

Watch a global variable passively with GDB

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>
#define NUM_THREADS    3

typedef struct SomeGlobalStruct
{
    long element0;
} TheStruct;

TheStruct x;

void *ChangeGlobal(void *threadid){
    long tid;
    tid = (long)threadid;

    x.element0 = tid;

    pthread_exit(NULL);
}

int main (int argc, char *argv[])
{
    pthread_t threads[NUM_THREADS];
    int rc;
    long t;
    for(t=0; t < NUM_THREADS; t++)
    {
        rc = pthread_create(&threads[t], NULL, ChangeGlobal, (void *)t);
        if (rc){
            printf("ERROR; return code from pthread_create() is %d\n", rc);
            exit(-1);
        }
    }
    pthread_exit(NULL);

    return 0;
}
```

Compiling the source code with info for debugging

```
pianodaemon@ubuntu:/tmp$ gcc example.c -lpthread -o ./example.exe -g
Running the program in debug mode
```

```
pianodaemon@ubuntu:/tmp$ gdb example.exe
Defining a watchpoint and creating a command for it
```

```
GNU gdb (GDB) 7.5.91.20130417-cvs-ubuntu
Copyright (C) 2013 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".
For bug reporting instructions, please see:
```

```
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /tmp/example.exe...done.
(gdb) watch x
Hardware watchpoint 1: x
(gdb) info watch
Num      Type          Disp Enb Address  What
1        hw watchpoint  keep y          x
(gdb) command 1
Type commands for breakpoint(s) 1, one per line.
End with a line saying just "end".
>echo watchpoint reached!!!\n
>print x
>cont
>end
(gdb) info watch
Num      Type          Disp Enb Address  What
1        hw watchpoint  keep y          x
          echo watchpoint reached!!!\n
          print x
          cont
(gdb)
Now. We can test our monitoring

(gdb) run
Starting program: /tmp/example.exe
warning: no loadable sections found in added symbol-file system-supplied DSO at
0x7ffff7ffa000
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[New Thread 0x7ffff77f4700 (LWP 27104)]
[New Thread 0x7ffff6ff3700 (LWP 27105)]
[Switching to Thread 0x7ffff6ff3700 (LWP 27105)]
Hardware watchpoint 1: x

Old value = {element0 = 0}
New value = {element0 = 1}
ChangeGlobal (threadid=0x1) at example.c:23
23      pthread_exit(NULL);
watchpoint reached!!!
$1 = {element0 = 1}
[New Thread 0x7ffff67f2700 (LWP 27106)]
[Switching to Thread 0x7ffff67f2700 (LWP 27106)]
Hardware watchpoint 1: x

Old value = {element0 = 1}
New value = {element0 = 2}
ChangeGlobal (threadid=0x2) at example.c:23
23      pthread_exit(NULL);
watchpoint reached!!!
$2 = {element0 = 2}
[Thread 0x7ffff7fde740 (LWP 27100) exited]
[Thread 0x7ffff6ff3700 (LWP 27105) exited]
[Thread 0x7ffff67f2700 (LWP 27106) exited]
[Inferior 1 (process 27100) exited normally]
```