

Raport do zadania 1

1. Opis zadania:

Zadanie polegało na napisaniu programu w języku assemblera w architekturze 32-bitowej na platformę Linux, który będzie szyfrował wpisany tekst za pomocą szyfru ROT13. Program po uruchomieniu wypisuje polecenie wpisania zdania które następnie jest szyfrowane i wypisywane w następnej linii.

2. Opis implementacji:

Na początku tworzę zmienne z komunikatami dla użytkownika i z długościami tych komunikatów.

```
buf: .ascii "Podaj zdanie do zaszyfrowania: "  
buf_len = .-buf  
  
buf1: .ascii "Zaszyfrowane zdanie to: "  
buf1_len = .-buf1
```

Następnie wypisuję pierwszy komunikat dla użytkownika w którym proszę go o wpisanie zdanie które następnie ma zostać zaszyfrowane.

```
mov $SYSWRITE, %eax  
mov $STDOUT, %ebx  
mov $buf, %ecx  
mov $buf_len, %edx  
int $SYSCALL32
```

W następnym kroku czytuje wpisane zdanie i kopiuje je do rejestru ecx, a jego długość do edx. Następnie kopiuję zawartość eax do esp i zmniejszam wartość esp o jeden bit. Eax zostawiam nie zmieniony ponieważ później używam jeszcze tego rejestru do innych wypisań. Później dodaję indeks pętli w rejestrze ebp. W następnym kroku w pętli tworzę 4 instrukcję warunkowe w których sprawdzam w którym przedziale znajduje się dana litera(A-M,N-Z,a-m,n-z) i w zależności od tego albo dodaje albo odejmuje 13.

```
od_A_do_M:  
cmpb $'A', input(%ebp)  
jl koniec  
cmpb $'M', input(%ebp)  
jg od_N_do_Z  
add $13, input(%ebp)  
jmp koniec
```

Po przejściu wszystkich instrukcji warunkowych zwiększam o jeden wartość ebp i jeżeli nie jest większa od esp wracam do etykiety petla. Na koniec wypisuję drugi komunikat do użytkownika i zaszyfrowane zdanie oraz kończę program.

3. Błędne ścieżki:

Miałem problem ponieważ na początku nie dodałem etykiety na koniec instrukcji warunkowej i w momencie gdy zmieniłem literę z przedziału A-M bądź a-m to w następnym kroku były one ponownie zmieniane przez następny blok warunkowy.

4.Wnioski:

W tym zadaniu nauczyłem się tworzyć pętle oraz instrukcje warunkowe i to było najtrudniejsze.