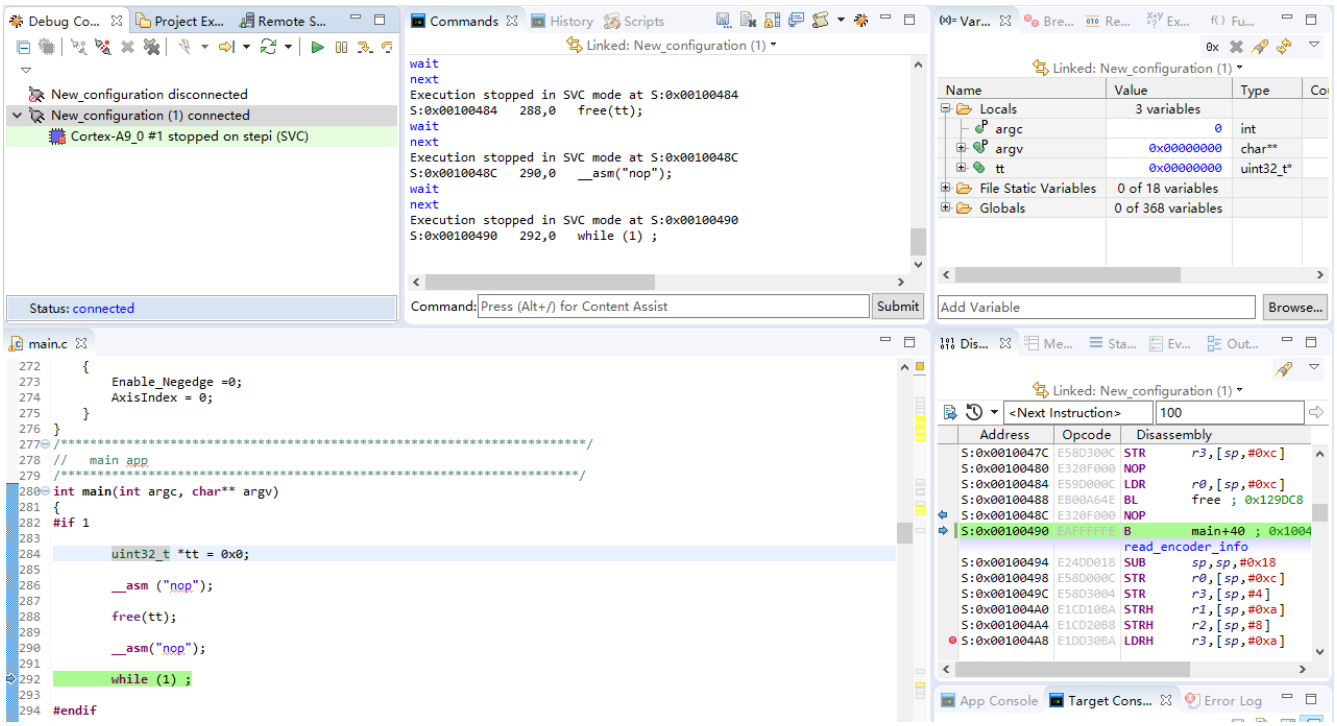
# newlib库移植前后的堆分配和释放函数的对比验证测试

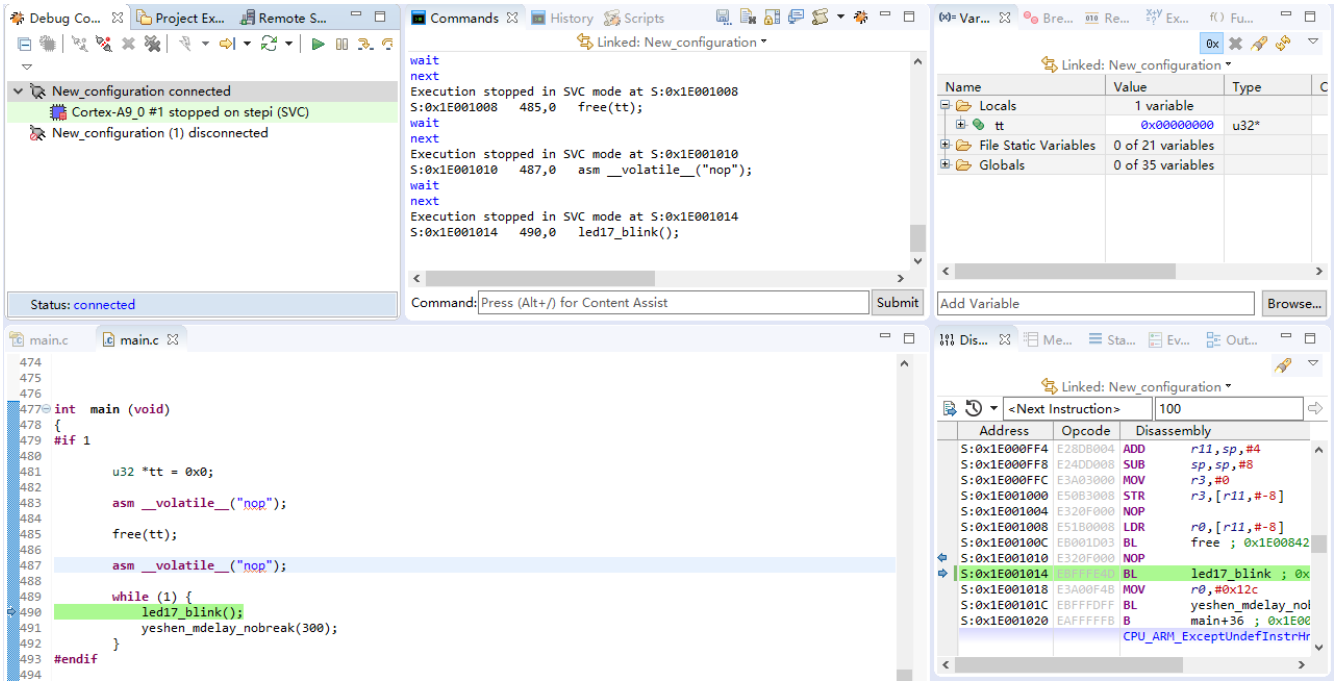
