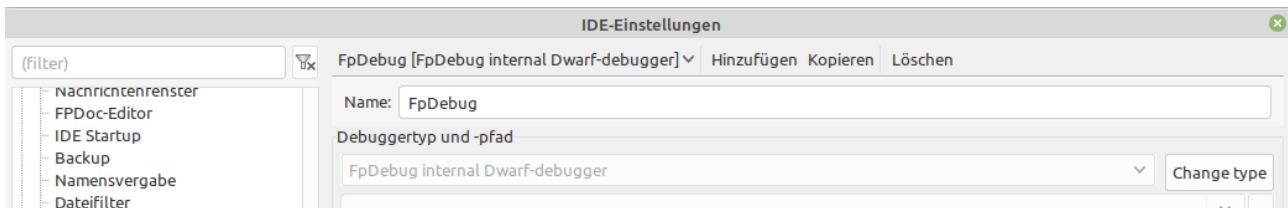
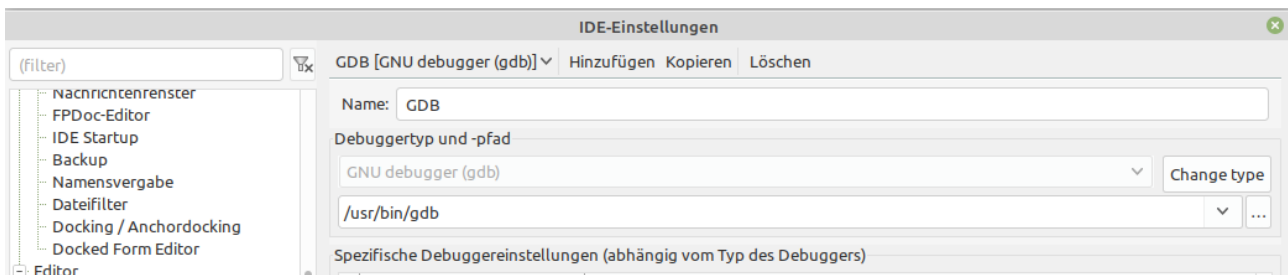


## IDE Einstellungen

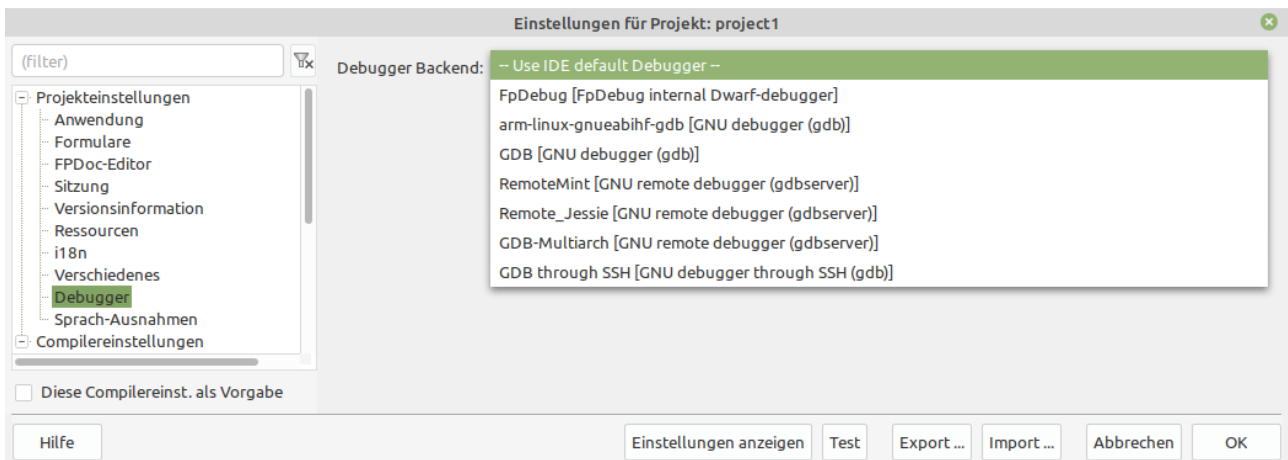
In Lazarus 2.2.0 ist standardmäßig FPDebug voreingestellt, während in älteren Versionen GDB (GnuDebugger) verwendet wird. Die globale Einstellung (also was bei einem neuen Projekt verwendet wird) findet man unter Werkzeuge, Einstellungen, Debugger, Debugger backend.



