lab_9 - Instrukcja do ćwiczenia

Teoria:

http://galaxy.agh.edu.pl/~amrozek/AK/lab9.pdf

oraz materiały z wykładu

Błędy w obliczeniach zmiennoprzecinkowych:

Teoretycznie wartość wyrażenia $\sqrt{x} * \sqrt{x} - x$ jest równa 0 dla dowolnych wartości ***x***. W praktyce, ze względu na ograniczoną dokładność reprezentacji liczb zmiennoprzecinkowych oraz ze względu na błędy zaokrągleń przy wykonywaniu obliczeń, pojawia się pewien błąd (końcowa wartość nie jest równa **0**). Błąd ten zależy zarówno od parametrów systemu, w którym dokonywane są obliczenia (np. precyzja wybranej reprezentacji **float/double**), jak i wartości ***x*** (dla pewnych wartości błąd może być mniejszy, dla innych większy).

Praktyka (lab_9a.s, lab_9b.s):

Działania:

1. Testujemy działanie programu lab_9a.s – służy on do sprawdzenia działania jednostki x87 po wystąpieniu sytuacji wyjątkowych (**exceptions**) w trakcie obliczeń. Szczegółowe informacje dotyczące wyjątków w jednostce x87 można znaleźć w materiałach wykładowych, dokumentacji producenta oraz w innych źródłach (Internet).
2. Sprawdzamy działanie programu bez modyfikacji domyślnego trybu pracy jednostki x87 charakteryzującego się pomijaniem ewentualnych sytuacji wyjątkowych.
3. CL (Compile, Link) – polecenie: **gcc –Wa,--defsym=CASE_1=1 –no-pie –o lab_9a lab_9a.s**
4. R (Run) – polecenie: **./lab_9a**
5. Efekt uzyskany po uruchomieniu wygląda następująco:

```
buba@buba-PC:~/AK/l9$ gcc -Wa,--defsym=CASE_1=1  -no-pie -o lab_9a lab_9a.s
buba@buba-PC:~/AK/l9$ ./lab_9a
x87 Control Register = 037F
x87 Control Register = 037F
x87 Status Register  = 0041
The end
```

6. Sprawdzamy pozostałe przypadki zmieniając definiowany symbol kolejno na CASE_2, CASE_3 oraz CASE_4 – za każdym razem ewidentne błędy w kodzie nie są w żaden sposób sygnalizowane.
7. Usuwamy komentarz przed linią:

```
#       and $EXCEPTIONS, %ax    # clear exceptions masks
```

8. CL (Compile, Link) – polecenie: **gcc –Wa,--defsym=CASE_1=1 –no-pie –o lab_9a lab_9a.s**
9. Tym razem efekt uzyskany po uruchomieniu wygląda następująco:

```
buba@buba-PC:~/AK/l9$ gcc -Wa,--defsym=CASE_4=1  -no-pie -o lab_9a lab_9a.s
buba@buba-PC:~/AK/l9$ ./lab_9a
x87 Control Register = 037F
x87 Control Register = 0340
Floating point exception (core dumped)
```

10. Sprawdzamy pozostałe przypadki zmieniając definiowany symbol kolejno na CASE_2, CASE_3 oraz CASE_4 – za każdym razem błędy w kodzie skutkują wystąpieniem wyjątku.

11. Możliwe jest ustawienie selektywnej reakcji na sytuacje wyjątkowe poprzez ustawienie bitów **b0** do **b5** w rejestrze sterującym (Control Register) jednostki x87 – stosowne instrukcje znajdują się poniżej instrukcji opisanej w punkcie 7.
12. Przechodzimy do programu lab_9b.s – wyznacza on wartość wyrażenia $\sqrt{x} * \sqrt{x} - x$. Odchyłka od **0.0** zależy od samego **x** (w programie **2.0**), jak i od parametrów systemu obliczeniowego: precyzji danych i metod zaokrągleń. Zmiana parametrów dokonywana jest poprzez odkomentowanie tylko jednej instrukcji **or** w ramach zmiany poszczególnych parametrów (żądane parametry wynikają z zawartości poszczególnych bitów Control Register jednostki **x87** – bity odpowiadające konkretnemu parametrowi są najpierw zerowane instrukcją **and**, a potem ustawiana jest konkretna kombinacja bitów przy pomocy instrukcji **or**).
13. CL (Compile, Link) – polecenie: **gcc –no-pie –o lab_9b lab_9b.s**
14. R (Run) – polecenie: **./lab_9b**
15. Przykladowe efekty uzyskane po uruchomieniu wyglądają następująco:

```
buba@buba-pc:~/AK/l9$ gcc -no-pie -o lab_9b lab_9b.s
buba@buba-pc:~/AK/l9$ ./lab_9b
Value = -0.000000000000000001084202
```

16. Jeżeli odkomentujemy instrukcję **`or $PREC_DOUBLE, %ax`** (i zakomentujemy tę, która wcześniej była odkomentowana), to rezultat będzie już inny:

```
buba@buba-pc:~/AK/l9$ gcc -no-pie -o lab_9b lab_9b.s
buba@buba-pc:~/AK/l9$ ./lab_9b
Value = 0.000000000000000004440892099
```

17. Podobnie, jeżeli odkomentujemy instrukcję **`or $PREC_SINGLE, %ax`** (i zakomentujemy tę, która wcześniej była odkomentowana), to rezultat będzie jeszcze inny:

```
buba@buba-pc:~/AK/l9$ gcc -no-pie -o lab_9b lab_9b.s
buba@buba-pc:~/AK/l9$ ./lab_9b
Value = -0.000000119209289550781250
```

18. Możemy też zbadać wpływ sposobu zaokrąglania na uzyskiwane rezultaty. Zmieniając tylko metodę zaokrąglania (tak samo jak wcześniej precyzję obliczeń) i pozostawiając pojedynczą precyzję obliczeń uzyskujemy następujące efekty:

dla ROUND_TE
```
buba@buba-pc:~/AK/l9$ gcc -no-pie -o lab_9b lab_9b.s
buba@buba-pc:~/AK/l9$ ./lab_9b
Value = -0.000000119209289550781250
```

dla ROUND_DN
```
buba@buba-pc:~/AK/l9$ gcc -no-pie -o lab_9b lab_9b.s
buba@buba-pc:~/AK/l9$ ./lab_9b
Value = -0.000000119209289550781250
```

dla ROUND_UP
```
buba@buba-pc:~/AK/l9$ gcc -no-pie -o lab_9b lab_9b.s
buba@buba-pc:~/AK/l9$ ./lab_9b
Value = 0.000000476837158203125000
```

dla ROUND_TZ
```
buba@buba-pc:~/AK/l9$ gcc -no-pie -o lab_9b lab_9b.s
buba@buba-pc:~/AK/l9$ ./lab_9b
Value = -0.000000119209289550781250
```

19. Z przeprowadzonych eksperymentów wynika, że największy błąd (odchyłka od wartości **0**) pojawia się dla kombinacji **PREC_SINGLE** i **ROUND_UP**.