## 一、释放空指针

原版：




上图是未移植newlib版的释放空指针的情况，可以看到释放空指针不会跑飞，然后正在往下运行。

移植newlib版：

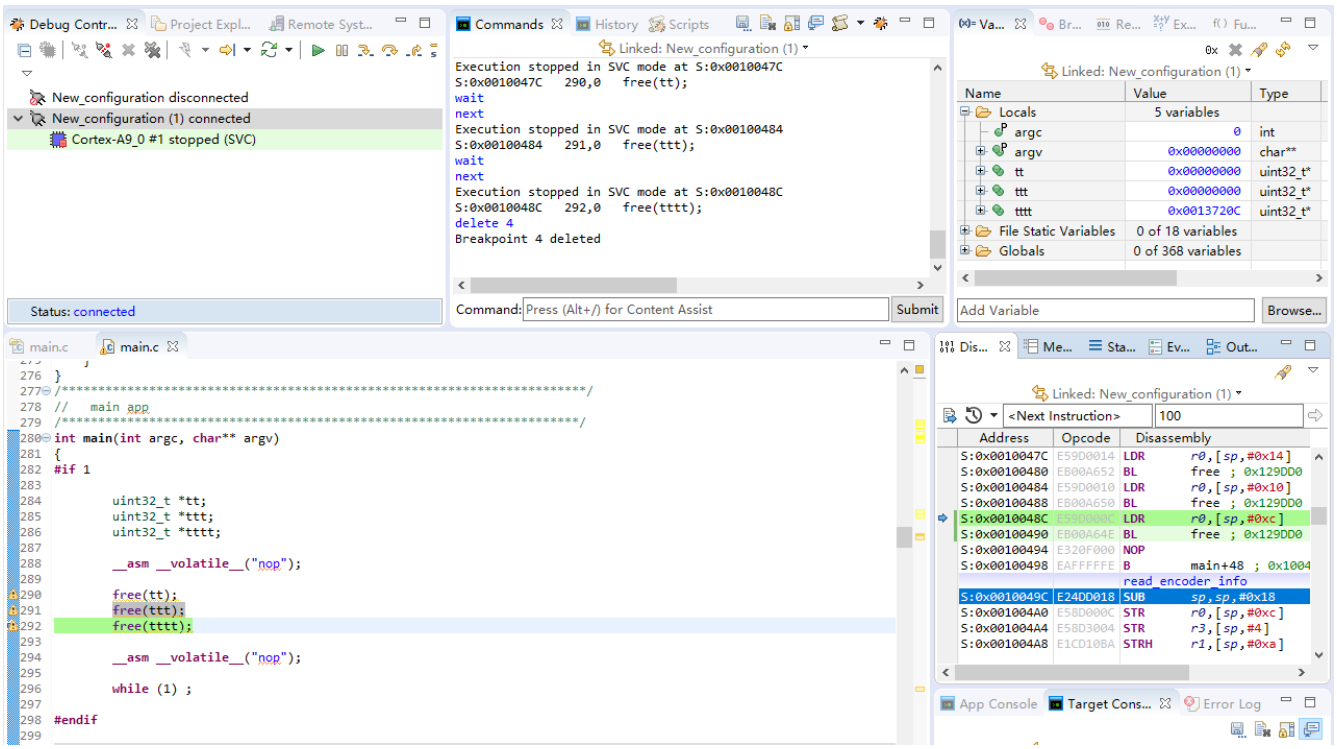


