Marcin Stępniak

Architektura systemów komputerowych
Laboratorium 8
Symulator SMS32
Tablice i Procedury

1. Informacje

1.1. Tablice

Tablice są to kontenery danych, zwykle o ustalonym rozmiarze, których poszczególne elementy dostępne są za pomocą klucza (zwykle są to elementy numeryczne zwane indeksami). W językach assemblera są dostępne instrukcje pozwalające wypełniać określone komórki pamięci wskazanymi danymi. Traktowanie określonych danych jako tablica jest tylko umowne. Do programisty należy zaprogramowanie obsługi tabel, czyli w podstawowej wersji wskazanie liczby elementów tabeli lub jej końca, a także pilnowanie, aby program nie wykroczył poza zakres tablicy podczas jej przetwarzania.

Tabela 1. Instrukcje (dyrektywy asemblera) związane z organizacją danych

Instr.	Operand	Działanie
ORG	adres	dyrektywa assemblera powodująca, że kod będzie generowany od adresu A
DB	wartość bajtu	zapisuje wartość podaną jako operand A w pamięci RAM
DB	"tekst"	zapisuje kody ASCII podanego ciągu znaków w kolejnych komórkach pamięci

Tabela 1 zawiera instrukcje potrzebne do umieszczenia danych w odpowiednich miejscach pamięci RAM. Do utworzenia tablicy należy wykorzystać wiele instrukcji *DB*. Listing 1 zawiera przykład tworzenia i wykorzystania

tablicy. W tym przykładzie mamy w tablicy 4 elementy. Pierwszy element tablicy ma adres 02. Pętla przechodzi po wszystkich elementach tablicy i kopiuje ich wartość do rejestru AL. Warunkiem stopu jest znana wartość ostatniej komórki, po której napotkaniu program rozpoczyna się od początku.

Listing 1. Definiowanie i wykorzystanie tablic

```
; tablice
   JMP   Start
   DB 84 ; pierwsza wartosc
   DB C8 ; druga wartosc
   DB 31 ; trzecia wartosc
   DB 58 ; czwarta wartosc i koniec danych
Start:
   MOV   BL,02 ; 02 to poczatek danych
Rep:
   MOV   AL,[BL]
   CMP   AL,58 ; czy ostatni element ?
   JZ Start ; jesli tak, to skocz do start, zaczynamy od poczatku
   INC   BL ; jesli nie, to wez kolejny element
   JMP   Rep   ; skok na poczatek
   END
```

1.2. Procedury

Procedury pozwalają na szybsze i prostsze pisanie programów, a przede wszystkim zmniejszają zużycie pamięci przeznaczonej na kod programu. Opis instrukcji wykorzystywanych przy wywoływaniu procedur, zawarty został w tabeli 2. Dodatkowo, przy tworzeniu procedur używa się zwykle instrukcji ORG, aby adres procedury był znany.

Tabela 2. Instrukcje związane z implementacją procedur

Instr.	Operand	Działanie
CALL	adres procedury	zapisuje wartość rejestru IP na stosie i skacze do adresu wskazanego przez operand
RET		przywraca rejestr IP ze stosu i skacze pod adres tam zawarty

Listing 2 zawiera przykładową procedurę. Przykład prezentuje także sposób zapisu i przywracania zawartości rejestrów.

Listing 2. Prosta procedura licząca x^3

```
   clo
   call 30
   JMP   end_label
   ORG   30 ; poczatek kodu
   PUSH  AL ; zapisanie al na stosie
   PUSHF    ; zapisanie flag na stosie
   mov bl,1
   mul bl,al
   mul bl,al
   mul bl,al
   POPF     ; przywroc flagi.
   POP   AL ;przywroc AL.
   RET
end_label:
   END
```

Dane (parametry) do procedur możemy przekazywać na różne sposoby. Mamy trzy możliwości:

1. dane w rejestrach,
2. dane w pamięci ram,
3. dane na stosie.

