

Projekt I z przedmiotu „Systemy Operacyjne”

Demon synchronizujący dwa podkatalogi

Autorzy:

- Awramiuk Piotr
- Szorc Michał

Opis:

Program uruchamia się przez podanie co najmniej dwóch argumentów, czyli ścieżki źródłowej i ścieżki docelowej do podkatalogów. Demon domyślnie porównuje zwykłe pliki z katalogu źródłowego. Porównywane są nazwy plików oraz daty ich modyfikacji. Jeżeli plik nie został odnaleziony w katalogu docelowym to jest on kopiowany, a gdy daty modyfikacji się różnią, to plik jest nadpisywany ze zmianą daty. Jeśli w katalogu docelowym znajduje się plik, którego brakuje w źródłowym, to jest on usuwany.

Funkcjonalność programu można rozbudować podając następujące parametry:

- **-R** - rekurencyjne kopiowanie podkatalogów znajdujących się w katalogu źródłowym,
- **-T [czas w sekundach]** - zmianę czasu uśpienia demona,
- **-S [rozmiar w KB]** - ustalenie granicy pomiędzy dużym a małym plikiem.

Demon może zostać natychmiastowo wybudzony poprzez wysłanie sygnału SIGUSR1 używając polecenia **kill -10 [numer procesu]**. Wszystkie informacje o akcjach wykonanych przez demona zostają zapisane w logach systemowych. Zależnie od wielkości pliku program wykorzystuje dwie metody do kopiowania plików, pierwsza opierająca się o funkcje read/write daje optymalne wyniki przy małych plikach, druga natomiast, przeznaczona do dużych plików, wykorzystuje mmap/write.

Instrukcja:

1. Znajdując się w katalogu z plikami źródłowymi należy wpisać w terminalu polecenie **make**, które utworzy pliki wykonawcze.
2. W celu testu można uruchomić skrypt (najpierw nadając mu uprawnienia wpisując polecenie **chmod 777 testscript** a następnie wpisując **./testscript** w terminalu). Utworzy on dwa przykładowe podkatalogi. Za ich pomocą możliwe jest przetestowanie funkcjonalności programu.
3. W terminalu należy wpisać **./SYNCdaemon**, następnie można podać wybrane parametry, a na końcu należy podać pełne ścieżki do katalogów. Przykładowe wywołanie:

```
./SYNCdaemon -R -T 10 -S 1024  
/home/user/Desktop/Awramiuk_Szorc_Projekt_Daemon/SourceDirectory  
/home/user/Desktop/Awramiuk_Szorc_Projekt_Daemon/TargetDirectory
```

Demon przy tym wywołaniu będzie synchronizował katalogi co 10 sekund, rekurencyjnie sprawdzał podkatalogi, natomiast granica małego pliku to 1024KB (1MB).

Funkcje:

- **void** copybigfile (**char*** path1, **char*** path2) – funkcja kopiująca duży plik. Jako argumenty dostaje ścieżkę do pliku źródłowego oraz ścieżkę do pliku docelowego.
- **void** copyfile (**char*** path1, **char*** path2) – funkcja kopiująca mały plik. Jako argumenty dostaje ścieżkę do pliku źródłowego oraz ścieżkę do pliku docelowego.

