sie nie powiedzie
- void loadSuperBlock(FILE* disc);
  void loadDTF(FILE* disc);
  void loadFAT(FILE* disc);
  funkcje służące do odczytu danych z pliku binarnego do superbloku i tablic w celu dalszych modfikacji lub odczytów
- int doesFileExist(char* const name); - funkcja sprawdza czy w DTF juz istnieje plik o podanej nazwie
- void actualizeSuperBlock(FILE* disc);

void actualizeDTF(FILE* disc);
void aztualizeFAT(FILE* disc);
funkcje służące do zapisu tablic do pliku binarnego (dysku)

- int copyOnDisc(char* const name, char* const name_after); - funkcja kopiująca plik o nazwie name do dysku pod nazwa name_after, zwraca odpowienie błędy w przypadku niepowodzenia. Aktualizuje SuperBlock, DTF oraz FAT. Jesli plik nie miesci sie w jednym bloku, dzieli go i umieszcza kolejno w wolnych na dysku blokach
- Int copyFromDisc(char* const name, char* const name_after); - funkcja kopiujaca plik z dysku o nazwie name do minixa pod nazwa name_after, zwraca odpowienie błędy w przypadku niepowodzenia. Aktualizuje SuperBlock, DTF oraz FAT.
- Void resetConnected(int DTFPosition, FILE* disc); - funkcja resetuje wszystkie dane zwiazane z plikiem na pozycji DTFPosition w DTF.
  Uzyta w deleteFile().
- Int deleteFile(char* const name); - funkcja usuwa plik o nazwie name z dysku. Czyści również ślad po pliku w data block (błednie z powodu wydłużenia czasu działania funkcji
- int showFolder(void); - funkcja pokazuje jakie pliki znajduja sie obecnie w dysku
- int resetDisc(void); - format dysku
- int deleteDisc(void); - usunięcie dysku
- int showInsides(void); - w czytelny sposob wyswietla aktualny stan superblock, DTF oraz FAT w dysku
- int readFile(char* const name); - wyświetla na stdout zawartość pliku o zadanej nazwie. W przypadku nie istnienia takiego pliku, zwraca odpowiedni błąd.
- int writeToFile(char* const name, char* const dataToWrite); - zapisuje zawartość dataToWrite do pliku o nazwie name. W przypadku nie istnienia tego pliku, tworzy go i zapisuje dane. Gdy zadany plik istnieje, dopisuje do niego dataToWrite, uwzględniając sytuację, gdy dopisywane dane nie zmieszczą się w pojedynczym bloku.