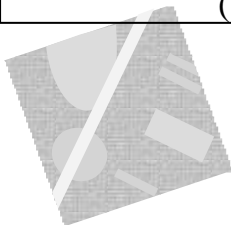


Jakso 3

Konekielinen ohjelmointi (TTK-91, KOKSI)



Muuttujat
Tietorakenteet
Kontrolli
Optimointi
Tarkistukset

7.8.2000 Teemu Kerola, K2000 1

Tieto ja sen osoite ⁽³⁾

```

X DC 12
LOAD R1, =X
LOAD R2, X
    
```

int x = 12;

muuttujan X arvo

symbolin X arvo

230
12345
12556
128765
12222
12
12998

- Muuttujan X osoite on 230
- Muuttujan X arvo on 12
- Symbolin X arvo on 230 X=230:
 - symbolit ovat yleensä olemassa vain käännoa aikana!
 - Virheilmoituksia varten symbolitaulua pidetään joskus yllä myös suoritus aikana

7.8.2000 Teemu Kerola, K2000 2

Tieto ja sen osoite ⁽⁵⁾

```

Xptr DC 0
X DC 12
LOAD R1, =X
STORE R1, Xptr
LOAD R2, X
LOAD R3, @Xptr
    
```

Xptr=225

X=230:

230
12345
12556
128765
12222
12
12998

- Muuttujan X osoite on 230
- Muuttujan X arvo on 12
- Osoitinmuuttujan (pointterin) Xptr osoite on 225
- Osoitinmuuttujan Xptr arvo on 230
- Osoitinmuuttujan Xptr osoittaman kokonaisluvun arvo on 12

7.8.2000 Teemu Kerola, K2000 3

Osoitinmuuttujat

- Muuttujia samalla tavoin kuin kokonaislukuarvoiset muuttujatkin
- Arvo on jonkun tiedon osoite muistissa
 - globaalin monisanaisen tiedon osoite
 - taulukot, tietueet, oliot
 - kasasta (heap) dynaamisesti (suoritus aikana) varatun tiedon osoite
 - Pascalin tai Javan "new" operaatio palauttaa varatun muistialueen osoitteen (tai virhekoodin, jos operaatiota ei voi toteuttaa)
 - aliohjelman tai metodin osoite
 - osoite ohjelmakoodiin!

7.8.2000 Teemu Kerola, K2000 4

Globaali data ⁽²⁾

- Globaalit muuttujat ja muut tietorakenteet sijaitsevat muistissa ohjelmakoodin jälkeä

– muuttujat

```

int X = 25;
short Y;
float Ft;
        
```

– tilan varaus

```

X DC 25 ; alkuarvo = 25
Y DC 0
lBig DC 1 ; 1=true, 0=false
        
```

– viittaaminen

```

LOAD R1, X
STORE R2, Y
        
```

```

char Ch;
char Str[] = "Pekka";
boolean lBig;
        
```

7.8.2000 Teemu Kerola, K2000 5

Aritmeettinen lauseke ⁽²⁾

```

int a, b, c;
...
b = 34;
a = b + 5 * c;
    
```

tilan varaus

A	DC	0
B	DC	0
C	DC	0

koodi

```

LOAD R1, =34
STORE R1, B
LOAD R1, B
LOAD R2, C
MUL R2, =5
ADD R1, R2
STORE R1, A
    
```

7.8.2000 Teemu Kerola, K2000 6

Globaalin taulukon tilan varaus ja käyttö (2)

```
int X, Y;
int [] Taulu = new int [30];
...
X = 5;
Y = Taulu[X];
```

X	DC	0
Y	DC	0
Taulu	DS	30
...		
	LOAD R1, =5	
	STORE R1, X	
	LOAD R1, X	
	LOAD R2, Taulu(R1)	
	STORE R2, Y	

X:		
Y:		
Taulu:		

7.8.2000
Teemu Kerola, K2000
7

Globaalien tietueiden tilan varaus ja käyttö (3)

```
int X;
public class Opisk
{ public int Pituus, Paino; }
...
Opisk Tauno = new Opisk ();
...
X = Tauno.Paino
```

