david@david: ~/projs/IVS_pM/src

```
david@david:~/projs/IVS_pM/src$ gdb our_math_debug
GNU gdb (Ubuntu 8.3-0ubuntu1) 8.3
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from our_math_debug...
(gdb) break main
Breakpoint 1 at 0x11e9: file our_math_main.cpp, line 4.
(gdb) run
Starting program: /home/david/projs/IVS_pM/src/our_math_debug

Breakpoint 1, main () at our_math_main.cpp:4
4       {
(gdb) watch result
Hardware watchpoint 2: result
(gdb) s
5           long double result = 0;
(gdb)

Hardware watchpoint 2: result

Old value = 5.1301622664024555 4018e-4937
New value = 0
main () at our_math_main.cpp:7
7           result = add(5, 6);
(gdb)
add (x=5, y=6) at our_math.h:17
17      {
(gdb)
18          return x + y;
(gdb)
19      }
(gdb)

Hardware watchpoint 2: result

Old value = 0
New value = 11
main () at our_math_main.cpp:9
9           result = sub(8, 10);
(gdb)
sub (x=8, y=10) at our_math.h:22
22      {
(gdb)
```

```
New value = 11
main () at our_math_main.cpp:9
9           result = sub(8, 10);
(gdb)
sub (x=8, y=10) at our_math.h:22
22      {
(gdb)
23          return x - y;
(gdb)
24      }
(gdb)

Hardware watchpoint 2: result

Old value = 11
New value = -2
main () at our_math_main.cpp:11
11          result = mul(2, 50);
(gdb)
mul (x=2, y=50) at our_math.h:27
27      {
(gdb)
28          return x * y;
(gdb)
29      }
(gdb)

Hardware watchpoint 2: result

Old value = -2
New value = 100
main () at our_math_main.cpp:13
13          result = div_(8.45, 5);
(gdb)
div_ (x=8.4499999999999928946, y=5) at our_math.h:32
32      {
(gdb)
33          return x / y;
(gdb)
34      }
(gdb)

Hardware watchpoint 2: result

Old value = 100
New value = 1.6899999999999985791
main () at our_math_main.cpp:15
15          result = factorial(15);
(gdb)
factorial (n=0) at our_math.h:42
42      {
(gdb)
43          if (n < 0)
(gdb)
46          if (n == 0 || n == 1 )
```

david@david: ~/projs/IVS_pM/src

```
(gdb)
factorial (n=0) at our_math.h:42
42      {
(gdb)
43          if (n < 0)
(gdb)
46          if (n == 0 || n == 1 )
(gdb)
52          long double x = (long double)n;
(gdb)
53          for (unsigned i = n; i > 1; --i, x*= i);
(gdb)
57          return x;
(gdb)
58      }
(gdb)

Hardware watchpoint 2: result

Old value = 1.68999999999999985791
New value = 1307674368000
main () at our_math_main.cpp:17
17          result = xpow(4, 1.5);
(gdb)
xpow (base=4, exponent=1.5) at our_math.cpp:25
25      {
(gdb) n
26          if (base < EPS && base > -EPS)
(gdb)
29          if (exponent < EPS && exponent > -EPS)
(gdb)
32          if (exponent >= 1 - EPS && exponent <= 1 + EPS)
(gdb)
36          if (exponent < 0)
(gdb)
43          long long natural = (long long)exponent;
(gdb)
44          long double rest = exponent - (long double)natural;
(gdb)
45          long double tmp = base;
(gdb)
48          if (natural)
(gdb)
49              while(--natural)
(gdb)
56          if (rest > XPOW_EPS - 0.00001)
(gdb)
59              long double top = 1;
(gdb)
60              long double bot = 2;
(gdb)
61              long double delta = rest - 0.5;
(gdb)
64              while (delta > XPOW_EPS - 0.00001 || delta < -XPOW_EPS + 0.00001)
(gdb)
```

