

Note: GDB with version ≥ 7.0 is needed. (The VM has GDB with version 8.1, so it should be fine)

Here is an example of using gdb reverse execution commands.

As before, in lab 8 directory, start the gdb and load the executable file by typing:

```
gdb fact
```

Before running the program, set a breakpoint so it won't run to the end. Let's use line 22 of factorial.c as an example:

```
(gdb) break factorial.c:22
```

After that, you can run the program:

```
(gdb) run
```

If thing goes well, the program stops at line 22 of factorial.c. To enable reverse execution commands, type:

```
(gdb) record
```

In this way the dynamic execution history afterwards is recorded. Now you can try using the reverse execution commands to instruct the program to go backward. For example, type:

```
(gdb) step
```

This will instruct the program to execute one instruction and go into the function `factorial`. To undo this (go backward), type:

```
(gdb) reverse-step
```

The program goes back to line 22 of factorial.c.

There are some other reverse execution commands, act as opposites of the forward execution commands. For example:

```
(gdb) reverse-next  
(gdb) reverse-finish  
(gdb) reverse-continue
```

Feel free to try them!

Alternatively, you can type:

```
(gdb) set exec-direction reverse
```

After that, all the forward execution commands (`step`, `next`, `finish`, `continue`...) instructs the program to go backward instead. To recover, type:

```
(gdb) set exec-direction forward
```

After you think the dynamic execution history is no longer needed, you can type

```
(gdb) record stop
```

This clears all the dynamic execution history and the program cannot go backward anymore (unless a new `record` is started).

Note:

1. Sometimes the message “No more reverse-execution history” shown by gdb is not correct (usually happens when trying to go forward after going backward). I am not sure whether it is a bug of gdb.
2. After `record` is typed, the forward execution becomes much slower. Please avoid executing standard library functions such as `printf` when `record` is typed.