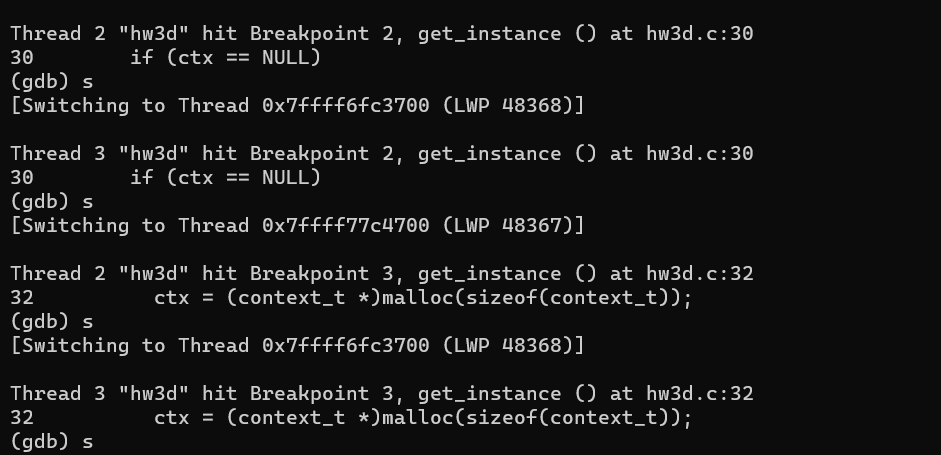
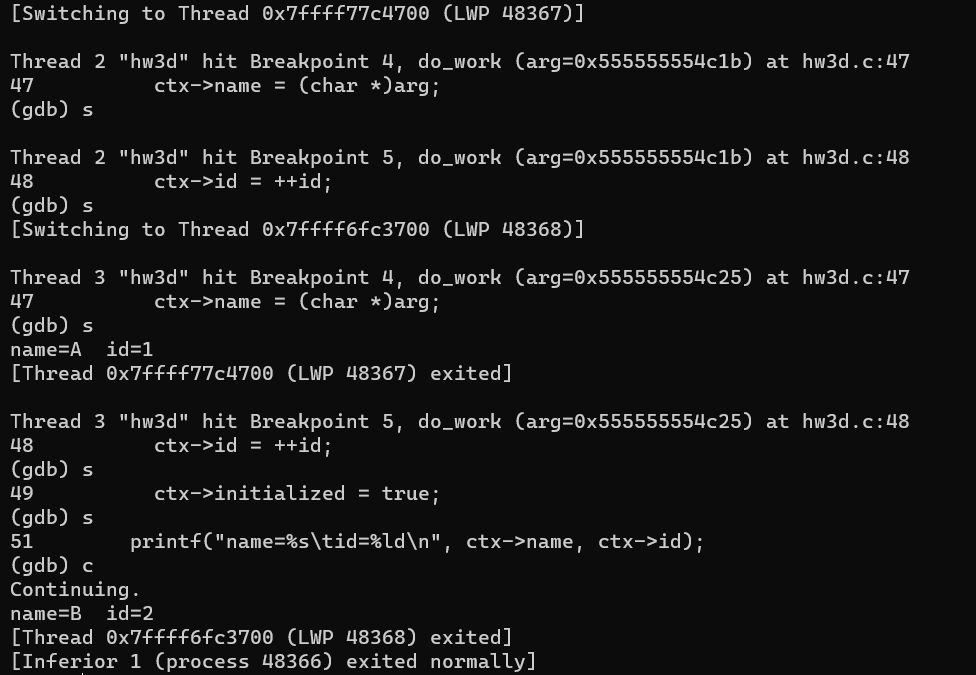
Benedictus Alvian Sofjan

3035603095

COMP3230 Tutorial 3 Exercise 3

Debugging Evaluation

Thread “A” and thread “B” both evaluate if (ctx == NULL) at the same time, and thus two instances of *context\_t* was created. This is a bug.



Thread “A” and thread “B” both access ctx->name and ctx->id at the same time, initialization is done twice, and race condition occurs. Thus, two different contexts are printed by thread “A” and “B”.

I fixed the bug by adding a mutex in get\_instance() so only one thread may evaluate the if (ctx == NULL) expression and thus only a single instance of *context\_t* will be created.

However, this is not enough to entirely eliminate the bug, as there is a very slim chance thread “A” and “B” will evaluate the if (!ctx->initialized) expression at the same time and thus initialize ctx twice. Thus, another mutex is necessary to be added before the if (!ctx->initialized) expression.