david@david: ~/projs/IVS_pM/src

```
(gdb)
60              long double bot = 2;
(gdb)
61              long double delta = rest - 0.5;
(gdb)
64              while (delta > XPOW_EPS - 0.00001 || delta < -XPOW_EPS + 0.00001)
(gdb)
74              long double n_res = tmp = nroot(tmp, bot);
(gdb)
76              while (--top)
(gdb)
80              return base * n_res;
(gdb)
86      }
(gdb) s

Hardware watchpoint 2: result

Old value = 1307674368000
New value = 8
main () at our_math_main.cpp:19
19          result = npow(5, 3);
(gdb)
npow (x=5, n=4294983680) at our_math.h:67
67      {
(gdb) n
68          if (n == 0)
(gdb)
70          long double hold = x;
(gdb)
71          if (x >= 1 - EPS && x <= 1 + EPS)
(gdb)
73          while (--n) x *= hold;
(gdb)
75          return x;
(gdb)
76      }
(gdb)

Hardware watchpoint 2: result

Old value = 8
New value = 125
main () at our_math_main.cpp:21
21          result = nroot(8, 3);
(gdb) s
nroot (x=8, n=0) at our_math.h:86
86      {
(gdb)
87          if (n == 0)
(gdb)
89          if (n == 1)
(gdb)
91          if (x >= 1 - EPS && x <= 1 + EPS)
(gdb) n
```

david@david: ~/projs/IVS_pM/src

```
(gdb)
87          if (n == 0)
(gdb)
89          if (n == 1)
(gdb)
91          if (x >= 1 - EPS && x <= 1 + EPS)
(gdb) n
94          if (x < 0 && !(n&1))
(gdb)
97          if (x >= -EPS && x <= EPS)
(gdb)
101         long double delta = x;
(gdb)
102         long unsigned i = 1;
(gdb)
104         while ((delta /= 10) > 1 && ++i);
(gdb)
106         long double guess = 1;
(gdb)
107         if (i / n)
(gdb)
116         delta = 0;
(gdb)
120             guess += delta;
(gdb)
121             delta = (((x/npow(guess, (n-1)))-guess)/n);
(gdb)
123         while(delta > EPS || delta < -EPS);
(gdb)
120             guess += delta;
(gdb)
121             delta = (((x/npow(guess, (n-1)))-guess)/n);
(gdb)
123         while(delta > EPS || delta < -EPS);
(gdb)
118         do
(gdb)
120             guess += delta;
(gdb)
121             delta = (((x/npow(guess, (n-1)))-guess)/n);
(gdb)
123         while(delta > EPS || delta < -EPS);
(gdb)
118         do
(gdb)
120             guess += delta;
(gdb)
121             delta = (((x/npow(guess, (n-1)))-guess)/n);
(gdb)
123         while(delta > EPS || delta < -EPS);
(gdb)
118         do
(gdb)
120             guess += delta;
(gdb)
```