Listing 3 ilustruje wykorzystanie każdej z powyższych metod. Należy pamiętać, że instrukcje procesora związane z procedurami odpowiadają tylko za zmianę zawartości rejestru IP. Programista powinien zapewnić, że procedura nie nadpisze rejestrów ustawionych przed jej wywołaniem (chyba że jest to pożądane). W tym celu należy używane rejestry bazowe i rejestr flag odłożyć na stos, a po wykonaniu procedury przywrócić ich wartości.

Listing 3. Sposoby przekazywania parametrów do procedur

```
; Procedury z parametrami
; W 3 przykladach chodzi o pokazanie roznych metod zapamietywania
; i odwolywania sie do danych

Jmp start ; pomijamy dane

   DB 0  ; Zarezerwuj bajt pamieci Ram pod adresem [02]
   DB 0  ; Zarezerwuj bajt pamieci Ram pod adresem [03]

; rejestry do zapamietania parametrow, parametow tyle ile rejestrow
start:
   MOV   AL,5
   MOV   BL,4
```

```
   CALL  30 ; Wywolanie procedury (al=al*bl)
         ; tak jak bylo do tej pory
; RAM do zapamietania parametrow
   MOV   AL,3
   MOV   [02],AL  ; al zapisujemy  w [02]
   MOV   BL,1
   MOV   [03],BL  ; bl zapisujemy [03]
   CALL  40 ;wywolanie procedury

; stos do zapamietania parametrow
; PUSH zapisuje na stosie, POP zdejmuje ze stosu
   MOV   AL,7
   PUSH  AL
   MOV   BL,2
   PUSH  BL
   CALL  60
   POP   BL
   POP   AL

;  dane w al i bl

   ORG   30 ; poczatek procedury1
   MUL   AL,BL

   RET      ; powrot


   ORG   40 ; Poczatek procedury2

   PUSH  CL ; zapmaietanie rejestrow
   PUSH  DL
   PUSHF    ; zapamietanie flag

   MOV   CL,[02]  ; pobranie parametrow z RAM
   MOV   DL,[03]
   MUL   CL,DL
   MOV   [02],CL  ; zapisanie wyniku

   POPF     ; przywrocenie oryginalnych zawartosci

   POP   DL
   POP   CL

   RET

   ORG   60 ; procedura 3
```

```
POP   DL
POP   BL
POP   AL

mul   AL,BL

PUSH  AL
PUSH  BL
PUSH  DL
RET

END
```

1.3. Dodatkowe informacje

— http://www.softwareforeducation.com/sms32v50/sms32v50_manual/index.htm
— https://pl.wikipedia.org/wiki/Stos_(informatyka)
— https://pl.wikipedia.org/wiki/Tablica_(informatyka)

2. Zadania

Każde zadanie polega na napisaniu programu zawierającego opisaną procedurę i podany sposób testowania.

2.1. Zadanie 1

Napisać procedurę, która dla dowolnej cyfry dziesiętnej (0..9) przekazanej w rejestrze AL zwróci w tym samym rejestrze AL kod ASCII dla podanej cyfry. Procedurę należy przetestować dla liczb 3, 7. Kody wyświetlić na wyświetlaczu VDU.

2.2. Zadanie 2

Napisać procedurę, która wyświetli tablicę cyfr znajdującą się w pamięci RAM. Adres pierwszego elementu tablicy powinien być przekazywany w rejestrze DL. Należy wykorzystać funkcję zamiany z poprzedniego zadania. Testowa tablica powinna zawierać przynajmniej 3 elementy. Procedura musi uwzględniać tablice różnej długości (jako ostatni element tablicy należy wstawić znaną wartość, np „-1”).

2.3. Zadanie 3

Napisać procedurę, która wyświetli ciąg znaków, zapisanych jako tablica w pamięci RAM, na wyświetlaczu VDU. Adres pierwszego elementu tablicy powinien być przekazywany w rejestrze AL. Należy przetestować procedurę na ciągu „Hello”. Procedura musi uwzględniać ciągi znaków różnej długości (jako ostatni element tablicy należy wstawić znaną wartość, zwykle jest to „0”).

2.4. Zadanie 4

Napisać procedurę, która spełnia założenia zadania 3, ale wyświetla tekst od końca. Zadanie można w prosty sposób rozwiązać wykorzystując stos.