Für GDB sieht die Einstellung so aus:



Möchte man für ein bestimmtes Projekt eine andere Einstellung dann kann man unter Projekt, Projekteinstellungen eine andere Auswahl treffen ohne die globale Einstellung zu verändern.



Damit der Debugger verwendet werden kann muss unter Projekt, Projekteinstellungen, Debuggen die Option Generate info for the debugger (slower/increases exe-size) aktiviert sein. Im Default Erstellmodi ist dies der Fall.

Wer sich in diesem Menü nicht selber seine Einstellungen zusammenstellen möchte, für den bietet Lazarus einen einfachen Weg einen Debug- und einen Releasemode zu erstellen. Siehe Debug und Releas Modes erstellen. Der Debugmode bietet, wie der Name schon sagt Einstellungen zum Debuggen, der Releasemode wird für das fertige Projekt verwendet. Kompilieren im Debugmode dauert länger und die fertige Executable wird größer.

<http://gcc.gnu.org/onlinedocs/gcc/Debugging-Options.html>

## Haltepunkt (Breakpoint) setzen.

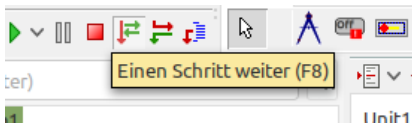
Haltepunkte setzt man am einfachsten durch einen Mausklick links neben die Zeilenzahl. Angezeigt wird er durch einen roten Kreis mit einem weißen Fragezeichen. Wirksam wird er nur wenn im Debugmodus gearbeitet wird.

```
75 s := INTCOST(1710);
76 c := RGB(random(255), random(255), random(255));
if b then b:=false else b:= true;
Output:
```

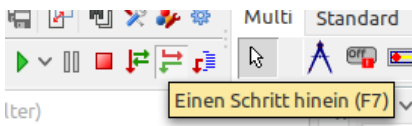
Ist er gesetzt und man Startet das Programm mit F9 läuft das Programm bis zum Haltepunkt und bleibt dort stehen. Die Zeile wird grau. Fährt man mit der Maus nun über eine Variable wird deren Wert angezeigt.

```
inc(i);
s := i = 1
75 c := private var TForm1.i: integer = longint
76 if b := _/home/bernd/Sprachen/Lazarus/64_bit/Test/DebugTest/Testproject/unit1.pas(23,5)_
Output:
```

Will man eine Zeile weiter drückt man F8 oder klickt:



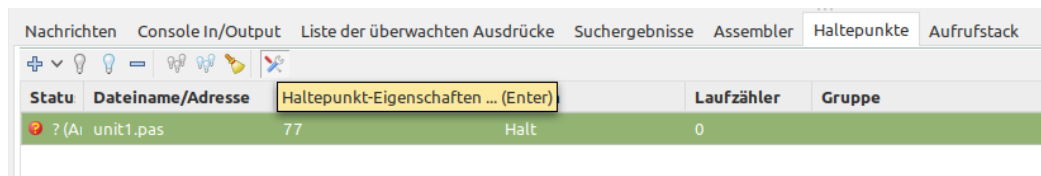
Mit F7 kann man in andere Units hinein debuggen (Umschalt +F8 wieder heraus)



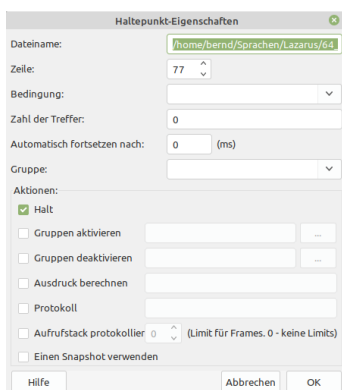
Um das Programm fortzusetzen F9 Start drücken.

## Haltepunkt Eigenschaften

Unter Ansicht, Debugger Fenster, Haltepunkte können Einstellungen für Haltepunkte vorgenommen werden. Das Fenster ist bei mir angedockt. Den Haltepunkt in der Liste markieren!



