

Politechnika Śląska

Wydział Informatyki, Elektroniki i Informatyki

Programowanie komputerów 2

Temat projektu: Archiwizator

autor	Hubert Olszewski
prowadzący	mgr.inż. Maciej Długosz
rok akademicki	2019/2020
kierunek	informatyka
rodzaj studiów	SSI
semestr	2
termin laboratorium	piątek, 10:15 - 11:45
sekcja	61
termin oddania	19.06.2020
sprawozdania	

1 Treść zadania

Napisać program umożliwiający przechowywanie struktury katalogów oraz zawartych w nich plików w jednym binarnym pliku archiwum. Program powinien mieć możliwość tworzenia i usuwania katalogów z archiwum, a także dodawania do niego plików z dysku oraz ich usuwania. Ponadto powinna istnieć możliwość przeglądania zawartości archiwum oraz wyodrębniania zarchiwizowanych plików na dysk. Poziom zagłębienia katalogów może być dowolny.

Uwagi techniczne Katalog ma podwieszone dwie listy: plików i katalogów potomnych (z kolejnymi listami katalogów i plików).

2 Analiza zadania

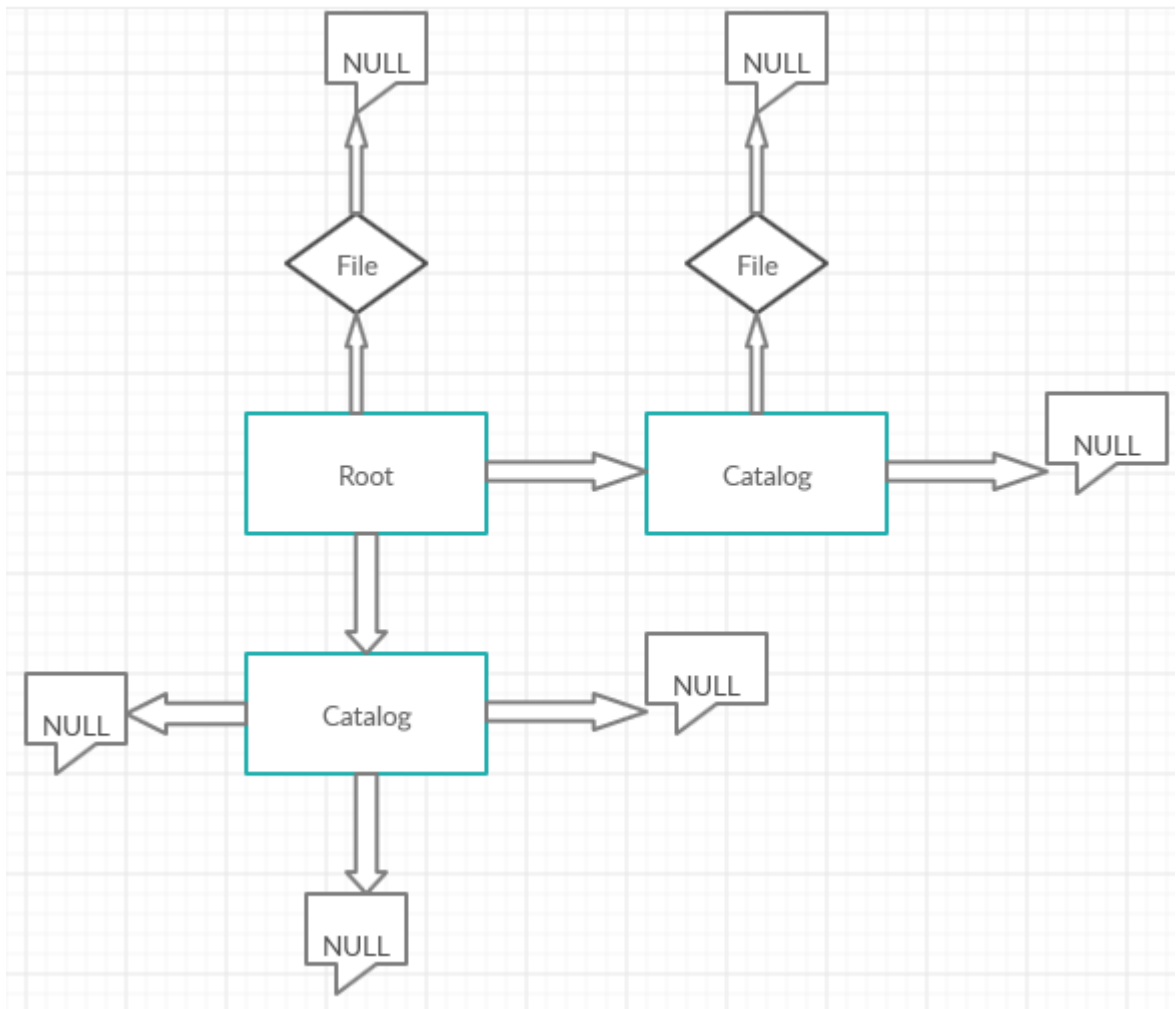
Zagadnienie przedstawia problem dodawania odczytywanych danych z plików do odpowiadającym im struktur, wykonywania operacji na tych danych oraz zapisywania wyniku operacji w jednym pliku binarnym, który przechowuje wprowadzone zmiany.

