Globala och lokala variabler,

inparametrar vid funktionsanrop,
och
returvärden från funktioner i XCC

samt

Styrstrukturer i C

2012-02-19

Variabler i XCC.

För variabler gäller: char 8 bitar, short och int 16 bitar, long 32 bitar.

Globala variabler i XCC.

Deklaration av globala (externa) variabler (görs utanför main).
De globala variablerna får fasta platser i minnet, som framgår nedan.

```
short   shortint;                       _shortint:      RMB   $2
long    longint;                        _longint:       RMB   $4
int     justint;                        _justint:       RMB   $2
int     intvec[3];                      _intvec:        RMB   $6

struct {
int     s1;
char    s2;
char    *s3;
} komplex;                              _komplex:       RMB   $5
```

Deklaration och initiering av globala variabler (görs utanför main).

```
short   shortint = 0x1122;              _shortint:      FDB   $1122
long    longint = 0x33445566;           _longint:       FQB   $33445566
int     justint = 0x7788;               _justint:       FDB   $7788
int     intvec[3] = {0x1234,0x5678,0x9ABC};  _intvec:   FDB   $1234
                                                        FDB   $5678
                                                        FDB   $9ABC
struct {
int     s1;
char    s2;
char    *s3;
} komplex = {0x3421,0xFF,0x3333};       _komplex:       FDB   $3421
                                                        FCB   $FF
                                                        FDB   $3333
```

Lokala variabler i XCC.

Deklaration av lokala variabler (görs inne i en funktion).

Alla lokala variabler läggs på stacken.

Lokala variabler som deklareras placeras på stacken i den ordning de deklareras, dvs sist behandlad finns överst i stacken. Övriga lokala variabler placeras också på stacken i den ordning behovet av dem uppstår, dvs den sista finns överst på stacken.

Varje funktion som har lokala variabler inleds med prologen `LEAS -?,SP`
och avslutas med epilogen `LEAS ?,SP` följt av `RTS.`

Prolog skapar alltså utrymme för de lokala variablerna på stacken genom att minska SP.

Epilog återställer stacken så att SP pekar på återhoppsadressen.

```
void main(void) {          _main:   LEAS    -19,SP    Prolog
  short  shortint;
  long   longint;
  int    justint;
  int    intvec[3];

  struct {
    int  s1;
    char  s2;
    char *s3;
  } komplex;

justint = 0;                        CLRA
                                    CLRB
                                    STD     11,SP     justint = 0 till stack

                                    LEAS    19,SP     Epilog
}                                   RTS
```

Hur lokala variabler tilldelas värden på stacken

```
void main(void) {                  _main: LEAS     -19,SP      Prolog

  short  shortint;
  long     longint;
  int        justint;
  int        intvec[3];

  struct {
    int  s1;
    char s2;
    char *s3;
  } komplex;

  shortint = 1;                          LDD      #1
                                         STD      17,SP
  longint = 2;                           LDY      #0x0000
                                         LDD      #0x0002
                                         STY      13,SP
                                         STD      15,SP
  justint = 3;                           LDD      #3
                                         STD      11,SP
  intvec[0] = 4;                         LDD      #4
                                         STD      5,SP
  intvec[1] = 5;                         LDD      #5
                                         STD      7,SP
  intvec[2] = 6;                         LDD      #6
                                         STD      9,SP
  komplex.s1 = 7;                        LDD      #7
                                         STD      0,SP
  komplex.s2 = 8;                        LDAB     #8
                                         STAB     2,SP
  komplex.s3 = (char *) 9;               LDD      #9
                                         STD      3,SP

                                         LEAS     19,SP       Epilog
}                                        RTS
```

Inparametrar vid funktionsanrop i XCC.

Inparameterlistan till en funktion behandlas från höger till vänster. Samtliga inparametrar placeras på stacken i den ordning de behandlas före funktionsanropet (subrutinanropet).

Funktionsanrop med två globala inparametrar
Inga lokala parametrar i den anropade funktionen

```
void callfunc(int aa, int bb);                          Deklaration av callfunc

int alfa = 0x1234;          _alfa:      FDB   $1234     Initiering av globala variabler
int beta = 0x5678;          _beta:      FDB   $5678

void main(){                _main:                      Inparametrar till stack
  callfunc(alfa,beta);                  LDD   _beta
                                        PSHD            beta till stack
                                        LDD   _alfa
                                        PSHD            alfa till stack
                                        JSR   _callfunc Funktionsanrop
                                        LEAS  4,SP      Justera stack efter anrop
}                                       RTS             Åter till start_up


void callfunc(aa,bb){       _callfunc:                  callfunc är en tom funktion
}                                       RTS
```