Klickt man auf das Eigenschaftsicon öffnet sich dieses Fenster:

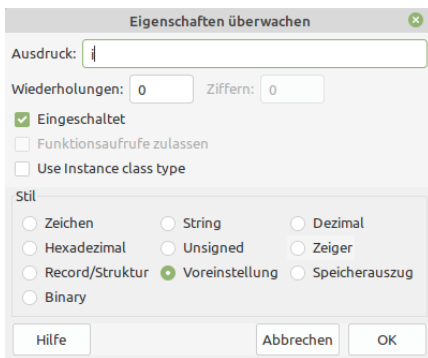


Trägt man bei Automatisch fortsetzen nach z. Bsp. 1000 ein wird das Programm nach 1 sec automatisch fortgesetzt. Trägt man bei Bedingung z. Bsp.  $i=2$  ein Hält das Programm nur an wenn  $i$  den Wert 2 besitzt. Soll der Haltepunkt beim 5ten Durchlauf wirksam sein muss man bei Zahl der Treffer eine 5 eingeben.

[https://wiki.lazarus.freepascal.org/IDE\\_Window:\\_Breakpoints](https://wiki.lazarus.freepascal.org/IDE_Window:_Breakpoints)

## Liste der überwachten Ausdrücke (Watches)

Unter Ansicht, Debugger Fenster, Überwachte Ausdrücke kann man sich die Werte von Variablen anzeigen lassen. Das Fenster ist bei mir angedockt. Zuerst drückt man auf das + Zeichen dann öffnet sich dieses Fenster:

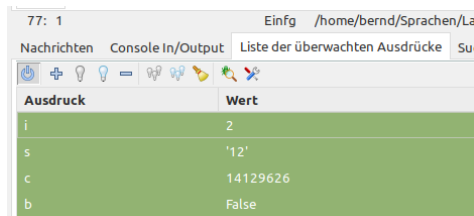


Hier gibt man bei Ausdruck die gewünschte Variable ein z. Bsp. i

Schaut dann so aus:

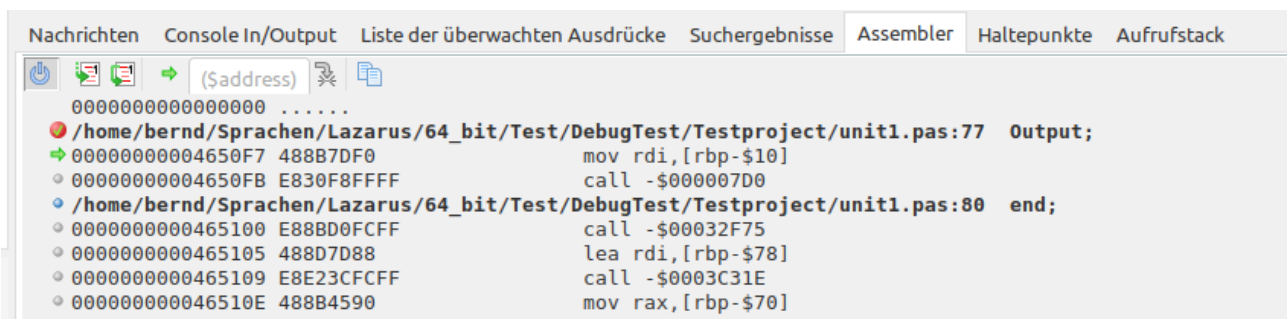
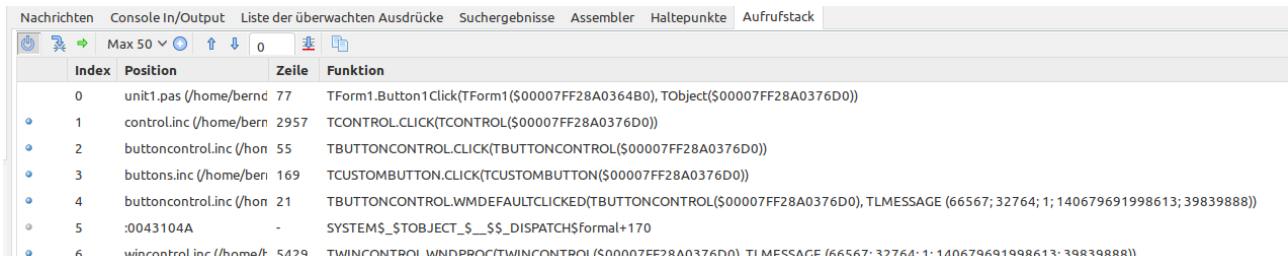