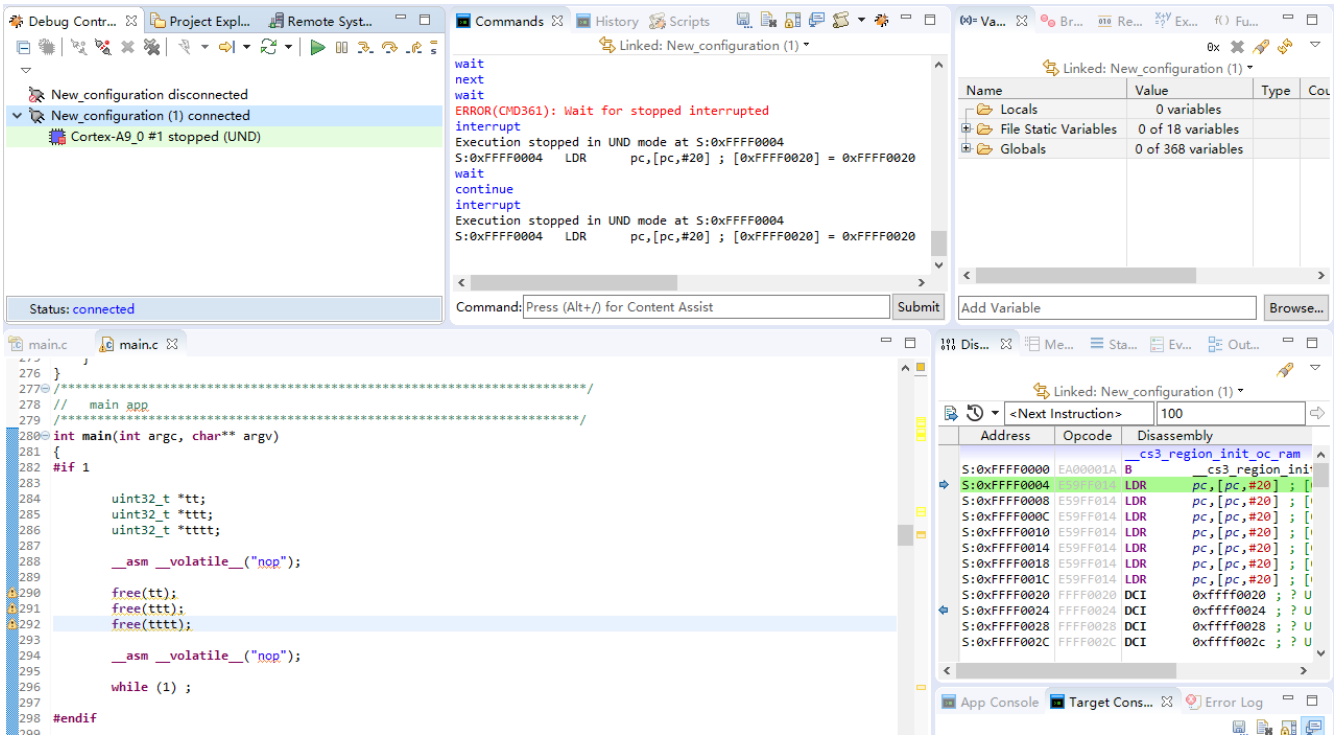

上图是移植newlib版的释放空指针的情况，可以看到释放空指针也不会跑飞，然后正在往下运行。

**结论：**从实验结果和newlib的源代码中分析到释放