

POLITECHNIKA CZĘSTOCHOWSKA
KATEDRA INTELIGENTNYCH SYSTEMÓW INFORMATYCZNYCH



PROGRAMOWANIE NISKOPOZIOMOWE

LABORATORIUM 6

OPERACJE NA ŁAŃCUCHACH

dr inż. Bartosz Kowalczyk

Częstochowa, 9 kwietnia 2024

Spis treści

Wstęp	3
1 Kopiowanie łańcuchów — MOVSB	5
2 Wypełnianie łańcuchów — STOSB	6
3 Porównywanie łańcuchów — CMPSB	7
4 Przeszukiwanie łańcuchów — SCASB	8
5 Odczytywanie łańcuchów — LODSB	9

Wstęp

Instrukcje łańcuchowe służą do zautomatyzowanego przetwarzania tzw. łańcuchów. Łańcuchem w kontekście programowania w języku Asembler jest ciągły obszar pamięci, na którego początek wskazuje określony rejestr indeksowy (ES: (R|E)DI i/lub DS: (R|E)SI).

Tabela 1: Mnemoniki i działanie instrukcji łańcuchowych

Mnemonik	Op. dst	Op. src	Działanie
MOVS(B W D Q)	ES: (R E)DI	DS: (R E)SI	MOV ES: [(R E)DI], DS: [(R E)SI]
STOS(B W D Q)	ES: (R E)DI	(R) (E)AX/AL	MOV ES: [(R E)DI], ((R) (E)AX/AL)
LODS(B W D Q)	(R) (E)AX/AL	DS: (R E)SI	MOV (R) (E)AX/AL, DS: [(R E)SI]
CMPS(B W D Q)	ES: (R E)DI	DS: (R E)SI	CMP ES: [(R E)DI], DS: [(R E)SI]
SCAS(B W D Q)	ES: (R E)DI	(R) (E)AX/AL	CMP ES: [(R E)DI], (R) (E)AX/AL

Pełną automatykę instrukcji łańcuchowych osiąga się poprzez zastosowanie odpowiedniego przedrostka. Każdorazowe wykonanie instrukcji opatrzonej przedrostkiem REP skutkuje dekrementacją rejestru (R) (E)CX.

Tabela 2: Przedrostki instrukcji łańcuchowych

Przedrostek	Działanie
REP	while((R) (E)CX != 0)
REPE	while((R) (E)CX != 0 && dst == src)
REPNE	while((R) (E)CX != 0 && dst != src)

Kierunek przetwarzania łańcucha określa wartość znacznika kierunku — DF (ang. *Direction Flag*). Dla DF = 0 (znacznik wyczyszczony) przetwarzanie odbywa się w kierunku rosnącym (od 0 do n) poprzez inkrementację wartości odpowiednich rejestrów indeksowych. W przeciwnym przypadku, tj. dla DF = 1 (znacznik ustawiony) przetwarzanie odbywa się w kierunku malejącym (od n do 0) poprzez dekrementację wartości odpowiednich rejestrów indeksowych.

Tabela 3: Kierunek przetwarzania łańcuchów

DF	Kierunek
0	Rosnący
1	Malejący

Inkrementacja oraz dekrementacja odbywają się automatycznie, każdorazowo kiedy zostanie wykonana instrukcja łańcuchowa. Poszczególne instrukcje łańcuchowe modyfikują wartości rejestrów indeksowych. Sposób modyfikacji (kierunek oraz wartość) są uwarunkowane poprzez wartość znacznika kierunku (DF) oraz postfix instrukcji (B — Byte, W — Word, D — Double word, Q — Quad word). Wpływ instrukcji łańcuchowych w poszczególnych wariantach na rejestry indeksowe jest pokazany w tabeli 4.

