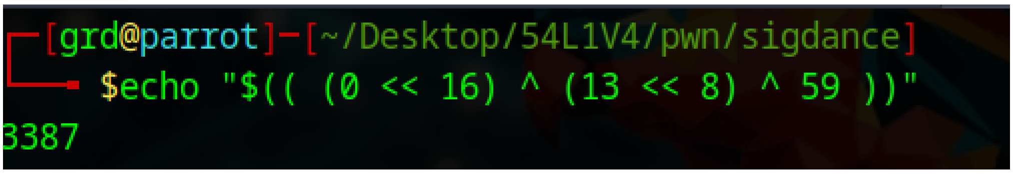
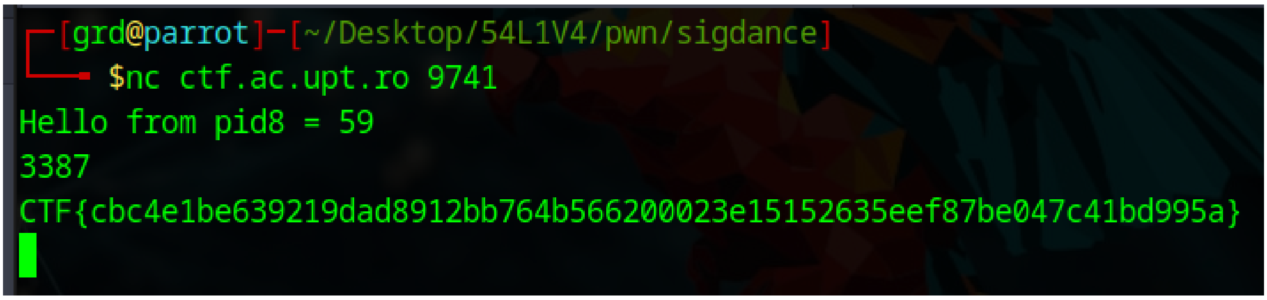
sigdance

main.c creates a new thread that repeatedly sends `SIGUSR1` signals to the main process every 5ms.
Because `SIGUSR1` arrives at 5 ms and `SIGALRM` first arrival would be at 7 ms, the `SIGUSR1` interrupts `nanosleep` immediately. The program then disables the alarm with `setitimer(..., 0)` before any `SIGALRM` can occur leaving `ac == 0`. The `SIGUSR1` sender thread still performs its loop and sends `13` signals, so `uc == 13`. The `pid & 255` is printed by the server as `pid8 = N` in the greeting.
So the expected (and predictable) final values are:

- `ac = 0`
- `uc = 13`



```
┌─[grd@parrot]─[~/Desktop/54L1V4/pwn/sigdance]
└──╼ $echo "$(( (0 << 16) ^ (13 << 8) ^ 59 ))"
3387
```



```
┌─[grd@parrot]─[~/Desktop/54L1V4/pwn/sigdance]
└──╼ $nc ctf.ac.upt.ro 9741
Hello from pid8 = 59
3387
CTF{cbc4e1be639219dad8912bb764b566200023e15152635eef87be047c41bd995a}
```