

SOI Laboratorium nr 6 – System plików

Sprawozdanie

Kuba Forczek

Program został napisany w języku C (standard C89, aby zachować zgodność z systemem Minix). Rozwiązanie zostało podzielone na następujące nagłówki oraz pliki źródłowe:

- *filesystem.h*, *filesystem.c* – odpowiedzialne za wykonywanie operacji powiązanych z systemem plików
- *gui.h*, *gui.c* – zajmujące się pobieraniem od użytkownika danych wejściowych oraz wyboru rodzaju operacji, jak również częściowym zabezpieczeniem przed wprowadzeniem niepoprawnych danych
- *main.c* – główny punkt wejścia programu, w funkcji *main()* wywoływana jest jedynie działająca w pętli metoda *main_menu()* z „*gui.h*”.

Z uwagi na ograniczoną ilość zasobów systemu Minix, podjęto decyzję, by przechowywane na wirtualnym dysku informacje niebędące faktycznymi plikami ograniczały się do absolutnego minimum. Zdefiniowane zostały struktury reprezentujące nagłówek dysku (zapisany na samym jego początku) oraz nagłówek pliku (poprzedzający każdy znajdujący się na dysku plik):

```
struct disk_header
{
    short file_count;
    off_t file_offsets[MAX_FILES];
};

struct file_header
{
    off_t file_size; /* with header */
    char file_name[MAX_FNAME_LENGTH+1];
};
```

Istotnym ograniczeniem przyjętego rozwiązania jest górna granica możliwej liczby plików na wirtualnym dysku, niezależna od ilości niewykorzystanego miejsca. Zlikwidowanie tego ograniczenia wymagałoby jednak dynamicznej zmiany rozmiaru nagłówka pliku, co nie tylko skomplikowałoby kod przy okazji każdego odczytu i zapisu tej struktury, jak również wymagałoby ewentualnego przeniesienia danych znajdujących się bezpośrednio za nią.

Po uruchomieniu programu, użytkownikowi ukazuje się następująca lista opcji:

```
SOI Lab 6 - Filesystem
MAIN MENU

1 - Create virtual disk
2 - Open virtual disk
3 - Copy file to virtual disk
4 - Get file from virtual disk
5 - Delete file from virtual disk
6 - List files on virtual disk
7 - Print memory info
8 - Defragment virtual disk
9 - Delete virtual disk
e - Exit program

Make a choice: >
```

Operacje na dysku wymagają jego otwarcia, zatem przykładowo wywołanie opcji 3 – kopiowanie pliku na dysk, nie będzie dozwolone:

```
Make a choice: > 3
Disk must be opened first!
Press ENTER to continue
```

Procedura tworzenia dysku wymaga podania ścieżki do pliku, który zostanie utworzony oraz pożądanego rozmiaru dysku (należy pamiętać, że część zostanie zajęta przez nagłówek dysku oraz nagłówki kopiowanych później plików). Przykładowo:

```
Make a choice: > 1
The disk will be saved in a file of a given size.
Provide a path to the disk file: > /home/kuba/Documents/vdisk3
Size of the disk (in bytes): > 1024

Creating disk... Disk created!
Press ENTER to continue
```

Wprowadzenie niepoprawnych logicznie danych skutkuje wyświetleniem odpowiedniego komunikatu, oraz powrotem do menu głównego. Ograniczeniem obecnym w całym programie jest jednak wymaganie podania bezwzględnej ścieżki.

