

Luento 3 (verkkoluento 4) Aliohjelmien toteutus

Tyypit, Parametrit
Aktivointitietue (AT)
AT-pino, rekursio

6.11.2012 Copyright 2012 Teemu Kerola 1

Aliohjelmatyypit

- Korkean tason ohjelmointikielen käsitteet
 - aliohjelma, proseduuri
 - parametrit
 - funktio
 - parametrit, paluuarvo
 - metodi
 - parametrit, ehkä paluuarvo
- Konekielitason vastaava käsite
 - aliohjelma
 - parametrit ja paluuarvo(t)

6.11.2012 Copyright 2012 Teemu Kerola 2

Parametrit ja paluuarvo

- Muodolliset parametrit
 - määritelty aliohjelmissä **ohjelmointihetkellä**
 - tietty **järjestys ja tyyppi**
 - paluuarvot
 - käsittely hyvin samalla tavalla kuin parametreillekin
- Todelliset parametrit ja paluuarvo
 - todelliset parametrit sijoitetaan muodollisten parametrien paikalle **kutsuhetkellä suoritusaikana**
 - paluuarvo saadaan **paluuhetkellä** ja sitä käytetään kuten mitä tahansa arvoa

Tulosta (int x, y)
void Tulosta (int x, y)

Laske(int x): int
int Laske(int x)

Tulosta (5, apu);
x = Laske(y+234);

6.11.2012 Copyright 2012 Teemu Kerola 3

Parametrityypit

- Arvoparametri
 - välitetään parametrin arvo (eli sen kopio) kutsuhetkellä
 - arvo voidaan lukea
 - alkuperäistä arvoa ei voi muuttaa, mutta arvon kopiota voi muuttaa
- Viiteparametri
 - välitetään parametrin osoite
 - arvo ja osoite voidaan lukea, arvoa voi muuttaa
- Nimiparametri (*ei yleensä käytetty*)
 - välitetään parametrin nimi
 - nimi (merkkijono) kuvataan arvoksi **kutsuhetkellä**
 - semantiikka määräytyy vasta kutsuhetkellä

```

swap(i,j)
tmp = i;
i = j;
j = tmp;
    
```

swap(k, T[k]);

tmp = k;
 k = T[k];
 T[k] = tmp;

Keskustele 4

6.11.2012 Copyright 2012 Teemu Kerola

Aliohjelmien toteutuksen osat

- Paluuosoite
 - kutsukohtaa seuraavan käskyn osoite
- Parametrien välitys
- Paluuarvon välitys
- Paikalliset muuttujat
- Rekisterien varaus (allokointi)
 - Kutsuva ohjelman osa haluaa säilyttää käyttämiensä rekisterien arvot!
 - pääohjelma, toinen aliohjelma, sama aliohjelma, metodi, ...
 - Aliohjelman pitää aluksi tallettaa muistiin käytettävien rekisterien arvot ja lopuksi palauttaa ne ennalleen

6.11.2012 Copyright 2012 Teemu Kerola 5

Aktivointitietue

(activation record, activation frame)

int funcA (int x,y);

- Aliohjelman toteutus (ttk-91)
 - funktion paluuarvo (tai kaikki paluuarvot)
 - kaikkien (sisäänmeno- ja ulostulo-) parametrien arvot
 - paluuosoite
 - kutsukohdan aktivointitietue
 - kaikki paikalliset muuttujat ja tietorakenteet
 - aliohjelman ajaksi talletettujen rekistereiden alkuperäiset arvot

paluuarvo	} -2 +1
param x	
param y	
vanha PC	
paik. m. i1	
vanha R1	
paik. m. i2	
vanha R2	

kasvava muistiosoite 6

6.11.2012 Copyright 2012 Teemu Kerola

Aktivointitietuepino muistissa

- Aktivointitietueet (AT) varataan ja vapautetaan dynaamisesti (suoritusaikana) pinosta (muistista)
 - SP (=R6) osoittaa pinon pinnalle
- Aktivointitietuepino
 - FP (R7) osoittaa voimassa olevan AT:n sovittuun kohtaan (ttk-91: vanhan FP:n osoite)
- Pinossa olevaa AT:tä rakennetaan ja puretaan käskyillä:
 - PUSH, POP, PUSHR, POPR
 - CALL, EXIT (SV, IRET)

Talleta R0-R5 pinoon

6.11.2012 Copyright 2012 Teemu Kerola 7

Aliohjelmakutsun toteutus

- Toteutus jaettu eri yksiköille

- Kutsuva rutiini**
 - varaava tilaa paluuarvolle pinosta
 - laita parametrit (arvot tai osoitteet) pinoon
- CALL käsky**
 - talleta vanha PC ja FP, aseta uudet PC ja FP
- Kutsuttu rutiini**
 - varaava tilaa paikallisille muuttujille
 - talleta käytettävien rekistereiden vanhat arvot pinoon
 - (itse aliohjelman toteutus – varsinainen työ)
- EXIT käsky**
 - palauta rekistereiden arvot
 - vapautaa paikallisten muuttujien tila
- Kutsuva rutiini**
 - palauttaa PC ja FP
 - vapautaa parametrien tila
 - ota paluuarvo pinosta

