Debugging a stuck Lix invocation

If you're experiencing an issue where a `lix` command appears to hang or make no visible progress, this page outlines multiple methods to investigate what’s going on.

Step 1: Increase Verbosity

Start by rerunning the Lix command with high verbosity to get a better sense of whether it's doing useful work.

```
nix build -vvv <your-command>
```

Check the output for any signs of activity such as:

- Downloads or substitutions
- Evaluations
- Store interactions

If logs show it's idle or stalled, move on to the next steps.

Step 2: Check CPU and memory usage

Examine whether the Lix process is still consuming CPU time or increasing its memory usage. This can be done using `top` (press `Shift+P` to sort by CPU time, `Shift+M` to sort by memory) or your favourite graphical system monitor.

If the client process has CPU usage close to 100% or its memory usage grows, the culprit is likely in the Nix code. Unfortunately no good debugging tools are available there, so one of the better ways is trying to narrow down the issue by trying to reproduce it with smaller packages or a system configuration with less services.

If the CPU usage is low and memory usage stays constant, proceed to the next step.

Step 3: `strace` the client

Attach `strace` to the stuck `lix` process to see where it is spending its time:

```
strace -yy -f nix build <your-command>
```

Watch for long periods spent on blocking syscalls like:

- `poll`
- `read`
- `recvmsg`
- `futex`

This means that it's stuck on communicating with the daemon, move on to the next steps.

Step 4: `strace` the subdaemon

Lix launches subdaemons to handle client requests. You can identify the correct subdaemon by inspecting processes:

```
ps ax | grep nix-daemon
```

Look for a line like:

```
356981 ?        Ssl    0:00 nix-daemon 67890
```

Here, `67890` is the PID of the client Lix process. The daemon serving it is `356981`.

Attach `strace` to the subdaemon:

```
strace -f -p 356981 -yy
```

This lets you inspect whether the daemon is doing work, waiting on a lock, or itself stalled.

Step 5: Investigate lock contention

If any of the traced processes are stuck waiting for file or directory locks (common with `.lock` files in the Nix store), you can identify who is holding the lock:

```
lsof <path to the lockfile>
```

Alternatively, you can ask all locks held by a process: `lsof -p <PID of the target process>`.

Another handy tool is `lslocks` for this.

Check whether the holder is another `nix-daemon` (or `nix-<something>`) process. If so, `strace` that one as well.

If that second process is:

- Doing useful work: this may be expected behavior.
- Also stuck on a lock or polling something indefinitely: you may be looking at a Lix bug.

Try to reduce the chain of waiting processes to the minimal reproducible set and report the issue with detailed diagnostics.

Other ideas

- Try `--pasta-path ""` to your invocation.

Revision #5
Created 2025-05-31 22:52:51 UTC by Raito Bezarius
Updated 2026-02-05 00:11:47 UTC by Raito Bezarius