Startet man nun das Programm und es kommt am Haltepunkt zur Unterbrechung werden die Werte angezeigt:



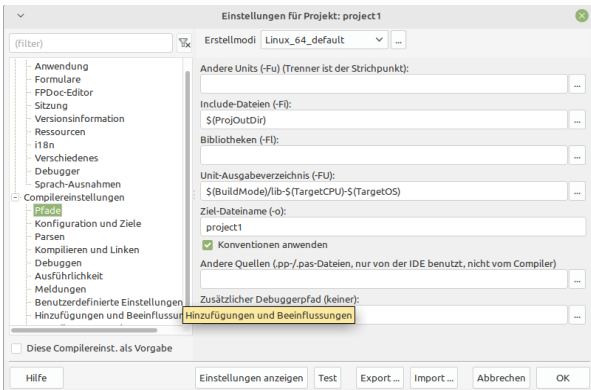
## Aufrufstack, Assemblerfenster

Unter Ansicht, Debugger Fenster findet man ebenfalls diese beiden Fenster. Eine Ausgabe erfolgt beim Erreichen eines Haltepunktes.



# Debug und Releas Modes erstellen

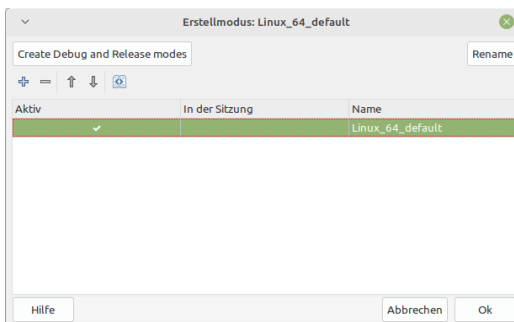
[https://wiki.freepascal.org/IDE\\_Window:\\_Compiler\\_Options/de#Hinzuf.C3.BCgen\\_eines\\_Release-\\_und\\_eines\\_Debugmodus](https://wiki.freepascal.org/IDE_Window:_Compiler_Options/de#Hinzuf.C3.BCgen_eines_Release-_und_eines_Debugmodus)



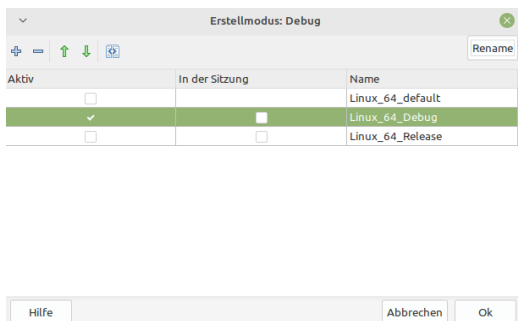
siehe: Debugger

Zuerst Projekt, dann Projekteinstellungen anklicken. Dort unter Compilereinstellungen auf Pfade klicken.

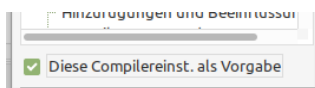
Jetzt oben bei Erstellmodi, hinter Linux\_64\_default auf die drei ... klicken.



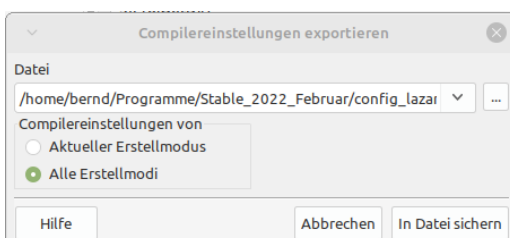
Jetzt auf Create Debug and Releas modes klicken.



Wenn man möchte Modis mit Rename umbenennen.



Damit diese Einstellung als Standard verwendet wird (falls gewünscht) bei Diese Compilereinst. als Vorgabe einen Haken setzen und mit Ok verlassen.



Dann die erzeugten Modi wie oben beschrieben sichern (Default\_Debug\_Release). Damit alle drei Modi in der Datei gesichert werden muss Alle Erstellmodi ausgewählt sein! ► Projekteinstellungen sichern