

Programování na strojové úrovni

Cvičení 8: Programové konstrukce a RW/32

Knihovna rw32



Funkce	Vstup	Výstup
ReadChar		AL = hodnota znaku
ReadString	EBX = maximalni počet znaku, které mohou být načteny, EDI adresa kam se řetězec načte, rezervace EBX+1	EAX = počet přečtených znaků
ReadInt8		AL = číslo <-128,127>
ReadInt16		AX = číslo <-32768, 32767>
ReadInt32		EAX
ReadUInt8		AL = číslo <0,255>
ReadUInt16		AX = číslo <0,65535>
ReadUInt32		EAX = <0, 2 ³² - 1>
ReadFloat		EAX = 32bitové reálné číslo
ReadDouble		ST0 = 64bitové reálné číslo
WriteChar	AL = znak	
WriteNewLine		Vypíše konec řádku.
WriteString	ESI = ukazatel na řetězec zakončený nulou	Vypíše řetězec na obrazovku.
WriteInt8	AL = číslo	Vypíše AL v dekadické podobě bez znaménka.
WriteInt16	AX = číslo	Vypíše AX v dekadické podobě bez znaménka.
WriteInt32	EAX = číslo	Vypíše číslo EAX v dekadické podobě bez znaménka.
WriteUInt8	AL = číslo	Vypíše AL v dekadické podobě se znaménkem.
WriteUInt16	AX = číslo	Vypíše AX v dekadické podobě se znaménkem.
WriteUInt32	EAX = číslo	Vypíše číslo EAX v dekadické podobě se znaménkem.
WriteFlags		Vypíše příznakový registr na obrazovku.
WriteFloat	EAX = reálné číslo s jednoduchou přesností	
WriteDouble	ST0 = reálné číslo v koprocesoru vypisuje se číslo s dvojitou přesností	

Programové konstrukce – IF-ELSE



IF-ELSE

Jedná ze základních programových konstrukcí sloužících k divergenci kódu na základě podmínky (v případě, že není splněná vykoná se else).

Jazyk C:

```
if(a>b)
{
    //if condition is true
}
else
{
    //if condition is false
}
```



```
Ver1:
    mov eax, [a] ; a dd 100
    mov ebx, [b] ; b dd 100
    cmp eax, ebx
    jng else
    ;if condition is true
    jmp End

Else:
    ;if condition is false

End:
Ver2:
    mov eax, [a]
    cmp eax, [b]
    jng else
    ;if condition is true
    jmp End2

Else:
    ;if condition is false

End2
```

Programové konstrukce – WHILE



WHILE

Jedná ze základních programových konstrukcí sloužících k cyklickému opakování dané části kódu.

Jazyk C:

```
while(a!=0)
{
    //part for repeat
}
```



```
Ver1:
    mov eax, [a] ; a
    dd 100
    cmp eax, 0
    jz cylce_end
cylce:
    ;cycle body
    cmp eax, 0
    jnz cycle
cycle_end:
```

Programové konstrukce – FOR



Cyklus se známým počtem opakování

V případě, že známe dopředu počet, kolikrát se má cyklus vykonat máme opět několik možností, jak cyklus implementovat.

Jazyk C:

```
for(int i=10; i>0; i--) {...}
```



```
;ecx=i  
mov ecx, 10      ;misto 10 muzete  
pouzit jiny počet iteraci  
cycle:  
    cmp ecx,0  
    jz end  
  
    ;statement  
  
    dec ecx  
    jmp cycle  
  
end:
```

Programové konstrukce – přepínač



Řídicí konstrukce SWITCH

Tato řídicí konstrukce slouží pro větvení programu. Výběr varianty můžeme řešit buďto pomocí série if...else if ... else if nebo pomocí tabulky skoků. V následujícím příkladu je switch vyřešen pomocí skoků.

Jazyk C:

```
switch(x) {  
    Case 1: break;  
    Case 2: break;  
    Default: break;  
}
```



```
case_val1:  
    CMP [x],dword 1  
    JNE case_val2  
    ;statement1  
    JMP break  
case_val2:  
    CMP [x],dword 2  
    JNE case_default  
    ;statement2  
    JMP break  
case_default:  
    ;sDefault  
break:
```

Vytvořte aplikaci pro cenzuru slova v daném řetězci

Cílem tohoto cvičení je vytvořit funkci, která provede vyhledání slova v řetězci. V případě, že se toto slovo v řetězci vyskytne, provede nahrazení všech jeho znaků znakem 'x'. Pro načtení slova, které se bude nahrazovat a pro načtení řetězce použijte knihovnu `rw32-2015`.

Ukázka funkce aplikace

Zadejte slovo pro cenzuru: FEKT

Zadejte řetězec: „Radeji jsem mel jit studovat na FEKT“

Vystup: „Radeji jsem mel jit studovat na xxxx“

Definice funkce v C

```
void replace(char *word, char *string, int string_length, int word_length)
{
    //part of code for reapt
}
```

Postup

Nejdříve si proveďte dekompozici. Jelikož se jedná o složitější úlohu, je zapotřebí si problém rozložit na menší podproblémy. Veškeré potřebné instrukce jsme si už ukazovali.

Podproblémy:

- načtení dvou řetězců
- volání funkce a předávání ukazatelů
- vyhledávání slova v řetězci a práce s index registry
- nahrazení znaky
- výstup

Nápověda

- pro řešení tohoto příkladu doporučuji použít instrukce pro práci s řetězci
- bez jednoho cyklu se zřejmě neobejdete
- uvědomte si, co se děje a dá dělat s ESI a EDI