Tabela 4: Wpływ instrukcji łańcuchowych na rejestry indeksowe

Mnemonic	DF	Rejestr indeksowy	Zmiana wartości
MOVSB	0	ES: (R E)DI; DS: (R E)SI	+1
MOVSB	1	ES: (R E)DI; DS: (R E)SI	-1
MOVSW	0	ES: (R E)DI; DS: (R E)SI	+2
MOVSW	1	ES: (R E)DI; DS: (R E)SI	-2
MOVSD	0	ES: (R E)DI; DS: (R E)SI	+4
MOVSD	1	ES: (R E)DI; DS: (R E)SI	-4
MOVSQ	0	ES: (R E)DI; DS: (R E)SI	+8
MOVSQ	1	ES: (R E)DI; DS: (R E)SI	-8
STOSB	0	ES: (R E)DI	+1
STOSB	1	ES: (R E)DI	-1
STOSW	0	ES: (R E)DI	+2
STOSW	1	ES: (R E)DI	-2
STOSD	0	ES: (R E)DI	+4
STOSD	1	ES: (R E)DI	-4
STOSQ	0	ES: (R E)DI	+8
STOSQ	1	ES: (R E)DI	-8
LODSB	0	DS: (R E)SI	+1
LODSB	1	DS: (R E)SI	-1
LODSW	0	DS: (R E)SI	+2
LODSW	1	DS: (R E)SI	-2
LODSD	0	DS: (R E)SI	+4
LODSD	1	DS: (R E)SI	-4
LODSQ	0	DS: (R E)SI	+8
LODSQ	1	DS: (R E)SI	-8
CMPSB	0	ES: (R E)DI; DS: (R E)SI	+1
CMPSB	1	ES: (R E)DI; DS: (R E)SI	-1
CMPSW	0	ES: (R E)DI; DS: (R E)SI	+2
CMPSW	1	ES: (R E)DI; DS: (R E)SI	-2
CMPSD	0	ES: (R E)DI; DS: (R E)SI	+4
CMPSD	1	ES: (R E)DI; DS: (R E)SI	-4
CMPSQ	0	ES: (R E)DI; DS: (R E)SI	+8
CMPSQ	1	ES: (R E)DI; DS: (R E)SI	-8
SCASB	0	ES: (R E)DI	+1
SCASB	1	ES: (R E)DI	-1
SCASW	0	ES: (R E)DI	+2
SCASW	1	ES: (R E)DI	-2
SCASD	0	ES: (R E)DI	+4
SCASD	1	ES: (R E)DI	-4
SCASQ	0	ES: (R E)DI	+8
SCASQ	1	ES: (R E)DI	-8

1 Kopiowanie łańcuchów — MOVS

Korzystając z instrukcji łańcuchowych i przedrostka REP skopiuj wszystkie elementy wektora **a** do wektora **b**, gdzie:

1. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `char`.
2. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `short`.
3. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `int`.
4. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `int64`.

2 Wypełnianie łańcuchów — STOS

Korzystając z instrukcji łańcuchowych i przedrostka `REP` wypełnij wszystkie elementy wektora `a` wartością `0x41`, gdzie:

1. Wektor $\mathbf{a} \in \mathbb{Z}^n$ zawiera liczby typu `char`.
2. Wektor $\mathbf{a} \in \mathbb{Z}^n$ zawiera liczby typu `short`.
3. Wektor $\mathbf{a} \in \mathbb{Z}^n$ zawiera liczby typu `int`.
4. Wektor $\mathbf{a} \in \mathbb{Z}^n$ zawiera liczby typu `int64`.

3 Porównywanie łańcuchów — CMPS

Korzystając z instrukcji łańcuchowych i odpowiedniego przedrostka (REP, REPE, REPNE) napisz procedury:

1. Sprawdź, czy wszystkie elementy wektora **a** są równe elementom wektora **b**. Jeżeli tak, zwróć wartość 1. W przeciwnym przypadku zwróć wartość 0.
2. Sprawdź, czy wszystkie elementy wektora **a** są równe elementom wektora **b**. Jeżeli tak, zwróć wartość 1. W przeciwnym przypadku zwróć indeks pierwszego różnego elementu.

Powyższe zadania rozwiąż dla następujących scenariuszy:

1. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `char`.
2. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `short`.
3. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `int`.
4. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `int64`.

