

Dokumentacja

Krótki opis działania programu

Program przyjmuje kod w pseudoassemblerze, bądź kod maszynowy, a następnie symuluje wykonanie programu na jeden z podanych sposobów:

- „Krok po kroku” z wykorzystaniem interfejsu graficznego (tylko dla kodu w pseudoassemblerze),
- **Od razu**, bez wykorzystania interfejsu graficznego

Po czym zwraca plik wynikowy w postaci kodu maszynowego z naniesionymi wynikami jego działania. (W trybie „krok po kroku” program daje możliwość niezwracania pliku wynikowego)

Środowisko programistyczne

Środowisko programistyczne, z którego korzystałem to Visual Studio 2019.

Zgodność działania programu z założeniami

- Cały kod pseudoassemblera **powinien** zostać zinterpretowany przez program w dokładnie taki sam sposób, w jaki było to zakładane w opisie projektu zaliczeniowego - „Interpreter Pseudoassemblera”,
- Plik wejściowy w postaci kodu pseudoassemblera:
 - **musi** przestrzegać zasad dotyczących podziału szerokości kolumn,
 - **nie może** zawierać komentarzy zajmujących całą linię przed sekcją danych, bądź przed sekcją rozkazów oraz komentarzy od razu bez spacji po jakiejś komendzie/definicji zmiennej, np.: (LR 1, 2//komentarz),
 - **maksymalny** odstęp między kolejnymi liniami kodu to jedna pusta linia,
 - odstęp między sekcją danych, a sekcją rozkazów **musi** być równy jednej pustej linii.
- Plik wejściowy w postaci kodu maszynowego **musi** przestrzegać zasad składni (jedna linia to 4, bądź 8 cyfr, przy czym każde dwie cyfry oddzielone jedną spacją, sekcja danych oddzielona od sekcji rozkazów jedną pustą linią) **oraz** po ostatniej linijce kodu **nie powinno** być żadnych pustych linii,
- Odstępy między kolejnymi elementami jednego wiersza (np. między rodzajem rozkazu i jego argumentami) **powinny** być dokonane tylko i wyłącznie za pomocą tabulatora, bądź spacji,
- Format pliku wejściowego **powinien** wyglądać dokładnie tak, jak zostało to podane w opisie projektu zaliczeniowego,
- Program **można** uruchomić przez cmd jednym, dwoma, bądź trzema argumentami:
 - Pierwszy argument oznacza ścieżkę względną do pliku wejściowego,
 - Drugi argument oznacza, czy plik wejściowy jest w postaci kodu w pseudoassemblerze, czy też kodu maszynowego,

- Trzeci argument oznacza, czy program ma zostać uruchomiony w trybie „krok po kroku”, czy też nie w tym trybie

Obsługa danych argumentów przez program powinna być taka sama, jak było to zakładane w opisie projektu zaliczeniowego - „Interpreter Pseudoassemblera”.

- Program po zakończonym prawidłowym działaniu wyrzuca plik wynikowy .txt, będący wynikiem symulacji kodu w pseudoassemblerze, bądź kodu maszynowego, do tego samego folderu, w którym umieszczony jest plik .exe

- **Jedynymi błędami obsługiwanymi przez program** to te związane z nieprawidłową nazwą pliku wejściowego, bądź niepodaniem tej nazwy w argumencie przez cmd,

- W interfejsie graficznym **strzałka** przy sekcji rozkazów wskazuje na rozkaz, który wykona się po naciśnięciu przycisku przejścia do następnego kroku

Symbolika kolorów w interfejsie graficznym

Biały – sekcja danych i sekcja rozkazów

Czerwony – rejestry

Niebieski – komórka, której zawartość uległa zmianie wskutek wykonania poprzedniego rozkazu

Zielony – dostępne komendy

Dostępne komendy w interfejsie graficznym

- [p] – Następny krok
- [s] – Wyjdź i zapisz aktualny stan symulacji
- [q] – Wyjdź bez zapisywania

Jak uruchomić program z folderu Exe?

- Klikając dwukrotnie prawym przyciskiem myszy plik .exe:

W folderze Exe, obok pliku .exe należy umieścić plik wejściowy .txt, z którego chcemy interpretować kod napisany w pseudoassemblerze, bądź kod maszynowy. Następnie należy otworzyć plik .exe i w wyniku tego na ekranie powinna pojawić się konsola. Program poprosi o podanie nazwy pliku (np. jeżeli nazwaliśmy nasz plik wejściowy „pseudoassembler”, to musimy wpisać „pseudoassembler.txt”), a następnie po wpisaniu nazwy program powinien zakończyć działanie. Wówczas w folderze Exe obok pliku .exe i naszego pliku .txt powinien się pojawić plik wynikowy o nazwie pliku wejściowego z dopiskiem „_out” zawierający interpretację kodu z pliku wejściowego na kod maszynowy z naniesionymi wynikami jego działania.

- Komendą przez cmd:

W folderze Exe, obok pliku .exe należy umieścić plik wejściowy .txt, z którego chcemy interpretować kod napisany w pseudoassemblerze, bądź kod maszynowy. Następnie w cmd, będąc w ścieżce do folderu Exe, należy uruchomić plik .exe z argumentami, z jakimi chcemy go uruchomić, po czym zależnie od argumentów program uruchomi się w trybie „krok po kroku”, bądź nie w tym trybie. Po wykonaniu programu, w folderze Exe obok pliku .exe i naszego pliku .txt powinien się pojawić plik wynikowy o nazwie pliku wejściowego z dopiskiem „_out” zawierający interpretację kodu z pliku wejściowego na kod maszynowy z naniesionymi wynikami jego działania.

Zalecenia

- Przy działaniu interfejsu graficznego zaleca się uruchomienie konsoli w trybie pełnoekranowym,
- Nie zaleca się uruchamiania programów wykorzystujących wyjątkowo dużą ilość pamięci

Krótki opis przykładów pseudoassemblerowych

W folderze „Przykłady” znajdują się przykłady działania programu składające się z plików typu: „nazwa.txt” i nazwa_out.txt”, które oznaczają kolejno plik wejściowy i wynikowy przykładu.

- baca.txt – program tworzący słowo „BACA”
- potegowanie.txt – program podnoszący liczbę 3 do potęgi 15