6.11.2012 Copyright 2012 Teemu Kerola 8

Aliohjelmaesimerkki

```

int fA (int x, y)
{
    int z = 5;
    z = x * z + y;
    return (z);
}
    
```

käyttö:

```

R    DC 24
...
PUSH SP,=0 ; tila paluuarvolle
...
PUSH SP, R    ← muistista muistiin!!
CALL SP, fA   ← talleta PC, FP aseta PC, kutsu & paluu palautaa FP, PC
POP  SP, R1   ← 2. operandi aina rekisteri
STORE R1, T
    
```

tämän hetkinen, nykyinen FP

6.11.2012 Copyright 2012 Teemu Kerola Keskustele 9

Aliohjelmaesimerkki

```

int fA (int x, y)
{
    int z = 5;
    z = x * z + y;
    return (z);
}
    
```

käyttö:

```

R    DC 24
...
PUSH SP,=0 ; ret. value space
PUSH SP,=200 ← muistista muistiin!!
PUSH SP, R    ← talleta PC, FP aseta PC, kutsu & paluu palautaa FP, PC
CALL SP, fA
POP  SP, R1   ← 2. operandi aina rekisteri
STORE R1, T
    
```

tämän hetkinen, nykyinen FP

6.11.2012 Copyright 2012 Teemu Kerola 10

Aliohjelmaesimerkki

```

aliohjelman toteutus:
retfA EQU -4 # params
parX  EQU -3
parY  EQU -2
locZ  EQU 1 # local vars

fA    PUSH SP,=0 ; alloc Z
      PUSH SP, R1 ; save R1

      LOAD R1,=-5; init Z
      STORE R1, locZ (FP)

      LOAD R1, parX (FP)
      MUL  R1, locZ (FP)
      ADD  R1, parY (FP)
      STORE R1, locZ (FP)
      STORE R1, retfA (FP)

      POP  SP, R1; recover R1
      SUB  SP,=1 ; free Z
      EXIT SP,=2 ; 2 param.
    
```

Kaikki viitteet näihin tehdään suhteessa FP:hen

6.11.2012 Copyright 2012 Teemu Kerola Keskustele 11

Aliohjelmaesimerkki

```

aliohjelman toteutus:
retfA EQU -4
parX  EQU -3
parY  EQU -2
locZ  EQU 1

fA    PUSH SP,=0 ; alloc Z
      PUSH SP, R1 ; save R1

      LOAD R1,=-5; init Z
      STORE R1, locZ (FP)

      LOAD R1, parX (FP)
      MUL  R1, locZ (FP)
      ADD  R1, parY (FP)
      STORE R1, locZ (FP)
      STORE R1, retfA (FP)

      POP  SP, R1; recover R1
      SUB  SP,=1 ; free Z
      EXIT SP,=2 ; 2 param.
    
```

Kaikki viitteet näihin tehdään suhteessa FP:hen

6.11.2012 Copyright 2012 Teemu Kerola 12

KJ-palvelun kutsu (proseduraalisesti)

- Samalla tavalla kuin aliohjelman kutsu
 - CALL käskyn asemesta SVC
- Tila paluuarvolle?
- Parametrit pinoon vai rekistereissä?
- SVC kutsu
 - Kutsuttavan rutiinin numero operandina
- IRET paluu
- Paluuarvo (OK, virhe) pois pinosta tarkistusta varten

```
fOK = ReadBlock (fp, 64)
...
PUSH SP, =0 ;paluuarvo
PUSH SP, =FileBuffer
PUSH SP, CharCnt
PUSH SP, FilePtr

SVC SP, =ReadFile

POP SP, R1
JNZER R1, FileTrouble
...
```

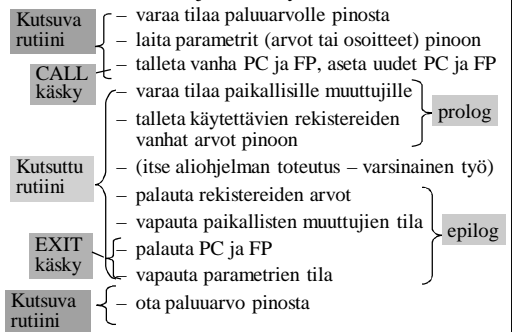
6.11.2012

Copyright 2012 Teemu Kerola

Keskustele 13

Aliohjelmakutsun toteutus

- Toteutus jaettu eri yksiköille



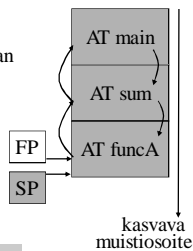
6.11.2012

Copyright 2012 Teemu Kerola

Keskustele 14

Aktivointitietuepino muistissa

- Aktivointitietueet (AT) varataan ja vapautetaan dynaamisesti (suoritusaikana) pinosta (muistista)
 - SP (=R6) osoittaa pinon pinnalle
- Aktivointitietuepino
 - FP (R7) osoittaa voimassa olevan AT:n sovittuun kohtaan (ttk-91: vanhan FP:n osoite)
- Pinossa olevaa AT:tä rakennetaan ja puretaan käskyillä:
 - PUSH, POP, PUSH, POPR
 - CALL, EXIT (SVC, IRET)



6.11.2012

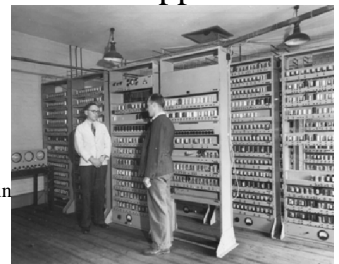
Copyright 2012 Teemu Kerola

Keskustele 15

-- Luennon 3 loppu --

M. Wilkes:
EDSAC I (1949)

- rekisterit (6 kpl), tyhjiöputkilla
- käsky- ja datamuisti, 32 elohopea-viiveputkea, kukin 32 kpl 18b sanaa (n. 2 Kb)
- kertolasku 5.4ms, http://www.cl.cam.ac.uk/Relics/archive_photos.html
- 650 IPS
- ensimmäinen "stored program" -tietokone
- 3000 tyhjiöputkea, sähkökulutus 12 kW, tila 5x4m



6.11.2012

Copyright 2012 Teemu Kerola

16