X	DC	0
Opisk	DS	2
Pituus	EQU	0
Paino	EQU	1
...		
	LOAD R1, =Opisk	
	LOAD R2, Paino(R1)	
	STORE R2, X	

Tietueen osoite	X:	
on sen ensimmäisen sanan osoite	Opisk:	pituus
		paino

7.8.2000
Teemu Kerola, K2000
8

Muistitilan käyttö yhdelle ohjelmalle (6)

7.8.2000
Teemu Kerola, K2000
9

Kontrolli - valinta konekielellä (4)

- Ehdoton hyppy
 - JUMP, CALL, EXIT, SVC
- Hyppy perustuen laiterekisterin arvoon
 - JZER, JPOS, ...
- Hyppy perustuen aikaisemmin asetetun tilarekisterin arvoon
 - COMP
 - JEQU, JGRE, ...

```
COMP R2, LIMIT
JEQU LOOP
```

7.8.2000
Teemu Kerola, K2000
10

If-then-else -valinta (1)

```
if (a < b)
  x = 5;
else
  x = y;
```

	LOAD R1, A
	COMP R1, B
	JNLE Else
	LOAD R1, =5
	STORE R1, X
	JUMP Done
Else	LOAD R1, Y
	STORE R1, X
Done	NOP

7.8.2000
Teemu Kerola, K2000
11

Toistolausekkeet (2)

- For-step-until -silmukka
- Do-until -silmukka
- Do-while -silmukka
- While-do -silmukka

ehto silmukan alussa

ehto silmukan lopussa

7.8.2000
Teemu Kerola, K2000
12

For lauseke (1)

```

for (int i=20; i < 50; ++i)
    T[i] = 0;
    
```

I	DC 0
	LOAD R1, =20 STORE R1, I
Loop	LOAD R2, =0 LOAD R1, I STORE R2, T(R1)
	LOAD R1, I ADD R1, =1 STORE R1, I
	LOAD R3, I COMP R3, =50 JLES Loop

7.8.2000 Teemu Kerola, K2000 13

While-do -lauseke (1)

```

X = 14325;
Xlog = 1;
Y = 10;
while (Y < X) {
    Xlog++;
    Y = 10*Y;
}
    
```

	LOAD R1, =14325 STORE R1, X
	LOAD R3, =1 ; R3=Xlog LOAD R2, =10 ; R2=Y
While	COMP R2, R1 JNLES Done
	ADD R3, =1 MUL R2, =10 JUMP While
Done	STORE R3, Xlog ; talleta tulos STORE R2, Y

7.8.2000 Teemu Kerola, K2000 14

Case lauseke (1)

```

Switch (lkm) {
case 4: x = 11;
       break;

case 0: break;

default: x = 0;
        break;
}
    
```

	LOAD R1, Lkm
Vrt4	COMP R1, =4 JNEQ Vrt0 LOAD R2, =11 STORE R2, X JUMP Cont
Vrt0	COMP R1, =0 JNEQ Def JUMP Cont
Def	LOAD R2, =0 STORE R2, X
Cont	NOP

7.8.2000 Teemu Kerola, K2000 15

Koodin generointi (9)

- Kääntäjän viimeinen vaihe
 - voi olla 50% käännösajasta
- Tavallinen koodin generointi
 - alustukset, lausekkeet, kontrollirakenteet
- Optimoidun koodin generointi
 - käännös kestää kauemmin
 - suoritus tapahtuu nopeammin
 - milloin globaalin muuttujan X arvo kannattaa pitää rekisterissä ja milloin ei?
 - Missä rekisterissä X:n arvo kannattaa pitää?

7.8.2000 Teemu Kerola, K2000 16

Optimoitu For lauseke

```

for (int i=20; i < 50; ++i)
    T[i] = 0;
    
```

I	DC 0
	LOAD R1, =20 STORE R1, I
Loop	LOAD R2, =0 LOAD R1, I STORE R2, T(R1)
	LOAD R1, I ADD R1, =1 STORE R1, I
	LOAD R3, I COMP R3, =50 JLES Loop

122 vs. 272 suoritettua käskyä!
Mitä eroja? Onko tämä OK?