Funktionsanrop med två globala inparametrar
Två lokala parametrar i den anropade funktionen

```
void callfunc(int aa, int bb);                          Deklaration av callfunc

int alfa = 0x1111;          _alfa:      FDB   $1111     Initiering av globala variabler
int beta = 0x5678;          _beta:      FDB   $5678

void main(void){            _main:                      Inparametrar till stack
  callfunc(alfa,beta);                  LDD   _beta
                                        PSHD            beta till stack
                                        LDD   _alfa
                                        PSHD            alfa till stack
                                        JSR   _callfunc Funktionsanrop
                                        LEAS  4,SP      Justera stack efter anrop
}                                       RTS             Åter till start_up

void callfunc(aa,bb){       _callfunc:  LEAS  -3,SP     Prolog till callfunc

  char gamma = 0x55;                    LDAB  #$55      Initiering av lokala variabler
                                        STAB  2,SP      gamma till stack
  int  delta = 0x9999;                  LDD   #$9999
                                        STD   0,SP      delta till stack

                                        LEAS  3,SP      Epilog till callfunc
}                                       RTS             Åter till main
```

Funktionsanrop med två lokala inparametrar från main
Två lokala parametrar i den anropade funktionen

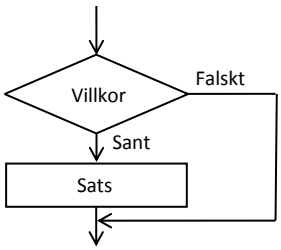
```
void callfunc(int aa, int bb);                                  Deklaration av callfunc

void main(void){        _main:  LEAS    -4,SP       Prolog till main

  int alfa = 0x1111;            LDD     #$1111      Initiering av lokala variabler
                                STD     2,SP        alfa till stack
  int beta = 0x5678;            LDD     #$5678
                                STD     0,SP        beta till stack

                        *                           Inparametrar till stack
callfunc(alfa,beta);            PSHD                beta finns i D-reg
                                LDD     4,SP        alfa
                                PSHD
                                JSR     _callfunc   Funktionsanrop
                                LEAS    4,SP        Justera stack efter anrop

                                LEAS    4,SP        Epilog
}                               RTS


void callfunc(aa,bb){   _callfunc:  LEAS    -3,SP   Prolog

  char gamma = 0x55;            LDAB    #$55        Tilldelningar
                                STAB    2,SP        gamma till stack
  int  delta = 0x9999;          LDD     #$9999
                                STD     0,SP        delta till stack

                                LEAS    3,SP        Epilog
}                               RTS
```

Returvärden från funktioner i XCC.

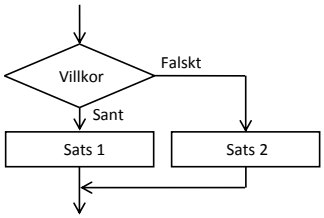
char, int, short, long och float returneras i register.
char i B-registret.
int och short i D-registret.
long och float i Y- och D-registret, med mest signifikant del i Y-registret.

Styrstrukturer i C

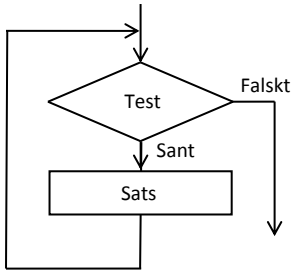
Ex.
```
if ( i > 0 )
  sats
```



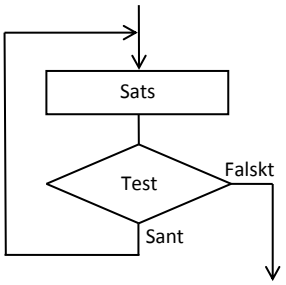
Ex.
```
if ( i > 0 )
  sats1
else
  sats2
```



Ex.
```
while ( i > 0 )
  sats
```

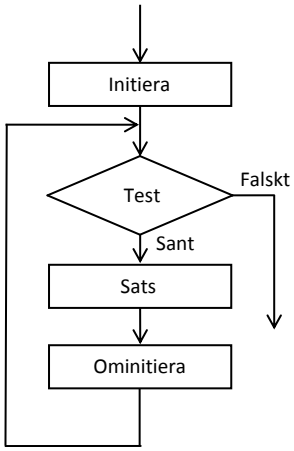


Ex.
```
do
   sats
  while ( i > 0 );
```



Ex.

```
for ( i = 0; i < n; i++)
  sats
```



Ex.

```
switch ( i ) {
   case   Fall1:
      satser
      break;
   case   Fall2:
      satser
      break;
   .
   .
   .
   default:
      satser
      break;
}
```