

Ćwiczenie 3. Realizacja funkcji, przekazywanie parametrów, stos i zmienne lokalne

Celem ćwiczenia jest zapoznanie się ze sposobem realizacji funkcji, sposobów przekazywania parametrów przez stos oraz implementacji zmiennych lokalnych.

Zagadnienia do przygotowania

- sposoby adresowania stosu – rejestr SP
- interpretacja pojęcia zmiennej lokalnej, statycznej i dynamicznej

Przebieg ćwiczenia

- przygotować schemat blokowy programu podanego w załączniku w języku asemblera,
- przeprowadzić jego kompilację i linkowanie za pomocą pakietu Borland C 3.1
- sprawdzić poprawność działania programu za pomocą Turbo Debuggera
- zwrócić szczególną uwagę na zawartość i rozmiar stosu przed i po zakończeniu wykonywania funkcji

Warunki zaliczenia ćwiczenia

Zaliczenie ćwiczenia polega na demonstracji działania programu prowadzącemu.

Sprawozdanie

Sprawozdanie powinno zawierać wydruk programu z obszernymi komentarzami oraz rysunki ilustrujące zawartość stosu w chwili wywołania funkcji i rozmieszczenie zmiennych lokalnych. Zwróć uwagę na rzutowanie typów.

Literatura

Kernigham, Ritchie, *Język C*, WNT 1989

Dokumentacja pakietu Borland C 3.1

Wróbel Eugeniusz, *Asembler 8086/88*

Wacławek Roland, *ABC Asemblera*

Scanlon Leo J., *Assembler 8086/8088/80286*

Kruk Stanisław, *Język Assembler dla początkujących*

Nawrocki J. R., *Programowanie komputerów IBM PC w języku Asemblera metodą systematyczną*

Syck Gary, *Turbo Assembler : biblia użytkownika*

Laboratorium: Programowanie niskopoziomowe

Deklaracje zmiennych

```
int i,j;
char c;
char napis1[4]="ala";
char napis2[4];
char *ptr1,*ptr2;
```

Program

```
/* funkcja bez zmiennych lokalnych -
   przedstaw graficznie zawartosc stosu
   w trakcie wykonywania funkcji */

char *copy(char *p1,char *p2)
{
/* W jaki sposob sa pobierane parametry? */
while(*p2++ = *p1++);
return p2;

/* W ktorym rejestrze zwracana jest wartosc wskaźnika ? */
}
/* funkcja ze zmiennymi lokalnymi i parametrami roznego
   rozmiaru - kopiowanie z ograniczeniem max. ilosci
   elementow */

int copy2(char *src,char *dst, char ile)
{
int i=0;
while(*dst++ = *src++)
if(++i>=ile) break;
*dst=0;
return i;
}

void main(void)
{
ptr1=napis1;
ptr2=napis2;

(void) copy(ptr1,ptr2);

/* Sprawdz, cz zawartosc stosu jest taka sama jak przed
   wywolaniem funkcji */

c=copy2(ptr1,ptr2,2);

/* uwaga na niezgodnosc typow !!! */
}
```