

Programmering i ANSI-C/Felhantering

Från Wikibooks

< Programmering i ANSI-C

Innehåll

- 1 Felhantering
 - 1.1 Feltester
 - 1.2 Felmeddelanden i text
 - 1.3 Felkoder

Programmering i ANSI-C

Källkoden|Villkorssatser|Preprocessorn|Kompilatorn|Nyckelord|Standardströmmarna

Filhantering|**Felhantering**|Programexempel|Standardbibliotek|Tabeller

Felhantering

Feltester

Många av funktioner som ingår i standardbiblioteken returnerar felkoder om något inte har fungerat som det är tänkt. Ett välskrivet program tar hänsyn till detta. För att uppnå det så kan feltester göras genom att anropet kapslas i en **if**-sats som innehåller ett jämförelsevillkor. Om villkoret visar sig vara SANT så har ett fel uppstått och programmet utför koden inuti satsen, om inte så kommer programmet i stället att fortsätta med koden som kommer efter **if**-satsen. Exempel:

```
/* Testa om ett fel uppstått */
if ( FEL == funktionsanrop ( parametrar ) )
{
    kod som bara utförs om ett fel uppstått
}
kod som alltid utförs
```

Ofta är det så att funktionerna efter feltestet inte kan utföras om ett fel uppstått. Dom kan till exempel vara beroende av resultatet från funktionen som gav felet. Det normala då är att programmet inte längre kan fortsätta och omedelbart måste avbrytas. Det görs med funktionen "abort" som är definierad på följande sätt:

```
void abort ( void );
```

Funktionen returnerar omedelbart till processen som startade programmet och ger då returvärdet = 3 (*exit code 3*).

Innan programmet avbryts så bör även ett felmeddelande skrivas ut så att användaren förstår vad som skett, (se "Felmeddelanden i text" nedan). Här är ett exempel med felmeddelande samt programslut:

```
/* Avbryt om ett fel uppstått */  
if ( FEL == funktionsanrop ( parametrar ) )  
{  
    perror ( "Nu gick det tokfel!" );  
    abort ( );  
}  
  
kod som alltid utförs om ett fel inte uppstått
```

Felmeddelanden i text

För att skriva ut ett felmeddelande till felmeddelandeströmmen "stderr" används funktionen "perror" som är definierad på följande sätt:

```
void perror ( const char *sträng );
```

Funktionen skriver ut en ledtext¹ följt av standardfelmeddelandet som representeras av det aktuella värdet på den globala variabeln "errno". Felmeddelandefunktionen returnerar inte någon felkod om något gått fel =)

¹ Ursprungligen var tanken är att sökvägen till filen med felet skall skickas som parameter till funktionen men det går bra med vilken text som helst.

```
/* Skriv felmeddelande till "stderr" */  
  
errno = 5;  
perror ( "Nu gick det tokfel!" );  
  
Ger följande utskrift:  
Nu gick det tokfel!: Permission denied
```

Om ett annat värde än "5" valts för variabeln "errno" så hade meddelandet varit något annat. I normala fall sätts "errno" till lämpligt värde av den funktion som senast gav ett fel.

Vill man bearbeta felmeddelandetexten innan den skrivs ut så kan funktionen "strerror" användas. Den returnerar en pekare till en sträng som innehåller felmeddelandet. Vid anropet anges felkoden som parameter till funktionen. Funktionen definieras på detta sätt:

```
char *strerror( int felkod )
```

Om en funktion har returnerat en felindikator så sätts den globala variabeln "errno" till ett värde som representerar felet. Då kan strängen med det aktuella felmeddelandet (se lista nedan) hämtas med hjälp av "strerror".

```
/* Hämta felmeddelandesträng */  
  
char *textpekare;  
  
textpekare = strerror( errno );
```

Sätter "textpekare" att peka på strängen med det felmeddelande som motsvarar det aktuella värdet på "errno".

Felkoder

När en funktion returnerar en felindikator så anges felkoden i den globala variabeln "errno". Det värdet kan sedan användas för att skriva ut passande felmeddelanden till felmeddelandeströmmen "stderr".

Felkoder vid filhantering:

Felkod	Felmeddelande
EACCES	Permission denied.
EEXIST	Permission denied: file already exists.
ENOENT	No such file or directory.
ENOTSAM	Not same device.

O.B.S. Tabellen är inte komplett ännu.

Hämtad från "http://sv.wikibooks.org/wiki/Programmering_i_ANSI-C/Felhantering"

Kategori: Programmering i ANSI-C

- Sidan ändrades senast den 2 maj 2010 kl. 22.26.
- Text är tillgänglig under Creative Commons Erkännande-Dela Lika-licens; ytterligare villkor kan gälla. Se Villkor för detaljer.
- Integritetspolicy
- Om Wikibooks
- Förbehåll