lab8_debug2 report

颜舒逸 523031910677

运行环境：WSL2 Ubuntu-24.04 LTS

调试工具：GDB

bug1

类别

segmentation fault

发现过程

利用GDB进行单步调试，逐行运行到`core.cpp`的`forward`函数时，在`memcpy`函数处崩溃并报错segmentation fault；重新调试，在`memcpy`前执行命令

```
print *(s->x)
```

报错`cannot access memory at...`，判断是指针未分配内存

修复方案

在`RunState`的带参构造函数内添加一行

```
x = new float[dim];
```

bug2

类别

segmentation fault

发现过程

利用GDB进行单步调试，进入到`forward`函数，再进入到`attention`函数后，在第二行进入`RunState`的拷贝构造函数后，在

```
memcpy(key_cache, other.key_cache, key_cache_size);
```

处崩溃并报错segmentation fault; 在执行到该语句前执行命令

```
print malloc_usable_size(key_cache)
print malloc_usable_size(other.key)
print sizeof(float) * config->n_layers * config->seg_len * kv_dim
```

输出依次为

```
$1 = 24
$2 = 24
$3 = 46137344
```

判断是在RunState的带参构造函数内使用uint32_t类型的变量计算得到了uint16_t类型的变量导致大小溢出。

修复方案

将RunState的带参构造函数内前五行的uint16_t类型全部替换为uint64_t。

bug3

类型

segmentation fault

发现过程

执行到Transformer类的函数generate时，检查while循环内的logits，发现该指针没有指向任何地址，判断是在forward函数内返回了局部变量的指针。

修复方案

将forward函数第四行的

```
float logits[p->vocab_size];
```

改为

```
float *logits = new float[p->vocab_size];
```

bug4

类型

use after free

发现过程

修复了以上三个bug后，程序仍然输出无意义内容

```
One day, theater.
```

阅读代码后发现，在`attention`函数内以深拷贝的方式创建了一个新的`RunState`的对象的指针，然而在退出函数前执行了

```
delete s;
```

而在释放`s`的内存空间前，程序仍然修改了`s`的一些成员值，这显然是没有必要的，因此判断`s`事实上在后续的逻辑中仍然需要被使用到。

修复方案

将`attention`函数内的

```
auto s = new RunState(*state);
```

改为

```
auto s = state;
```

并删去

```
delete s;
```

bug5

类型

memory leak

发现过程

`RunState`类和`Transformer`类均有动态分配的内存，但他们的析构函数中却没有释放；`generate`函数的`while`循环内也有动态分配的指针`logits`，同样也没有释放。将cmake选项中的`-DENABLE_SANITIZERS`改为`ON`并重新运行后，程序报错

```
LeakSanitizer: detected memory leaks
```

修复方案

将RunState类的析构函数修改为

```
RunState::~RunState() {
    delete[] x;
    delete[] xb;
    delete[] xb2;
    delete[] hb;
    delete[] hb2;
    delete[] q;
    delete[] key_cache;
    delete[] value_cache;
    delete[] att;
    delete[] logits;
}
```

将Transformer类的析构函数修改为

```
Transformer::~Transformer() {
    gguf_free(gguf_ctx_);
    ggml_free(ggml_ctx_);
    delete config;
    delete weight;
    delete state;
}
```

在generate的while循环的最后一行加上

```
delete[] logits;
```