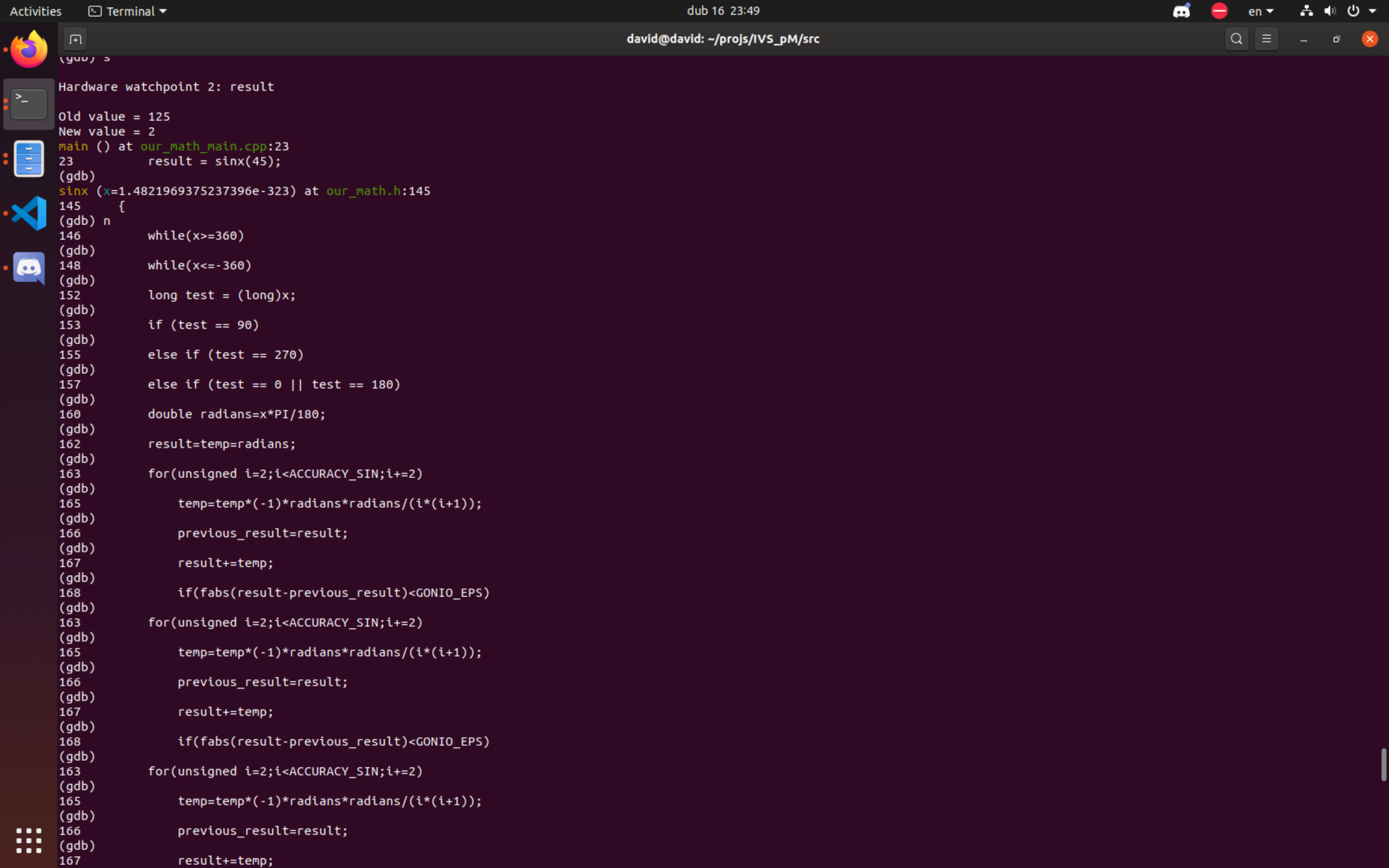
Activities   Terminal   dub 16 23:49   en

david@david: ~/projs/IVS_pM/src

```
(gdb)
120             guess += delta;
(gdb)
121             delta = (((x/npow(guess, (n-1)))-guess)/n);
(gdb)
123         while(delta > EPS || delta < -EPS);
(gdb)
118         do
(gdb)
120             guess += delta;
(gdb)
121             delta = (((x/npow(guess, (n-1)))-guess)/n);
(gdb)
123         while(delta > EPS || delta < -EPS);
(gdb)
118         do
(gdb)
120             guess += delta;
(gdb)
121             delta = (((x/npow(guess, (n-1)))-guess)/n);
(gdb)
123         while(delta > EPS || delta < -EPS);
(gdb)
118         do
(gdb)
120             guess += delta;
(gdb)
121             delta = (((x/npow(guess, (n-1)))-guess)/n);
(gdb)
123         while(delta > EPS || delta < -EPS);
(gdb)
118         do
(gdb)
120             guess += delta;
(gdb)
121             delta = (((x/npow(guess, (n-1)))-guess)/n);
(gdb)
123         while(delta > EPS || delta < -EPS);
(gdb)
124         return guess+delta;
(gdb)
125     }
(gdb) s

Hardware watchpoint 2: result

Old value = 125
New value = 2
main () at our_math_main.cpp:23
23          result = sinx(45);
(gdb)
sinx (x=1.4821969375237396e-323) at our_math.h:145
145     {
(gdb) n
146         while(x>=360)
```