空指针不会有任何效果，直接返回，也不会发生错误，两个版本释放空指针行为一致。

# 二、释放未初始化指针

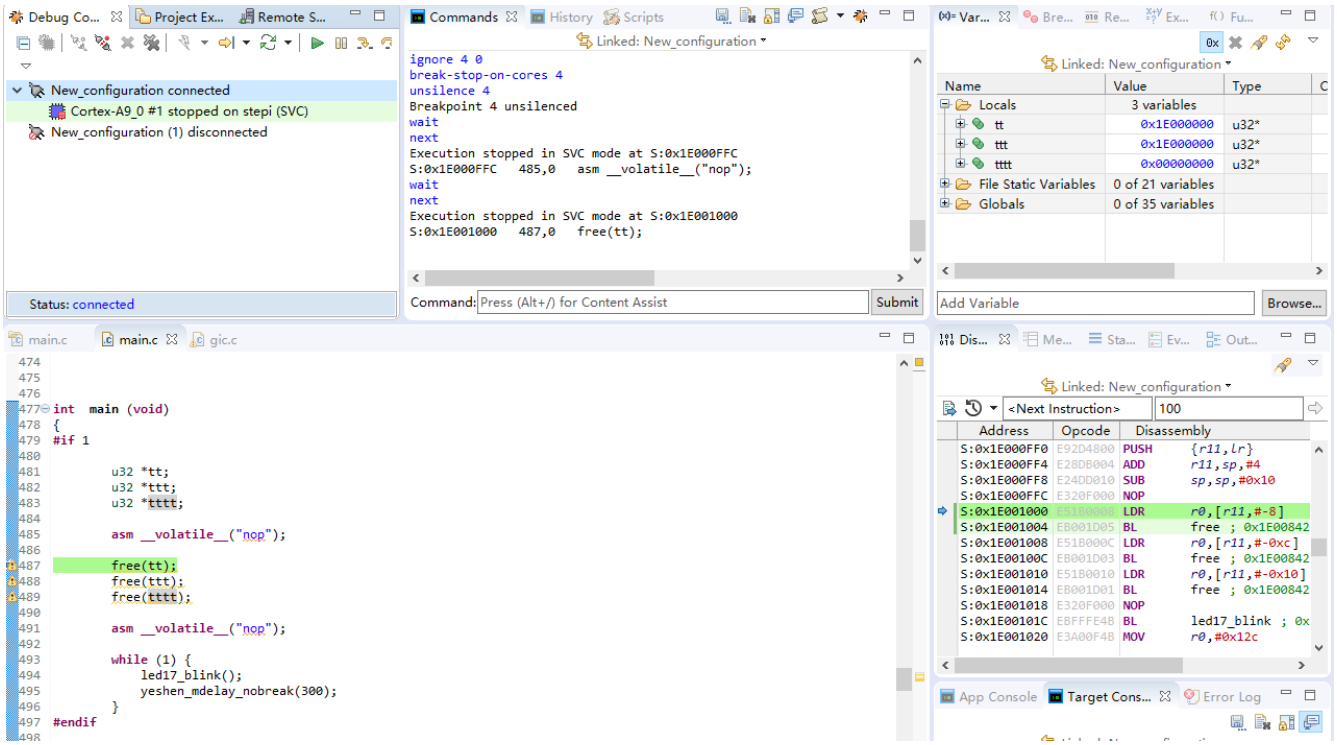
原版：

