0.ścieżka, uruchamianie i przykłady

ścieżka nie przewiduje wykorzystania znaków polskich

ścieżka podawana jest względem pliku .exe

w trzecim etapie program można uruchomić wyłącznie z poziomy konsoli gdy chcemy uruchomić go dla kodu maszynowego, lub z możliwością włączenia opcji debug, w przeciwnym przypadku można go uruchomić w środowisku graficznym.

w trzecim etapie pierwszym i jedynym argumentem programu może być ścieżka względna do pliku tekstowego zawierającego program w pseudoassemblerze **ALBO** pierwszym argumentem jest ścieżka do pliku w podanym formacie(pseudoassemblera bądź kodu maszynowego) a drugim parametr specyfikujący ten typ, odpowiednio msck\_code dla kodu maszynowego bądź psa\_code dla kodu pseudoassemblera **ALBO** pierwszym argumentem jest ścieżka do pliku w podanym formacie(pseudoassemblera bądź kodu maszynowego) a drugim parametr specyfikujący ten typ, odpowiednio msck\_code dla kodu maszynowego bądź psa\_code dla kodu pseudoassemblera, trzecim argumentem jest debug co powoduje odpalenie programu w trybie krok po kroku

np: pseudoassembler-interpreter.exe ..\\przyklady\dec\_to\_bin.txt

np: pseudoassembler-interpreter.exe ..\\przyklady\minmax.txt psa\_code

np: pseudoassembler-interpreter.exe ..\\przyklady\dec\_to\_bin.txt msck\_code

np: pseudoassembler-interpreter.exe ..\\przyklady\hexex.txt msck\_code debug

np: pseudoassembler-interpreter.exe ..\\przyklady\dec\_to\_bin.txt psa\_code debug

opis przykładów:

hexex.txt: Program realizuje to samo zadanie co dec\_to\_bin.txt tylko jest zapisany w kodzie maszynowym.

dec\_to\_bin.txt: Program wczytuje naturalną liczbę dziesiętną, a następnie zapisuje ją w tablicy w postaci naturalnego kodu binarnego w tablicy od końca.

factorial.txt: Program liczy silnię z 10 i zapisuje ją w pamięci.

errtest.txt: Celowo błędny przykład w którym występuje przepełnienie typu integer.

1.program wejściowy:

sekcja danych oddzielona od sekcji rozkazów dokładnie jedną linią zawierającą dokładnie jeden znak "\n"

wolna linia pomiędzy sekcjami jest jedyną wolną linia jaką przewiduje się w pliku wejściowym

komentarze zajmujące całą linijkę muszą zaczynać się od znaków "//" na samym początku linii(kolumnie 0)

komentarze nie mogą być wprowadzane bezpośrednio przed sekcją rozkazów ani bezpośrednio po sekcji danych

wolna kolumna na początku linii oznacza brak etykiety w danym wierszu

nie zakłada się komentarzy w pliku w kodzie maszynowym

2.wyjście programu

Program zwraca w przypadku kodu maszynowego plik „var.txt” zawierający wyniki obliczeń(wypełnioną sekcję danych oraz końcowe stany rejestrów **ponieważ pisałem program przed ustaleniem nowej specyfikacji**)

Program zwraca w przypadku kodu pseudoassemblera 2 pliki „output.txt” oraz „var.txt” gdzie pierwszy zawiera podany kod w postaci maszynowej a drugi zawiera wyniki obliczeń te same co w przypadku kodu maszynowego.

Plik var.txt zawiera wartości losowe w miejscu niezadeklarowanej pamięci.

Program przerywa działanie i zwraca błąd w przypadku wyjście poza sekcję rozkazów w instrukcjach jump.

Program przerywa działanie i zwraca błąd w przypadku wyjście poza sekcję danych w instrukcjach operujących na danych.

Program przerywa działanie i zwraca błąd w przypadku podzielenia przez zero.

Program przerywa działanie i zwraca błąd w przypadku przepełnienie zmiennej int.

Jeżeli program napotka błąd automatycznie zakończy wykonanie i zwróci komunikat o błędzie.

Program w przypadku skutecznego wykonania zwaca w konsoli „exec\_end”

3. tryb krok po kroku

Jeżeli program napotka błąd automatycznie zakończy wykonanie i zwróci komunikat o błędzie.

Do realizacji została użyta biblioteka nscurses.

Zakładane są domyślne wymiary okna wiersza komend:

Wysokość nie mniejsza niż 33 linie

Szerokość nie mniejsza inż. 125 kolumn.

Kolumny są opisane.

Jeżeli program jest podany w wersji kodu pseudoassemblera to w oddzielnej kolumnie wypisywany jest również kod pseudoassemblera.

Do przejścia do kolejnego rozkazu służy klawisz „x”

Do przewijania w górę dół(jeżeli któraś sekcja jest dłuższa niż 30 linii) służą:

„w” i „s” do przewijania sekcji rozkazów odpowiednio w górę i w dół

„e” i „d” do przewijania sekcji danych odpowiednio w górę i w dół

„r” i „f” do przewijania sekcji psudoassemlera odpowiednio w górę i w dół

Wiersze w kolumnach danych i rozkazów są numerowane.

Kolorystyka:

Kolor żółty służy do zaznaczenia rozkazu który zostanie wykonany **jako następny** zarówno w sekcji rozkazów jak i w rejestrze stanu programu jak i w kodzie pseudoassemblera(jeżeli wystepuje).

Kolor czerwony służy do zaznaczenia komórki sekcji danych do której odwołuje się obecny rozkaz oraz drugiego półbajtu drugiego bajtu rozkazu gdy wskazuje on na sekcję danych.

Kolor zielony służy w instrukcji jump do zaznaczenia 1 lub 3 ostatnich bajtów instrukcji na którą wskazuje jump oraz sekcji danych samej instrukcji jump wraz z drugim półbajtem drugiego bajtu instrukcji jump.

Kolor niebieski służy w instrukcji rejestr-pamięć(innej niż jump) do zaznaczenia wykorzystywanego rejestru.

Kolor zielony i niebieski służą w instrukcji rejestr-rejest do zaznaczenia dwóch półbajtów drugiego bajtu i odpowiadających mu rejestrów. Jeżeli odwołanie jest do tego samego rejestru wykorzystywany jest tylko kolor zielony.