Activities
Terminal
dub 16 23:49
en
david@david: ~/projs/IVS_pM/src
(gdb) s
Hardware watchpoint 2: result
Old value = 125
New value = 2
main () at our_math_main.cpp:23
23   result = sinx(45);
(gdb)
sinx (x=1.4821969375237396e-323) at our_math.h:145
145   {
(gdb) n
146   while(x>=360)
(gdb)
148   while(x<=-360)
(gdb)
152   long test = (long)x;
(gdb)
153   if (test == 90)
(gdb)
155   else if (test == 270)
(gdb)
157   else if (test == 0 || test == 180)
(gdb)
160   double radians=x*PI/180;
(gdb)
162   result=temp=radians;
(gdb)
163   for(unsigned i=2;i<ACCURACY_SIN;i+=2)
(gdb)
165   temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
166   previous_result=result;
(gdb)
167   result+=temp;
(gdb)
168   if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
163   for(unsigned i=2;i<ACCURACY_SIN;i+=2)
(gdb)
165   temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
166   previous_result=result;
(gdb)
167   result+=temp;
(gdb)
168   if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
163   for(unsigned i=2;i<ACCURACY_SIN;i+=2)
(gdb)
165   temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
166   previous_result=result;
(gdb)
167   result+=temp;

david@david: ~/projs/IVS_pM/src

```
(gdb)
165             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
166             previous_result=result;
(gdb)
167             result+=temp;
(gdb)
168             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
163         for(unsigned i=2;i<ACCURACY_SIN;i+=2)
(gdb)
165             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
166             previous_result=result;
(gdb)
167             result+=temp;
(gdb)
168             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
169                 break;
(gdb)
171         return result;
(gdb)
172     }
(gdb) s

Hardware watchpoint 2: result

Old value = 2
New value = 0.70710678293686708701
main () at our_math_main.cpp:25
25          result = cosx(60);
(gdb)
cosx (x=45) at our_math.h:190
190     {
(gdb)
191         while(x>=360)
(gdb) n
193         while(x<=-360)
(gdb)
196         long test = (long)x;
(gdb)
197         if (test == 0)
(gdb)
199         else if (test == 180)
(gdb)
201         else if (test == 90 || test == 270)
(gdb)
204         double radians=x*PI/180;
(gdb)
206         result=temp=1;
(gdb)
207         for(unsigned i = 1; i < ACCURACY_COS; i += 2)
(gdb)
```

Activities Terminal dub 16 23:49 en

david@david: ~/projs/IVS_pM/src

```
New value = 0.70710678293080708701
main () at our_math_main.cpp:25
25          result = cosx(60);
(gdb)
cosx (x=45) at our_math.h:190
190     {
(gdb)
191         while(x>=360)
(gdb) n
193         while(x<=-360)
(gdb)
196         long test = (long)x;
(gdb)
197         if (test == 0)
(gdb)
199         else if (test == 180)
(gdb)
201         else if (test == 90 || test == 270)
(gdb)
204         double radians=x*PI/180;
(gdb)
206         result=temp=1;
(gdb)
207         for(unsigned i = 1; i < ACCURACY_COS; i += 2)
(gdb)
209             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
210             previous_result=result;
(gdb)
211             result+=temp;
(gdb)
212             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
207         for(unsigned i = 1; i < ACCURACY_COS; i += 2)
(gdb)
209             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
210             previous_result=result;
(gdb)
211             result+=temp;
(gdb)
212             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
207         for(unsigned i = 1; i < ACCURACY_COS; i += 2)
(gdb)
209             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
210             previous_result=result;
(gdb)
211             result+=temp;
(gdb)
212             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
207         for(unsigned i = 1; i < ACCURACY_COS; i += 2)
(gdb)
```

