Code-Lite ”tutorial” (2013-03-20/RoJ)

Följ dessa anvisningar för att skapa projekt, kompilera/länka och testa dina laborationsuppgifter 3,4.

- ”Project” – Projekt, innehåller bland annat ett antal källtextfiler som krävs för att skapa en exekverbar fil.
- ”Workspace” – Arbetsutrymme, kan innehålla ett antal olika projekt.

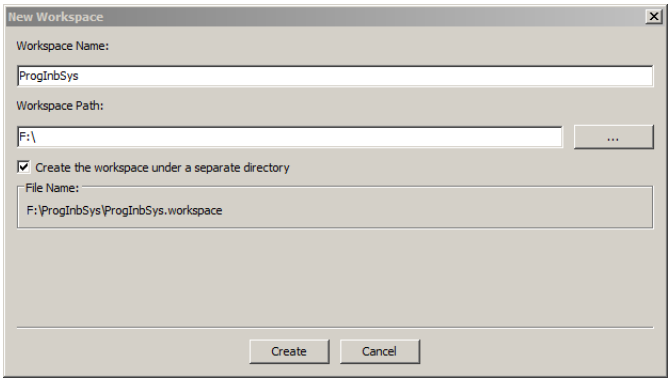
Skapa ett nytt arbetsutrymme

Då du skapar ett nytt arbetsutrymme är det viktigt att du tänker på sökvägen (”Workspace Path”). Välj en lämplig plats i din hemkatalog där det nya arbetsutrymmet blir en ”rotkatalog” för dina laborationer i kursen. Ett lämpligt namn kan förslagsvis vara ”ProgInbSys” (Programmering av inbyggda system).

Skapa ett nytt arbetsutrymme, välj från menyn:

Workspace | New Workspace

Välj en lämplig plats (”Workspace Path”) och ett lämpligt namn (”Workspace Name”):

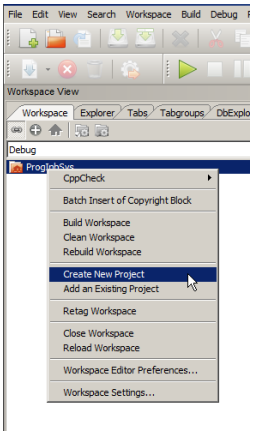


Klicka på Create för att skapa arbetsutrymmet.

Skapa ett nytt projekt

Högerklicka på det nya arbetsutrymmets ikon i ”Workspace”-fliken, välj

Create New Project



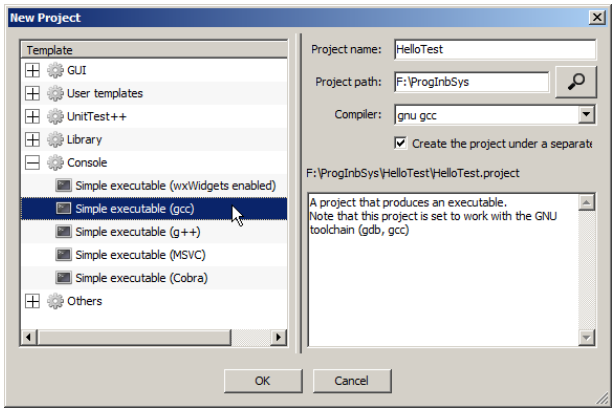
”New Project”-dialogen öppnas:
Välj

Console | Simple executable (gcc)

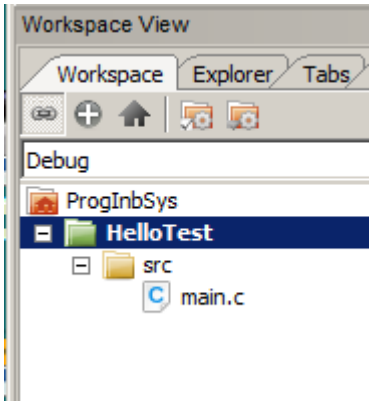
Ge projektet ett namn, som exempel

`HelloTest`

Klicka OK.