2.1 Struktury danych

W programie wykorzystano strukturę wielopoziomową. W pierwszej kolejności jest drzewo binarne przeznaczone do przechowywania informacji o katalogach. Drzewo binarne przechowuje dane w węzłach. Węzeł może mieć od 0 do 2 potomków, przy czym po lewej stronie węzła znajdują się potomki przechowujące katalogi potomne, natomiast po prawej stronie węzła znajdują się katalogi będące na równym poziomie zagnieżdżenia. Dodatkowo każdy węzeł posiada również listę jednokierunkową, w której przechowywane są dane z odczytanych plików. **Rysunek 1** przedstawia przykład takiej struktury wielopoziomowej. Taka struktura danych pozwala na łatwe nawigowanie pomiędzy katalogami i plikami.

2.2 Algorytmy

Program dodaje do drzewa binarnego nazwy katalogów oraz ścieżki, które zawierają drogę do niego. Sposób dodawania takiego węzła do drzewa jest określany na podstawie ścieżki, która jest podawana przez użytkownika. Ścieżka ta jest nawigatorem, dzięki któremu pliki są umieszczane do odpowiednich katalogów. Do listy jednokierunkowej przechowującej dane plików, elementy dodawane są w sposób posortowany według nazwy pliku. Wypisanie wielopoziomowej struktury jest realizowane poprzez rekurencyjne przejście przez węzły drzewa oraz iteracyjnie dla każdej listy jednokierunkowej. Zwolnienie pamięci wykonuje się w sposób przejścia do ostatnich elementów struktury.



Rysunek 1. Przykład struktury wielopoziomowej przechowującej katalogi oraz pliki.

3 Specyfikacja zewnętrzna

Program jest uruchamiany z linii poleceń. Są trzy możliwe otwarcia programu.

- Uruchomienie programu bez żadnych argumentów lub z błędnymi argumentami spowoduje zakończenie działania i wyświetlenie krótkiej pomocy np.

Archiwizator.exe x1 x2

- Po przekazaniu do programu pierwszego argumentu po przełączniku **-o** spowoduje pominięcie odczytywania pliku i udzielenie dostępu do menu programu np.

Archiwizator.exe -o

- W momencie przekazania argumentów w następującej kolejności, nastąpi archiwizacja podanych plików do podanego istniejącego katalogu w **archiwum**, a następnie uzyskamy dostęp do menu programu np.

Archiwizator.exe katalog1/katalog2 plik1.txt plik2.bin

4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs od logiki aplikacji.

4.1 Ogólna struktura programu

W funkcji głównej wywoływana jest funkcja **PobierzArchiwum**. Funkcja ta zaczytuje do struktury zawartość pliku binarnego **Archiwum**, w przypadku, jeśli nie istnieje to wywoływana jest funkcja **UtwórzArchiwum**, która je tworzy i kończy program, w przeciwnym wypadku następuje wywołanie funkcji **CzytajParametry**, która sprawdza podane argumenty i w zależności od nich wykonuje się jeden z przypadków opisanych w specyfikacji zewnętrznej.

4.2 Opis typów i funkcji

Szczegółowy opis typów i funkcji został opisany w załączniku.

5 Testowanie

Program został przetestowany na różnego rodzaju plikach oraz na różnego rodzaju argumentach przy uruchamianiu. Argumenty nie poprawne (nie zawierające oczekiwanego formatu, niezgodne ze specyfikacją) powodują zgłoszenie błędu oraz wypisanie stosownego komunikatu.

Program został sprawdzony pod kątem wycieków pamięci przy użyciu funkcji z biblioteki zewnętrznej **visual leaks detector**.

6 Wnioski

Zrealizowałem zadanie, które polegało na napisaniu programu do archiwizacji danych. Program nie jest prosty, lecz nie jest również bardzo skomplikowany. Napisanie go wymagało umiejętności samodzielnego zarządzania pamięcią oraz zaznajomienia się z funkcjami biblioteki `string.h`, czyli obsługi łańcuchów znakowych. Dość wymagające okazało się usunięcie wycieków pamięci oraz odpowiednie przekazywanie typów zmiennych między funkcjami. Wiele razy w zły sposób przekazywałem argumenty do różnych funkcji, co sprawiało błędy w odczytywaniu danych. Wiele czasu musiałem zainwestować w znalezienie i naprawienie tych problemów, dzięki temu zdobyłem potrzebną wiedzę, którą wykorzystam aby nie powielać takich błędów w przyszłości.