Autor: Mateusz Krupa

Nr indeksu: 256280

Sprawozdanie

Organizacja i Architektura Komputerów

1. Opis zadania do wykonania

Do wykonania na laboratoria był kalkulator dla liczb rzeczywistych.

Program powinien zapytać użytkownika o podanie dwóch liczb rzeczywistych, operacji do wykonanie (+ - * /) oraz sposobu zaokrąglania wyniku FPU.

Program należało napisać w języku assemblera w architekturze 32-bitowej na platformę Linux.

Założenia:

- do wczytywania / wyświetlania można używać funkcji z libc (printf, scanf)
- liczby wpisywane są w sposób dziesiętny, np. 1.23456789
- wynik wyświetlamy też w sposobie dziesiętnym
- sposoby zaokrąglania FPU (wg FPU control word): nearest (even if tie), down, up, to zero

2. Opis wykonania

W moim programie do komunikowanie użyłem funkcji scanf i printf, na początku zainicjalizowałem zmienne oraz komunikaty które będą wykorzystywane.

```
.data
    float: .asciz "%f"
    result: .asciz "%f\n"
    string: .ascii "%s\0"
    n1: .float 0.0
    n2: .float 0.0
    choice: .float
    qadd: .ascii "1. Dodawanie\n\0"
    qsub: .ascii "2. Odejmowanie\n\0"
    qmul: .ascii "3. Mnozenie\n\0"
    qdiv: .ascii "4. Dzielenie\n\0"

    qn1: .ascii "Podaj pierwsza liczbe: \0"
    qn2: .ascii "Podaj druga liczbe: \0"
```

Wszystkie zmienne są typu double poza zmienną która służy do określenia o jaką operację chodzi użytkownikowi stwierdziłem że na zmiennej typu string będzie mi wygodniej użyć mnemonika cmp.

W dalszej części programu użyłem printf do wydrukowaniu w terminalu wiadomości.

```
push %edx
push %eax
push $qadd
call printf
add $12, %esp
```

Aby jednak printf działał prawidłowo bez naruszenia pamięci musiałem na stos przekazać rejestr edx oraz eax a na koniec przesunąć wskaźnik stosu.

Do wczytywania warości użyłem scanf.

```
pushl $n2
push $float
call scanf
addl $8, %esp
```

W assemblerze kolejność w jakiej się podaje jest na odwrót jak w C jednak w moim przypadku gdy używałem tylko jednej zmiennej nie musiałem się tym przejmować. Wartości wczytywane przechowuję na stosie więc aby nie wystąpił błąd naruszenia pamięci na koniec przesuwam wskaźnik stosu.

Menu programu to po prostu wypisanie opcji jakie może użyc aktor.

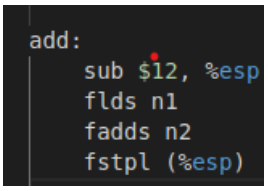
```
cmpb $49, choice
je add
# 2
cmpb $50, choice
je sub
# 3
cmpb $51, choice
je mul
# 4
cmpb $52, choice
je div
```

Moja zmienna jest typu string mogę po prostu porównać wartość kodu ascii w adresie w jakim przedtem umieściłem wybór użytkownika.

Ze względu na to że wartości na jakich działam to liczby rzeczywiste to nie mogę wyniku po prostu wyświetlić za pomocą scanf, ponieważ wyświetli mi tylko wartość całkowitą.

```
movl (%esp), %eax
movl 4(%esp), %edx
call pr
addl $12, %esp
```

Do wykonywania operacji użyłem podstawowych instrukcji arytmetycznych dla liczb zmiennoprzecinkowych do ładowania i pobierania i ściągania ze stosu.



fld – ładuje wartość na stos

fstp – zapisuje a następnie usuwa wartość ze stosu

W kolejnej części kodu za pomocą control word wykonałem zaokrąglenie.

3. Wnioski

Podczas laboratorium nauczyłem się lepiej działać na stosie. Same wywoływanie funkcji printf i scanf sprawiło mi wiele trudności. Dla liczb całkowitych wszystko działało poprawnie jednak dla rzeczywistych wyświetlało tylko cześć całkowitą aby to naprawić musiałem dodatkowo użyć stosu który też sprawił mi problem ponieważ zapomniałem na koniec przesunąć wskaźnik stosu co powodowało błąd naruszenia pamięci. Dobranie odpowiednich wartości aby zaokrąglenie działało poprawnie również zajęło mi chwilę jednak po dłuższym zastanowieniu się udało się.