7.8.2000 Teemu Kerola, K2000 17

Virhetilanteisiin varautuminen (3)

- Suoritin tarkistaa käskyt
 - integer overflow, ADD R1, R2 ; overflow??
 - divide by zero, ... DIV R4, =0 ; divide-by-zero
- Generoidut konekäskyt tarkistavat ja explisiittisesti aiheuttavat keskeytyksen tai käyttöjärjestelmän palvelupyynnön tarvittaessa
 - index out of bounds, bad method, bad operand, iban, mitti, pois, haluat, taitat

	LOAD R3, Tsize ; tarkista
	COMP R1, R3
	JLES IndexOK
	SVC SP, =BadIndex

Index OKADD R2, Taulu(R1); R1 = 12 345 000 ??

7.8.2000 Teemu Kerola, K2000 18

Taulukon indeksitarkistus

```

for (int i=20; i < 50; ++i)
    T[i] = 0;
    
```

I	DC	0
T	DS	50
Tsize	DC	50 ;koko
...		

```

LOAD R1, =20
STORE R1, I
Loop
LOAD R2, =0
LOAD R1, I
JNNEG R1, ok1
SVC SP(=BadIndex)
ok1
COMP R1, Tsize
JLES ok2
SVC SP(=BadIndex)
ok2
STORE R2, T(R1)
LOAD R1, I
ADD R1, =1
STORE R1, I ; 50 OK!
LOAD R3, I
COMP R3, =50
JLES Loop
    
```

7.8.2000 Teemu Kerola, K2000 19

Taulukon alaindeksi ei ala nolasta (2)

```

for (int i=20; i < 50; ++i)
    T[i] = 0;
    
```

DC	0	
T	DS	30 ; 30 alkioita
Tlow	DC	20 ; alaraja
Thigh	DC	50 ; yläraja+1
...		

```

LOAD R1, =20
STORE R1, I
Loop
LOAD R2, =0
LOAD R1, I
SUB R1, Tlow
STORE R2, T(R1)

LOAD R1, I
ADD R1, =1
STORE R1, I

LOAD R3, I
COMP R3, =50
JLES Loop
    
```

indeksitarkistukset...

7.8.2000 Teemu Kerola, K2000 20

2-ulotteiset taulukot (6)

```

int[][] T = new int[4][3];
...
Y = T[i][j];
    
```

T	DS	12
Trows	DC	4
Tcols	DC	3
...		

```

LOAD R1, I
R1
MULT R1, Tcols
ADD R1, J
LOAD R2, T(R1)
STORE R2, Y
    
```

Tarkistukset.... ?

7.8.2000 Teemu Kerola, K2000 21

Moni-ulotteiset taulukot (3)

- Talletus riveittäin
 - C, Pascal, Java?
- Talletus sarakeittain
 - Fortran
- 3- tai useampi ulotteiset
 - samalla tavalla!

7.8.2000 Teemu Kerola, K2000 22

Linkitetty lista (6)

```

list_sumk91
Data EQU 0 ; suht. osoite
Next EQU 1
Sum DC 0
Main
LOAD R1, First ; ptrRec
JNEG R1, Done
LOAD R2, =0 ; sum
Loop
ADD R2, Data(R1)
LOAD R1, Next(R1)
JNNEG R1, Loop
STORE R2, Sum
Done
SVC SP, =HALT
    
```

7.8.2000 Teemu Kerola, K2000 23

Monimutkaiset tietorakenteet

- 2-ulotteinen taulukko T, jonka jokainen alkio on tietue, jossa neljä kenttää:
 - pituus
 - ikä
 - viime vuoden palkka kunakin kuukautena
 - viime vuoden töissäolopäivien lukumäärä kunakin kuukautena
- Talletustapa?
- Viitteet? $X = T[\text{yloppNum}][\text{opNum}].\text{palkka}[\text{kk}];$

7.8.2000 Teemu Kerola, K2000 24

