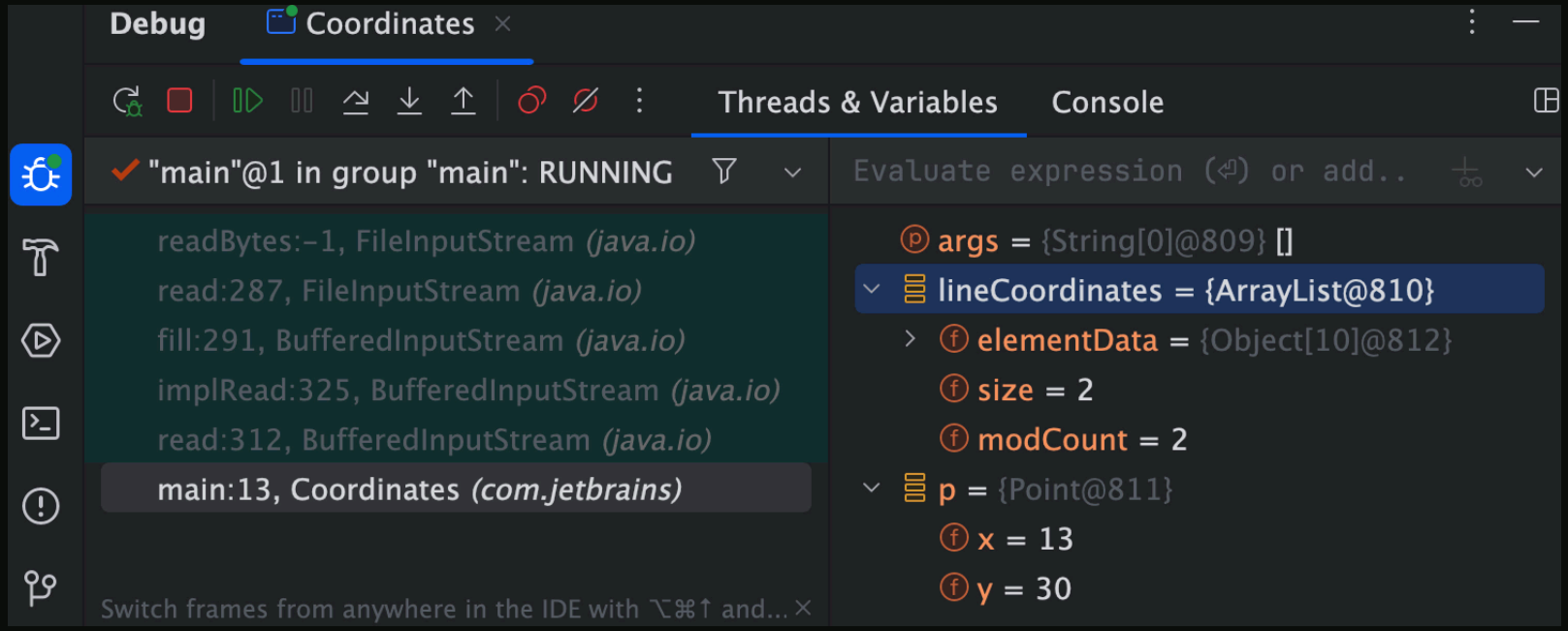
The GNU Debugger

GDB

> Ein Debugger na toll, kenn' ich schon

Sowas habt ihr bestimmt schon mal gesehen:



Und wofür jetzt ein workshop?

> Das *Terminal*

GDB, ein debugger im Terminal.
Warum würde man sich sowas antun?

```
(gdb) break 23
Breakpoint 3 at 0x5555555552a2: file multi_thread.c, line 23.
(gdb) run
Starting program: /home/debx/Projects/multi_thread
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[New Thread 0x7ffff7bff6c0 (LWP 10334)]
Thread 1 here!
[New Thread 0x7ffff73fe6c0 (LWP 10335)]
Thread 2 here!
[New Thread 0x7ffff6bfd6c0 (LWP 10336)]
Thread 3 here!
[New Thread 0x7ffff63fc6c0 (LWP 10337)]
Thread 4 here!

Thread 1 "multi_thread" hit Breakpoint 3, main () at multi_thread.c:23
23          sleep(5); // Let the threads run for 5 seconds
(gdb) info threads
  Id   Target Id                                       Frame
* 1    Thread 0x7ffff7fa6740 (LWP 10333) "multi_thread" main () at multi_thread.c:23
  2    Thread 0x7ffff7bff6c0 (LWP 10334) "multi_thread" 0x00007ffff7cdcb11 in __GI___clock_nanosleep (
    clock_id=clock_id@entry=0, flags=flags@entry=0, req=req@entry=0x7ffff7bfee80, rem=rem@entry=0x0)
    at ../sysdeps/unix/sysv/linux/clock_nanosleep.c:78
  3    Thread 0x7ffff73fe6c0 (LWP 10335) "multi_thread" 0x00007ffff7cdcb11 in __GI___clock_nanosleep (
    clock_id=clock_id@entry=0, flags=flags@entry=0, req=req@entry=0x7ffff73fde80, rem=rem@entry=0x0)
    at ../sysdeps/unix/sysv/linux/clock_nanosleep.c:78
  4    Thread 0x7ffff6bfd6c0 (LWP 10336) "multi_thread" 0x00007ffff7cdcb11 in __GI___clock_nanosleep (
    clock_id=clock_id@entry=0, flags=flags@entry=0, req=req@entry=0x7ffff6bfce80, rem=rem@entry=0x0)
    at ../sysdeps/unix/sysv/linux/clock_nanosleep.c:78
  5    Thread 0x7ffff63fc6c0 (LWP 10337) "multi_thread" 0x00007ffff7cdcb11 in __GI___clock_nanosleep (
    clock_id=clock_id@entry=0, flags=flags@entry=0, req=req@entry=0x7ffff63fbe80, rem=rem@entry=0x0)
    at ../sysdeps/unix/sysv/linux/clock_nanosleep.c:78
(gdb)
```

> CTF Challenge

“Schau mal hier ist ein Binary. Das fragt nach nem Passwort und nur wenn du das richtige Passwort eingibst, bekommst du die flag.”

> CTF Challenge

“Schau mal hier ist ein Binary. Das fragt nach nem Passwort und nur wenn du das richtige Passwort eingibst, bekommst du die flag.”

Nichts leichter als das!

> Einfacher Ansatz

`strings` command

Gibt alle Strings in einer Datei aus.

> Einfacher Ansatz

`strings` command

Gibt alle Strings in einer Datei aus.

Demo time (Bypass Me)

> Die Lösung: Debugger!

Einfach mit dem debugger durchsteppen. Zur Laufzeit sollte irgendwann das Passwort im Speicher liegen!

> Die Lösung: Debugger!

Einfach mit dem debugger durchsteppen. Zur Laufzeit sollte irgendwann das Passwort im Speicher liegen!

Aber moment mal…

> Die Lösung: Debugger!

Einfach mit dem debugger durchsteppen. Zur Laufzeit sollte irgendwann das Passwort im Speicher liegen!

Aber moment mal…

wir haben ja gar keinen source code!

> GDB ein Binary-Only Debugger

Mit GDB geht das!

> GDB ein Binary-Only Debugger

Mit GDB geht das!

Schauen wir es uns an!