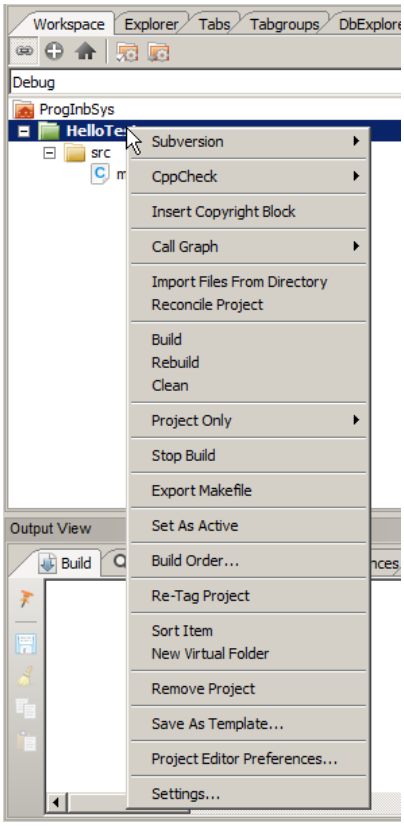
Projektet `HelloTest` skapas
expandera dess ikon i ”Workspace”-fliken:



Här har du nu fått en ny fil, `main.c` att börja ditt projekt med.

Kompilera och länka

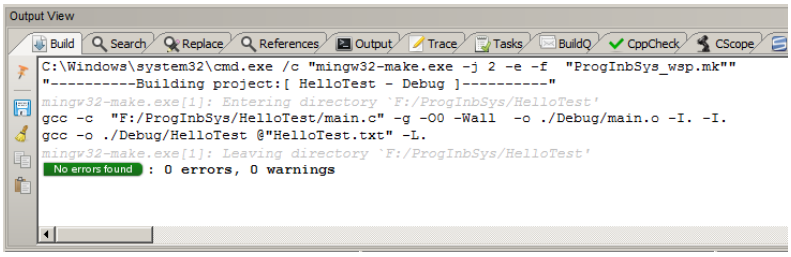
Högerklicka då projektnamnet är valt:



Via Popup-menyn kan du göra en mängd olika saker, som att lägga till ytterligare filer i projektet, strukturera vyn med hjälp av virtuella foldrar (påverkar ej filernas placering på disken...) etc.

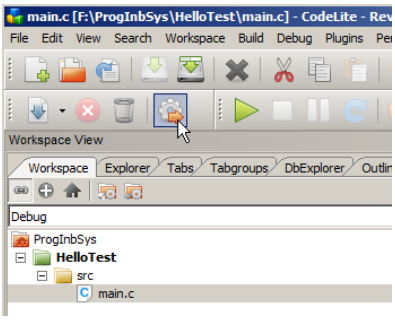
Alternativet Build används för att kompilera nödvändiga filer i projektet till objektfiler (.o) för att avslutningsvis länka samman objektfiler med standardbibliotek till en exekverbar fil (.exe). Alternativet Clean tar bort samtliga objektfiler och Rebuild kompilerar om samtliga källtextfiler.

För att nu skapa en exekverbar fil väljer du Build från menyn, ”Build”-fönstret längst ned visar resultatet.

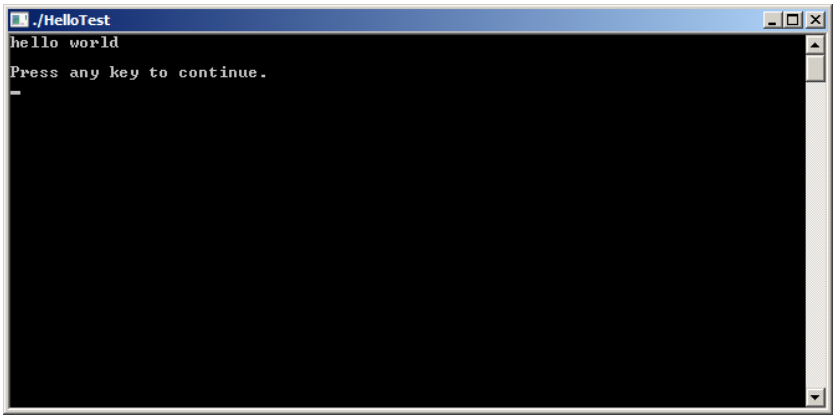


Provkör programmet

För att köra programmet klickar du på ikonen RunActiveProject i verktygslisten.



Ett konsolfönster skapas automatiskt (vi valde ju denna projekttyp) och programmets utskrift skickas till detta fönster...



Kontrollera att utskriften från programmet är den förväntade.

