

Autor: Mateusz Krupa

Nr indeksu: 256280

Sprawozdanie

Organizacja i Architektura Komputerów

1. Opis zadania do wykonania

Do wykonania na laboratorium było mnożenie liczb hexadecymalnych.

Program powinien zapytać użytkownika o dwie całkowite liczby. Użytkownik wpisuje liczby w notacji hexadecymalnej (szesnastkowo). Program powinien wypisać na standardowe wyjście wynik mnożenia tych dwóch liczb. Program należało napisać w języku assemblera w architekturze 32-bitowej na platformę Linux.

Założenia:

- liczby są podawane z klawiatury, a więc jako tekst, trzeba je przekonwertować do obliczeń
- podane liczby mogą być bardzo duże
- należy wykorzystać odpowiedni algorytm mnożenia dużych liczb

2. Opis wykonania

Na początku trzeba było wczytać ciąg znaków i aby to zrobić wykorzystałem `sysread`. Pierwszym krokiem jaki należało zrobić to trzeba było przekonwertować z tekstu na wartości binarne które mógłbym zapisać w rejestrze.

```
dec %eax
mov $0, %edx
movl %eax, %ebx
movl $8, %ecx
div %ecx
cmp $0, %eax
jnz empty1
add $1, %eax

empty1:
movl %eax, first_len
movl %ebx, %eax
dec %eax
mov $0, %edi
```

Powyższy kod pozwala określić długość naszej liczby. Oraz pozwoli nam wczytać liczbę od końca.

```

mov $0, %ecx
movb n1(,%eax,1), %cl
call to_number1
dec %eax
movb n1(,%eax,1), %bl
call to_number2
shl $4, %bl
add %bl, %cl
mov %cl, first(,%edi,1)
inc %edi
dec %eax
cmp $0, %eax
jge convert1

```

Dość ważny kawałek kodu znajduję się powyżej. W tym miejscu następuje przekonwertowanie oraz zapisanie wartości w rejestrze cl a następnie do 'first'. Same call to_number sprawdza czy to jest wartość 1-9 czy A-F. Całość polega na pobraniu dwóch ostatnich wartości oraz połączeniu jej w jedno, ponieważ rejestr cl i bl ma 9 bitów to po wpisaniu wartości tylko do rejestru dostalibyśmy błędny efekt więc musimy pobrać dwie cyfry i złączyć w jednym rejestrze za pomocą 'shl', czytamy wartość od końca ponieważ przyda się to w mnożeniu więc w wyniku wpisania A123 dostajemy w zmiennej 'first' 1A32. Z drugą liczbą postępujemy tak samo.

Mnożenie wykonałem metodą przez części. Pobieramy po 4 bity z jednej i drugiej zmiennej następnie je wymnażamy i zapisujemy w wyniku na określonej pozycji potem zwiększamy pozycję o jeden i dodajemy przeniesienie jeśli wystąpiło.

```

loop1:
mov first(,%ebx,4), %eax
mov second(,%esi,4), %edi
mul %edi
add %eax, result(,%ecx,4)
inc %ecx
adc %edx, result(,%ecx,4)
inc %ecx
adc $1, result(,%ecx,4)
dec %ecx
inc %ebx
cmp first_len, %ebx
jb loop2
inc %esi
cmp second_len, %esi
jl loop1

```

Na koniec pozostaje nam ponowne odpowiednie zkonwertowanie wyniku poprzez przesuwanie na bitach tak aby wyświetlić poprawny wynik.

3. Wnioski

Podczas laboratorium nauczyłem się lepiej rozumieć jak są przechowywane dane w rejestrach i jak na tych rejestrach działać aby otrzymać oczekiwany rezultat. Każde założenie sprawiło mi trudności i musiałem nad każdym posiedzieć i przemyśleć jak je wykonać. Ograniczenia w postaci rejestrów w tym przypadku wymagało poszukania w wielu źródłach aby w pełni potrafić zrealizować zadanie.