Nowo otwarty dysk może zostać otworzony. Na początku nie ma na nim żadnych plików:

```
Make a choice: > 2
Path to the disk: > /home/kuba/Documents/vdisk3

Opening disk... Disk opened!
Press ENTER to continue
```

```

Make a choice: > 6

File count: 0

INDEX |          FILE NAME          | SIZE (B)

Press ENTER to continue

```

Natomiast informacja o poszczególnych regionach pamięci prezentuje się następująco:

```

Make a choice: > 7
Region list

ADDRESS |    SIZE (B)    | TYPE
      0 |           168 | Disk header
      a8 |           856 | Free space

Total disk size: 1024 B
Occupied space: 168 B (17 %)
Free space: 856 B (83 %)

```

Teraz przykładowy plik tekstowy zostanie skopiowany na dysk.

```

Make a choice: > 3
Path to the file to be copied: > /home/kuba/Documents/example.txt

Copying file...

0% |          | 100%
|.....|
File copied!
Press ENTER to continue

```

Ponowne wyświetlenie listy plików oraz informacji o pamięci skutkuje następującym wyjściem:

```

Make a choice: > 6

File count: 1

INDEX |          FILE NAME          | SIZE (B)
    1 |          example.txt      |    80

Press ENTER to continue

```

```

Make a choice: > 7
Region list

  ADDRESS |    SIZE (B)    |  TYPE
    0 |           168 | Disk header
   a8 |           40 | File header
   d0 |           40 | File data
   f8 |          776 | Free space

Total disk size: 1024 B
Occupied space: 248 B (25 %)
Free space: 776 B (75 %)

Press ENTER to continue

```

Usuwanie pliku odbywa się poprzez usunięcie referencji do początku jego nagłówka, zatem przy istniejących np. dwóch plikach i usunięciu pierwszego z nich (w sensie kolejności ułożenia w pamięci):

```

Make a choice: > 5
How do you want to select the file?
1 - Choose from the list
2 - Type its name
> 1

File count: 2

INDEX |      FILE NAME      | SIZE (B)
  1 |      example.txt    | 80
  2 |    otherfile.txt    | 101

Choose index: > 1
Are you sure you want to delete the file from the virtual disk? This cannot be undone. (y/n) > y
Deleting file... File deleted.
Press ENTER to continue

```

wolne miejsce w pamięci będzie podzielone:

```

Make a choice: > 7
Region list

  ADDRESS |    SIZE (B)    |  TYPE
    0 |           168 | Disk header
   a8 |           80 | Free space
   f8 |           40 | File header
  120 |           61 | File data
  15d |          675 | Free space

Total disk size: 1024 B
Occupied space: 269 B (27 %)
Free space: 755 B (73 %)

```

Wówczas możliwa jest operacja defragmentacji (wywoływana również automatycznie z metody kopiującej plik na dysk, gdy skumulowanie wolnego miejsca pod jednym adresem umożliwi wykonanie pożądanej operacji). Nie jest to defragmentacja w pełnym rozumieniu tego słowa (wszakże program nie dzieli plików, jedynie zapisuje je w całości), lecz skutkuje przesunięciem istniejących danych tak, aby nie zostawiać pomiędzy nimi niezagospodarowanej przestrzeni. W ukazanym scenariuszu, defragmentacja oraz podział pamięci będzie wyglądać następująco:

```
Make a choice: > 8
Confirm defragmentation? (y/n) > y

Defragmenting...
Moving file "otherfile.txt"...

0% |                               | 100%
   |.....|
File "otherfile.txt" moved successfully.
Defragmentation successful!
Press ENTER to continue
```

```
Make a choice: > 7
Region list

ADDRESS | SIZE (B) | TYPE
-----|-----|-----
      0 |        168 | Disk header
      a8 |         40 | File header
      d0 |         61 | File data
     10d |        755 | Free space

Total disk size: 1024 B
Occupied space: 269 B (27 %)
Free space: 755 B (73 %)
```

Jak widać, choć ilość ani treść faktycznych danych na dysku nie zmieniła się (zajętych 269 bajtów, istnieje 1 plik), to wolne miejsce podzielone wcześniej na 80 i 675 bajtów, zostało skumulowane za plikiem i wynosi 755 bajtów.

Wyjście z pętli menu głównego odbywa się, gdy użytkownik wybierze opcję zakończenia działania:

```
Make a choice: > 0
Goodbye!

Process finished with exit code 0
```

W tym momencie strumień pliku reprezentującego wirtualny dysk jest zamykany, a program kończy pracę.