Test, felsökning och felavhjälpning

Börja med att skapa ett nytt projekt `FibonacciTest` i arbetsutrymmet.
Ersätt filen `main.c` med en ny fil `Fibonacci.c`
För detta exempel använder vi nu kod hämtad från:

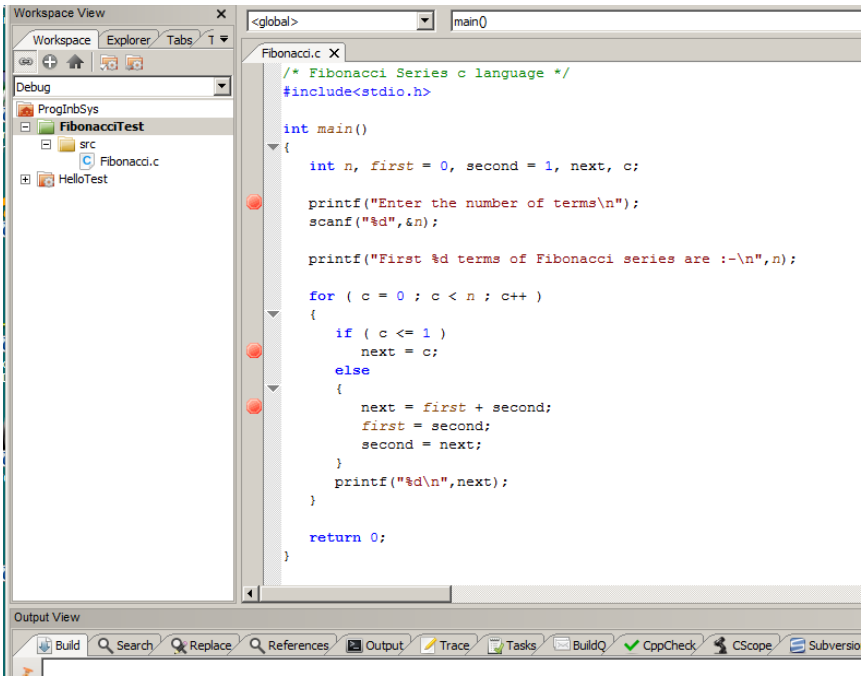
http://www.programmingsimplified.com/c-program-generate-fibonacci-series

Hittar du inte detta kan du klippa koden härifrån:

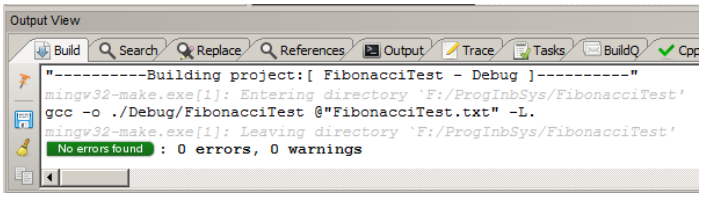
`Fibonacci.c:`

```
/* Fibonacci Series c language */
#include<stdio.h>

int main()
{
   int n, first = 0, second = 1, next, c;

   printf("Enter the number of terms\n");
   scanf("%d",&n);

   printf("First %d terms of Fibonacci series are :-\n",n);

   for ( c = 0 ; c < n ; c++ )
   {
      if ( c <= 1 )
         next = c;
      else
      {
         next = first + second;
         first = second;
         second = next;
      }
      printf("%d\n",next);
   }
   return 0;
}
```

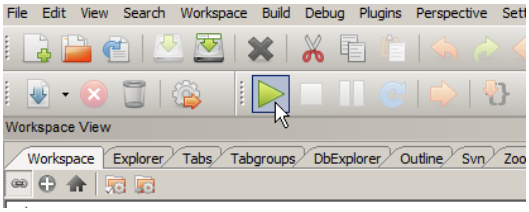
Placera markören i det grå fältet i texteditorns vänstra kant och högerklicka, välj AddBreakpoint för att sätta ut en brytpunkt på raden. Följande bild visar hur vi satt brytpunkter på tre strategiska ställen i programmet:



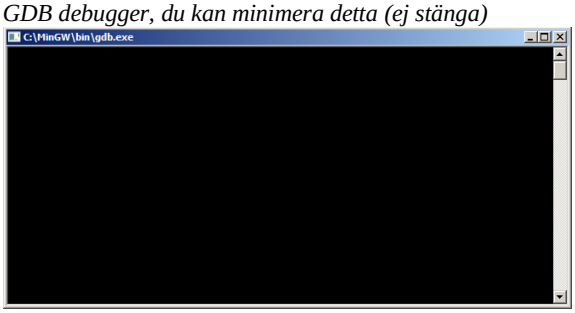
Kompilera och länka projektet `FibonacciTest` (Build). Om du får felmeddelanden, rätta till och kompilera igen.