4 Przeszukiwanie łańcuchów — SCAS

Korzystając z instrukcji łańcuchowych i odpowiedniego przedrostka (REP, REPE, REPNE) napisz procedury:

1. Sprawdź, czy wektor **a** zawiera wartość 0. Jeżeli tak, zwróć wartość 1. W przeciwnym przypadku zwróć wartość 0.
2. Sprawdź, czy wektor **a** zawiera wartość 0. Jeżeli tak, zwróć indeks pierwszego napotkanego elementu. W przeciwnym przypadku zwróć wartość 0.
3. Policz ile wartości 0 znajduje się w wektorze **a**.
4. Policz ile wartości różnych od 0 znajduje się w wektorze **a**.

Powyższe zadania rozwiąż dla następujących scenariuszy:

1. Wektor $\mathbf{a} \in \mathbb{Z}^n$ zawiera liczby typu `char`.
2. Wektor $\mathbf{a} \in \mathbb{Z}^n$ zawiera liczby typu `short`.
3. Wektor $\mathbf{a} \in \mathbb{Z}^n$ zawiera liczby typu `int`.
4. Wektor $\mathbf{a} \in \mathbb{Z}^n$ zawiera liczby typu `int64`.

Procedury do edycji łańcuchów znaków

Korzystając z instrukcji łańcuchowych i odpowiednich przedrostków zrealizuj poniższe zadania:

1. Napisz procedurę `asm_rtrim`, której celem jest usunięcie wszystkich nadmiarowych spacji znajdujących się z prawej strony łańcucha znaków.
2. Napisz procedurę `asm_ltrim`, której celem jest usunięcie wszystkich nadmiarowych spacji znajdujących się z lewej strony łańcucha znaków.
3. Napisz procedurę `asm_trim`, której celem jest usunięcie wszystkich nadmiarowych spacji znajdujących się zarówno z prawej, jak i z lewej strony łańcucha znaków.
4. Napisz procedurę `asm_replace_all`, której celem jest zamiana wszystkich wystąpień znaku spacji, na znak o kodzie ASCII `0x03`. Oba znaki należy przekazać jako argumenty procedury.

Powyższe zadania należy przetestować podanymi ciągami znaków:

Listing 1: Testowe ciągi znaków

```
1 const uint len = 40u;  
2 const uint testCasesCnt = 3;  
3 char strings[testCasesCnt][len] = {  
4     "                Asembler jest super",  
5     "Asembler jest super                ",  
6     "                Asembler jest super    "  
7 };
```


5 Odczytywanie łańcuchów — LODS

Korzystając z instrukcji łańcuchowych napisz procedury:

1. Policz ile wartości przystych znajduje się w wektorze **a**.
2. Policz ile wartości nieprzystych znajduje się w wektorze **a**.
3. Policz ile wartości mniejszych od 0 znajduje się w wektorze **a**.
4. Policz ile wartości większych od 0 znajduje się w wektorze **a**.
5. Zmodyfikuj *in situ* elementy wektora $\mathbf{a} \in \mathbb{Z}^n$ zgodnie z funkcją $a_i = \frac{4a_i + 5}{2}$.
6. Oblicz wartość funkcji $b_i = \frac{4a_i + 5}{2}$ dla wszystkich elementów wektora **a**. Wyniki obliczeń umieść w wektorze **b**.
7. Skopiuj elementy wektora $\mathbf{a} \in \mathbb{Z}^n$ i zapisz je w wektorze $\mathbf{b} \in \mathbb{Z}^n$ w odwrotnej kolejności.
8. Odwróć kolejność elementów wektora $\mathbf{a} \in \mathbb{Z}^n$.

Powyższe zadania rozwiąż dla następujących scenariuszy:

1. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `char`.
2. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `short`.
3. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `int`.
4. Wektory **a** i **b** $\in \mathbb{Z}^n$ zawierają liczby typu `int64`.

Uwaga. W tym rozdziale nie należy stosować przedrostka `REP`. Zamiast tego należy zadeklarować jawną pętlę. Operacje pobrania i zapisania do łańcucha należy zrealizować przy użyciu instrukcji łańcuchowych.