Activities Terminal dub 16 23:49 en

david@david: ~/projs/IVS_pM/src

```
(gdb)
209             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
210             previous_result=result;
(gdb)
211             result+=temp;
(gdb)
212             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
207         for(unsigned i = 1; i < ACCURACY_COS; i += 2)
(gdb)
209             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
210             previous_result=result;
(gdb)
211             result+=temp;
(gdb)
212             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
207         for(unsigned i = 1; i < ACCURACY_COS; i += 2)
(gdb)
209             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
210             previous_result=result;
(gdb)
211             result+=temp;
(gdb)
212             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
207         for(unsigned i = 1; i < ACCURACY_COS; i += 2)
(gdb)
209             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
210             previous_result=result;
(gdb)
211             result+=temp;
(gdb)
212             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
207         for(unsigned i = 1; i < ACCURACY_COS; i += 2)
(gdb)
209             temp=temp*(-1)*radians*radians/(i*(i+1));
(gdb)
210             previous_result=result;
(gdb)
211             result+=temp;
(gdb)
212             if(fabs(result-previous_result)<GONIO_EPS)
(gdb)
213                 break;
(gdb)
215         return result;
(gdb)
216     }
(gdb) s
```





david@david: ~/projs/IVS_pM/src

```
(gdb)
216     }
(gdb) s

Hardware watchpoint 2: result

Old value = 0.70710678293686708701
New value = 0.499999996390943237464
main () at our_math_main.cpp:27
27          result = tanx(45);
(gdb)
tanx (x=4.635570538286745e-310) at our_math.h:234
234     {
(gdb)
235         if(!(fmod(x,90)) && (fmod(x,180)))
(gdb)
std::fmod<double, int> (__x=1.0471975511965976, __y=0) at /usr/include/c++/9/cmath:290
290         fmod(_Tp __x, _Up __y)
(gdb) n
293           return fmod(__type(__x), __type(__y));
(gdb)
294         }
(gdb)
tanx (x=45) at our_math.h:235
235         if(!(fmod(x,90)) && (fmod(x,180)))
(gdb)
237         return sinx(x)/cosx(x);
(gdb)
238     }
(gdb)

Hardware watchpoint 2: result

Old value = 0.499999996390943237464
New value = 0.99999996831695300868
main () at our_math_main.cpp:29
29          return 0;
(gdb)
30      }
(gdb)

Watchpoint 2 deleted because the program has left the block in
which its expression is valid.
__libc_start_main (main=0x5555555551e9 <main()>, argc=1, argv=0x7fffffffdee8, init=<optimized out>, fini=<optimized out>, rtld_fini=<optimized out>, stack_end=0x7fffffffded8)
    at ../csu/libc-start.c:342
342     ../csu/libc-start.c: No such file or directory.
(gdb)
[Inferior 1 (process 26268) exited normally]
(gdb)
The program is not being run.
(gdb) quit
david@david:~/projs/IVS_pM/src$
david@david:~/projs/IVS_pM/src$
david@david:~/projs/IVS_pM/src$
```

GNU nano 4.3 our_math.cpp Modified

```
/**
 * @file our_math.cpp
 * @author Mihola David
 * @author Foltyn Lukas
 * @copyright ©Pied Piper
 * @brief This source file contains extern declarations of inline functions and definition of xpow function from the math library.
*/

#include "../include/our_math.h"

extern long double add (long double x, long double y);

extern long double sub (long double x, long double y);

extern long double mul (long double x, long double y);

extern long double div_ (long double x, long double y);

extern long double cosx (double x);

extern long double sinx (double x);

extern long double tanx (double x);

extern long double npow (long double x, long unsigned n);

extern long double nroot (long double x, long unsigned n);

extern long double factorial (int n);

long double xpow (long double base, long double exponent)
{
    if (base < EPS && base > -EPS)
        return 0;

    if (exponent < EPS && exponent > -EPS)
        return 1;

    if (exponent >= 1 - EPS && exponent <= 1 + EPS)
        return base;

    //switching the values when exponent is negative
    if (exponent < 0)
    {
        base = 1 / base;
        exponent = -exponent;
    }

    //separating the number to it's natural part and decimal part
    long long natural = (long long)exponent;
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo M-A Mark Text M-] To Bracket M-Q Previous
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo M-6 Copy Text ^Q Where Was M-W Next