Starta nu debuggern genom att klicka på den gröna startikonen.
Två konsolfönster skapas, ett för debuggern (gdb.exe) och ett för applikationen (Fibonaccitest.exe), det är det senare fönstret vi använder för in- och utmatning med vårt testprogram.

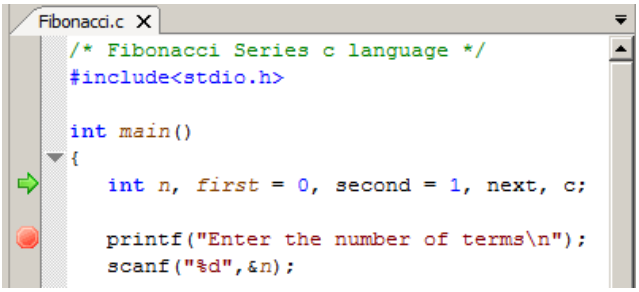


GDB debugger, du kan minimera detta (ej stänga)

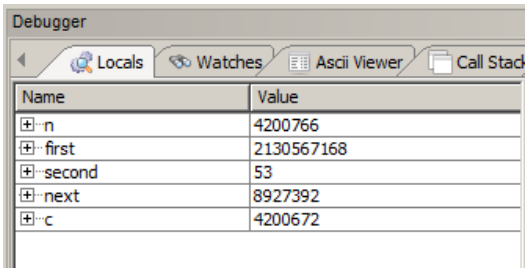


Applikation, in- och utmatning för ditt program

Den gröna pilen visar exekveringspunkten, dvs. den rad i programmet, som står i tur att utföras, i detta fall programmets första exekverbara sats:

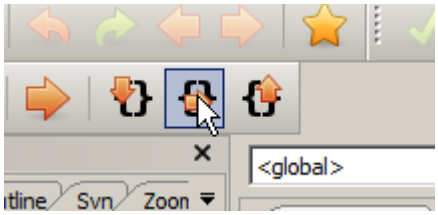


Observera hur de lokala variablerna (Locals) har ”nonsensvärden”, det kan vara helt andra värden på din skärm, variablerna är än så länge oinitierade, detta görs av programmets första sats.



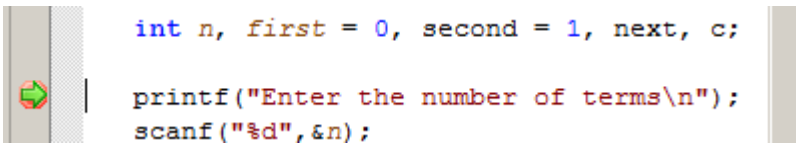
De tre ikonerna StepIn, Next och StepOut används för att utföra delar av programmet.

- Next: Utför nästa rad, dvs. den rad den gröna pilen pekar på.
- StepIn: Om nästa exekveringspunkt är en funktion så följ programmet in i funktionen. Använd Next om du vill utföra hela funktionen.
- StepOut, utför hela funktionen

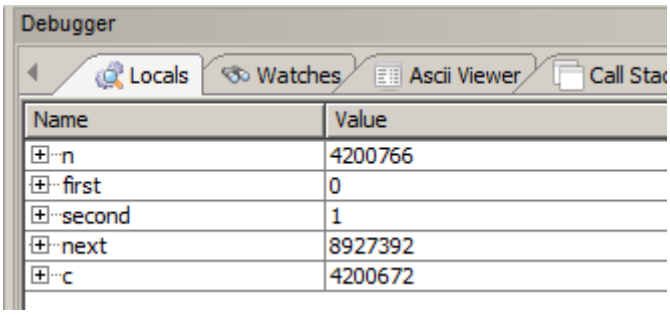


Här är det lämpligt att stega med Next:

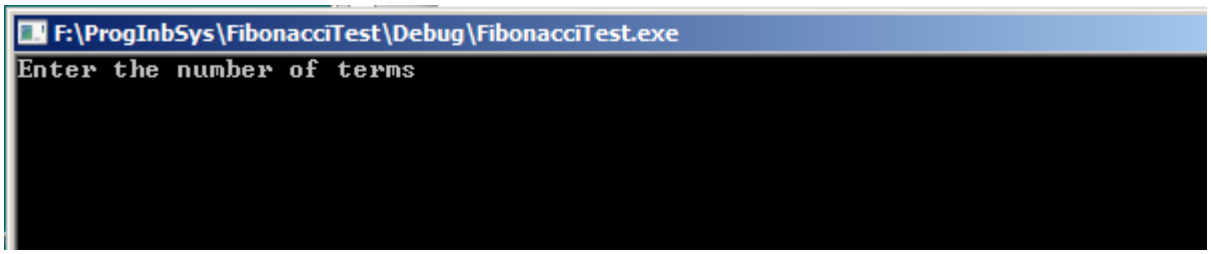
Den första raden, dvs. tilldelningarna av first och second utförs, exekveringspunkten flyttas till nästa rad:



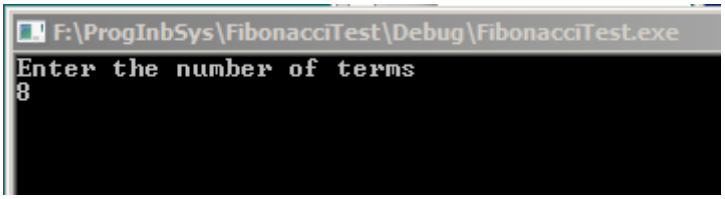
Observera hur fönstret med variabler uppdateras:



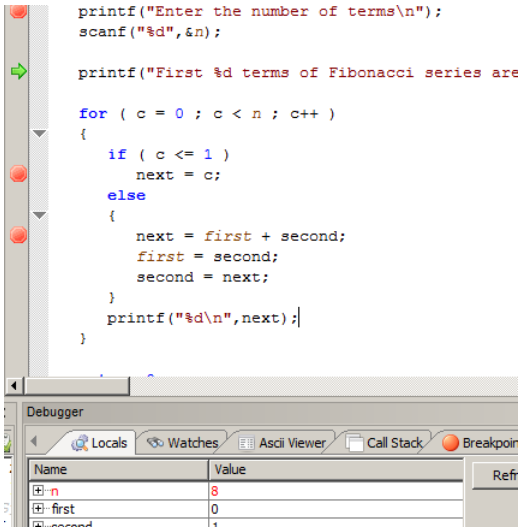
Utför nu hela `printf`-satsen, exekveringspunkten flyttas till nästa rad, *som är en inmatningssats* (`scanf`). Växla till konsollfönstret och observera programmets utskrift:



Klicka nu ytterligare en gång på Next-ikonen, dvs. utför raden med `scanf`-satsen, observera att den gröna pilen med nästa exekvering-spunkt försvinner, det beror på att programmet väntar på inmatning, ge ett heltal, exempelvis 8 och tryck därefter <Enter> för att avsluta inmatningen.

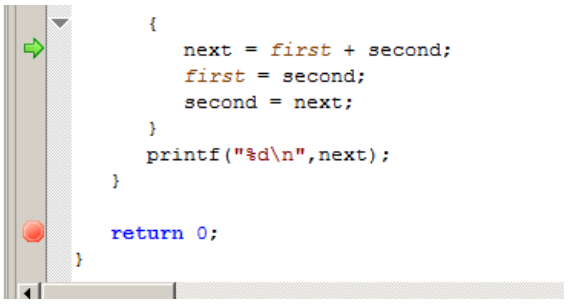


Debuggern tar nu tillbaks kontrollen, exekveringspunkten placeras på nästa `printf`-sats. Observera också hur variabeln **n**, som tilldelas i `scanf`, uppdateras.



Som nästa steg kan du prova att exekvera programmet till nästa brytpunkt, klicka än en gång på den gröna startikonen (Start or continue debugger)

- Inspektera de lokala variablernas värden
- ta bort brytpunkten programmet stannade på genom att vänsterklicka på den orange markeringen (markeringen försvinner)
- fortsätt därefter till nästa brytpunkt, ta även bort denna brytpunkt,
- sätt ut en ny brytpunkt på programmets sista rad (`return 0`) genom att vänsterklicka i det grå fältet mitt för raden, en ny brytpunktsmarkering ska då visas.



Tips inför laborationerna

Du kan redan nu skapa projekt för de kommande laborationerna i ditt arbetsutrymme.
Tänk på att laboration 3 omfattar två olika exekverbara program och det kan därför vara lämpligt att skapa två olika projekt för den laborationen. Tänk också på att samma filer mycket väl kan ingå i olika projekt, skapa inga onödiga kopior och därmed olika versioner av dina källtextfiler. Källtextfilerna till laboration 3 skapar du helt och hållet själv.

Källtextfiler för laboration 4 finns delvis på resurssidan. Kopiera dessa och lägg i ett nytt projekt.