- **void** modtime (**struct** stat sb1, **char*** targetpath) – funkcja odpowiedzialna za zmianę daty modyfikacji. Jako argumenty otrzymuje strukturę stat (przechowującą m.in. informację o dacie modyfikacji) pliku znajdującego się w katalogu źródłowym oraz ścieżkę do pliku docelowego.
- **char*** getpath (**char*** path, **char*** filename) – funkcja odpowiedzialna za łączenie dwóch dynamicznych tablic znaków, zwraca pełną ścieżkę do pliku. Jako argumenty otrzymuje nazwę pliku oraz ścieżkę do katalogu, w którym ten plik się znajduje.
- **void** searchSTT (**char*** sourcepath, **char*** targetpath, **long int** S) – funkcja odpowiedzialna za porównywanie plików w katalogach oraz ewentualne ich kopiowanie. Jako argumenty otrzymuje ścieżkę do katalogu źródłowego, ścieżkę do katalogu docelowego oraz liczbę reprezentującą granicę między dużym a małym plikiem.
- **void** searchSTTDIR (**char*** sourcepath, **char*** targetpath, **long int** S) - funkcja działająca rekurencyjnie, odpowiedzialna za porównywanie plików, katalogów oraz ewentualne ich kopiowanie. Jako argumenty otrzymuje ścieżkę do katalogu źródłowego oraz ścieżkę do katalogu docelowego.
- **void** searchTTS (**char*** sourcepath, **char*** targetpath) – funkcja odpowiedzialna za porównywanie plików w katalogach oraz ewentualne ich usuwanie. Jako argumenty otrzymuje ścieżkę do katalogu źródłowego, ścieżkę do katalogu docelowego oraz liczbę reprezentującą granicę między dużym a małym plikiem.
- **void** Remove (**char*** path) – funkcja odpowiedzialna za usuwanie całej zawartości w katalogu. Jako argument otrzymuje ścieżkę do katalogu, z którego ma zostać usunięta cała zawartość.
- **void** searchTTS DIR (**char*** sourcepath, **char*** targetpath) - funkcja działająca rekurencyjnie, odpowiedzialna za porównywanie plików, katalogów oraz ewentualne ich usuwanie. Jako argumenty otrzymuje ścieżkę do katalogu źródłowego oraz ścieżkę do katalogu docelowego.

Porównanie funkcji kopiujących:

Testy zostały przeprowadzone na pliku o wielkości 1GB.

```
Apr 23 17:23:41 musztarda Daemon[1852]: Time needed to copy the file using copyfile: 2.035263
Apr 23 17:23:41 musztarda Daemon[1852]: Copied file duzy_plik.txt to /home/musztarda/Pulpit/Projekt
/TargetDirectory using read/write copying method

Apr 23 17:24:50 musztarda Daemon[1879]: Time needed to copy the file using copyfile: 1.952411
Apr 23 17:24:50 musztarda Daemon[1879]: Copied file duzy_plik.txt to /home/musztarda/Pulpit/Projekt
/TargetDirectory using read/write copying method

Apr 23 17:27:02 musztarda Daemon[1931]: Time needed to copy the file using copyfile: 1.996923
Apr 23 17:27:02 musztarda Daemon[1931]: Copied file duzy_plik.txt to /home/musztarda/Pulpit/Projekt
/TargetDirectory using read/write copying method
```

Rysunek 1. Testowe pomiary czasu kopiowania metodą read/write

```
Apr 23 17:08:32 musztarda Daemon[1584]: Time needed to copy the file using copybigfile: 1.665492
Apr 23 17:08:32 musztarda Daemon[1584]: Copied file duzy_plik.txt to /home/musztarda/Pulpit/Projekt
/TargetDirectory using mmap copying method

Apr 23 17:09:56 musztarda Daemon[1621]: Time needed to copy the file using copybigfile: 1.480738
Apr 23 17:09:56 musztarda Daemon[1621]: Copied file duzy_plik.txt to /home/musztarda/Pulpit/Projekt
/TargetDirectory using mmap copying method

Apr 23 17:11:13 musztarda Daemon[1639]: Time needed to copy the file using copybigfile: 1.520883
Apr 23 17:11:13 musztarda Daemon[1639]: Copied file duzy_plik.txt to /home/musztarda/Pulpit/Projekt
/TargetDirectory using mmap copying method
```

Rysunek 2. Testowe pomiary czasu kopiowania metodą mmap/write

Średni czas kopiowania pliku metodą read/write wynosił 1.99s, natomiast metodą mmap/write 1.56s. Zastosowanie funkcji mmap pozwala na niewielkie przyspieszenie procesu kopiowania. Jest to szczególnie widoczne przy dużych plikach.