

# Architektura Komputerów 2 – Laboratoria

## Laboratorium nr.3

Piotr Stachnio 241268

### Cel laboratoriów:

Celem laboratoriów było stworzenie programu w języku assembly x86\_32 który wykonywał podstawowe operacje arytmetyczne(dodawanie, odejmowanie, mnożenie, dzielenie) na liczbach zmiennoprzecinkowych pojedynczej oraz podwójnej precyzji. Drugim zadaniem było stworzenie sytuacji w której zostaną utworzone wszystkie wyjątki standardu IEEE-754.

### Zadanie nr 1

Kalkulator liczb zmiennoprzecinkowych

```
6  .data
7
8  zeroF: .float 0.0
9  liczbaDF: .float 1.1
10 liczbaUF: .float -2.2
11
12 zeroD: .double 0.0
13 liczbaDD: .double 1.1
14 liczbaUD: .double -2.2
15
16 .text
17
18 .global _start
19 _start:
20
21     finit
22
23     calc:
24     # floaty
25     fld liczbaDF      # ładuje liczbę do st0
26     fadd liczbaUF     # dodaje liczbę do st0
27
28     fld liczbaDF
29     fsub liczbaUF     # rozkaz analogiczny do fadd
30
31     fld liczbaDF
32     fmul liczbaUF
33
34     fld liczbaDF
35     fdiv liczbaUF
36
37     # double
38
39     fld liczbaDD      # operacje analogicznie do wcześniejszych ale dla double
40     fadd liczbaUD
41
42     fld liczbaDD
43     fsub liczbaUD
44
45     fld liczbaDD
46     fmul liczbaUD
47
48     fld liczbaDD
49     fdiv liczbaUD
50
```

Przed rozpoczęciem wykonywania obliczeń musimy zainicjalizować jednostkę zmiennoprzecinkową, do tego właśnie celu służy nam rozkaz `finit`. FPU(float point unit) pracuje na swoich własnych rejestrach. Jest ich dokładnie 8 (ST0-ST7), ponadto są one zbudowane w sposób podobny do stosu. Dlatego można używać na nich operacji `pop(fstp)` czy też `push(fld)`, warto zauważyć że szczytem stosu jest rejestr ST) do którego trafiają wszystkie pushowane dane.

## Zadanie nr2

Zadanie nr2 polegało na stworzeniu sytuacji w których zostaną utworzone wyjątki standardu IEEE-754

```
51  exc:
52      finit
53      # float exceptions
54      fld zeroF
55      fmul liczbaDF
56
57      fld zeroF
58      fmul liczbaUF
59
60      fld liczbaDF
61      fdiv zeroF
62
63      fld liczbaUF
64      fdiv zeroF
65
66      fld zeroF
67      fdiv zeroF
68
69      # double exceptions
70      fld zeroD
71      fmul liczbaDD
72
73      fld zeroD
74      fmul liczbaUD
75
76      fld liczbaDD
77      fdiv zeroD
78
79      fld liczbaUD
80      fdiv zeroD
81
82      fld zeroD
83      fdiv zeroD
84
```

Zrzut ekranu z gdb dla wyjątków float:

```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from calc...
(gdb) break 52
Breakpoint 1 at 0x8049063: file calc.s, line 52.
(gdb) run
Starting program: /home/cyta/Desktop/lab3AK/calc

Breakpoint 1, _exc () at calc.s:52
52      finit
(gdb) n
54      fld zeroF
(gdb) n
55      fmul liczbaDF
(gdb) n
57      fld zeroF
(gdb) p $st0
$1 = 0
(gdb) n
58      fmul liczbaUF
(gdb) n
60      fld liczbaDF
(gdb) p $st0
$2 = -0
(gdb) n
61      fdiv zeroF
(gdb) n
63      fld liczbaUF
(gdb) p $st0
$3 = inf
(gdb) n
64      fdiv zeroF
(gdb) n
66      fld zeroF
(gdb) p $st0
$4 = -inf
(gdb) n
67      fdiv zeroF
(gdb) n
70      fld zeroD
(gdb) p $st0
$5 = -nan(0xc000000000000000)
(gdb)
```

Zrzut ekranu z gdb dla wyjątków double:

```
(gdb) n
71      fmul liczbaDD
(gdb) n
73      fld zeroD
(gdb) p $st0
$6 = -0
(gdb) n
74      fmul liczbaUD
(gdb) n
76      fld liczbaDD
(gdb) p $st0
$7 = -0
(gdb) n
77      fdiv zeroD
(gdb) n
79      fld liczbaUD
(gdb) p $st0
$8 = -inf
(gdb) n
80      fdiv zeroD
(gdb) n
82      fld zeroD
(gdb) p $st0
$9 = -nan(0xc000000000000000)
```

# Makefile

```
all: calc

calc: calc.o
    ld -m elf_i386 -o calc calc.o

calc.o: calc.s
    as --32 --gstabs -o calc.o calc.s

clean:
    rm -rf calc.o calc
```

Logika pliku makefile nie zmieniła się względem poprzednich laboratoriów.

## Problemy napotkanie podczas realizacji zadania:

Największym napotkanym przeze mnie problemem było zrozumienie jak działają rozkazy jednostki zmiennoprzecinkowej, na szczęście z pomocą przyszedł internet. Ze zrozumieniem działania tej jednostki pomogły mi następujące źródła:

- [https://www.freebsd.org/doc/en\\_US.ISO8859-1/books/developers-handbook/x86-fpu.html](https://www.freebsd.org/doc/en_US.ISO8859-1/books/developers-handbook/x86-fpu.html)
- [https://sourceware.org/binutils/docs-2.22/as/i386\\_002dFloat.html#i386\\_002dFloat](https://sourceware.org/binutils/docs-2.22/as/i386_002dFloat.html#i386_002dFloat)

## Wnioski:

Poznałem zasady działania jednostki zmiennoprzecinkowej działającej w architekturze x86\_32