

Datorsystemteknik

för D2, ICT2, E3 och Mek3

Nicholas Wickström

Högskolan i Halmstad
Sverige

Datorsystemteknik – p.1/18

Översikt

- Sammanfattning av föregående fl.
- Satser *if*, *switch*, *while*, *do*, *for*, *goto*
- Operatorer
- Vektorer och matriser
- Funktioner
- Preprocessor

Datorsystem

Satser - (if, if-(then)-else)

```
/* if sats */
if (n > 0) {
    fAvg = fSum / (float)n;
}

/* if-then-else */
if (n >= 0)
    nPositive = TRUE;
else
    nPositive = FALSE;
```

Datorsystemteknik – p.3/18

Satser - (if, if-(then)-else, forts.)

```
/* Nästlad if-then-else sats, flervärdig */
if (nColor == YELLOW)
    nYellow++;
else if (nColor == BLUE)
    nBlue++;
else if (nColor == GREEN)
    nGreen++;
else
    nOthers++;
```

Datorsystem

Satser - (switch)

```
switch (nColor) { /* switch-sats, flervals
  case YELLOW:
    nYellow++;
    break;
  case BLUE:
    nBlue++;
    break;
  ...
  default:
    nOthers++;
    break;
}
```

Datorsystemteknik – p.5/18

Satser - (while)

```
/* Beräkna n-fakulteten (n!) */
nfak = i = 1;
while (++i <= n)
  nfak *= i;
```

Datorsystem

Satser - (do)

```
do {
  if (x-y < 0)
    nError += nSmallAdjust;
  else
    nError -= nSmallAdjust;
} while (nError > MAX_ERR ||
        nError < -MAX_ERR)
```

Datorsystemteknik – p.7/18

Satser - (for)

```
for (expr1; expr2; expr3)
  sats
```

ekivalent med

```
expr1;
while (expr2) {
  sats
  expr3;
}
```

Datorsystem

Satser - (for forts.)

```
/* x raised to n */  
for (i = 1; i <= n; i++)  
    nResult *= x;
```

```
/* Beräkna n-fakulteten (n!) (for-sats) */  
for (i = 1; i <= n; i++)  
    nfak *= i;
```

Operatorer (ökning/minskning)

```
nAntal++; /* Ekv. nAntal = nAntal +  
++nAntal; /* Ekv. nAntal = nAntal +
```

```
i = 4; j = 7;  
a = ++i * --j; /* a => 30 */  
i = 4; j = 7;  
a = i++ * j--; /* a => 28 */
```

Operatorer (Jämförelse)

< > <= >= == !=

```
x < y    a > 3.5    i + j <= 4  
x == y   i != 0
```

Operatorer (Logiska)

&& || !

OCH, ELLER, ICKE

```
x > 0 || !(i == 2 && j < 4)
```

Obs!! i == !!i;

```
/* Behöver inte vara sant! */
```

Operatorer (Bit)

~ << >> & ^ |

```
BNEG, VSHFT, HSHFT, BOCH, BXOR, BOR
c = 0x8F; /* 10001111 */
c2 = ~c /* 01110000, eller 0x70 */
```

```
c3 = c >> 2 /* 00100011 */
```

```
k = k | 0x20; /* Sätt 6e biten */
k = k & 0xFFDF; /* Nolla 6e biten */
```

Datorsystemteknik – p.13/18

Operatorer (Tilldelning)

```
i = 0; /* Enkel vanlig tilldelning */
j = i; /* Alt. i = j = 0; */
```

```
i = i + j; /* Alt. i += j; */
```

Datorsystemteknik – p.14/18

Vektorer, matriser

```
int nVector[100];
int nMatrix[3][3] =
{ { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };
int nTranspose[3][3];
/* Medelvärdet av vektorn */
for (i = 0; i < 100; i++)
    nSum += nVector[i]/100;

/* Transponat av matrisen nMatrix */
for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        nTranspose[i][j]=nMatrix[j][i];
```

Datorsystemteknik – p.15/18

Funktioner

```
float calc_avg (float x, float y)
{
    return ((x + y) / 2.0);
}

/* Anropas med */
nAvg = calc_avg(a, b);
/* Begränsning! Kan bara returnera en värde */
/* Kan lösas med pointer */
/* Men egentligen behöver vi pekare!
```

Datorsystemteknik – p.16/18

Preprocessor

```
#define N 1000
/* Makro */

#define TRUE 1
#define FALSE !TRUE

#ifdef DEBUG
/* print debug info */
#endif
```

Preprocessor forts.

```
#ifndef _MYHEADER_H_
#define _MYHEADER_H_

/* include file here */

#endif

/* Preprocessor byter dessa innan k
```