INGI1113 - Rappel : GDB et les threads

08 Feb 2012

Gregory Detal - Fabien Duchêne – Christoph Paasch

GDB & threads

- GDB rend disponible:
  - La notification de nouveaux threads
  - De switcher entre les threads (thread <no>)
  - D’obtenir des informations sur les threads (info threads)
  - D’appliquer une commande sur tous les threads (thread apply [no] [all] args).
  - De placer des breakpoints dans les threads

GDB & threads


```
int shared_var = 0;

void * thread_produce(void * ctx)
{
shared_var = 42;
pthread_exit(NULL);
}
```

```
void * thread_consume(void * ctx)
{
    int result;
    result = 42 / shared_var;
    printf("Result:%d\n", result);
    pthread_exit(NULL);}
}
```

```
$ ./testFloating point exception
```

GDB & threads


```
$ gdb ./test
(gdb) run
Starting program: /Users/gdetal/Desktop/test
Reading symbols for shared libraries +. done
Program received signal EXC_ARITHMETIC, Arithmetic exception.
[Switching to process 34811]
0x0000000100000e0a in thread_consume (ctx=0x0) at test.c:17
17      result = 42 / shared_var;
(gdb) info threads
   3 port# 0x30f 0x00007fffffe00295 in __spin_lock ()
* 2 port# 0x417 0x0000000100000e0a in thread_consume
(ctx=0x0) at test.c:17
   1 port# 0xa0f 0x00007fff88eeb2da in mach_msg_trap ()
(gdb) quit
```

GDB & threads


```
$ gdb ./test
(gdb) break thread_consume
Breakpoint 1 at 0x100000df8: file test.c, line 17.
(gdb) break thread_produce
Breakpoint 2 at 0x100000dd8: file test.c, line 9.
(gdb) run
[Switching to process 34854]
Breakpoint 1, thread_consume (ctx=0x0) at test.c:17
17     result = 42 / shared_var;
(gdb) thread apply 2 bt
#0 thread_consume (ctx=0x0) at test.c:17
#1 0x00007fff88f24536 in _pthread_start ()
#2 0x00007fff88f243e9 in thread_start ()
(gdb) p/d shared_var
$1 = 0
(gdb) set shared_var=42
(gdb) continue
Continuing.
Program exited normally.
```