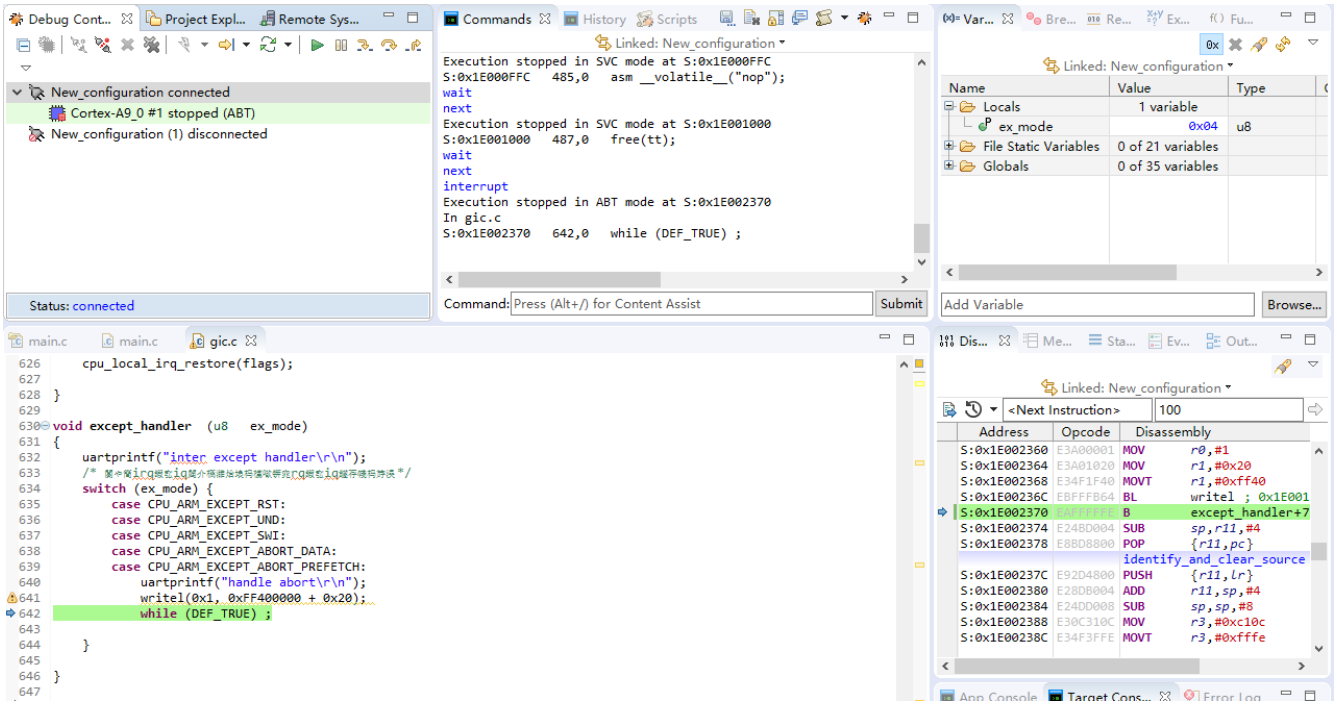





上图释放未初始化的三个指针tt,ttt,tttt。释放前两个没有跑飞，因为tt和ttt刚好是0,就是空指针，而释放tttt的时候跑飞了，因为这不是0。

移植newlib版:



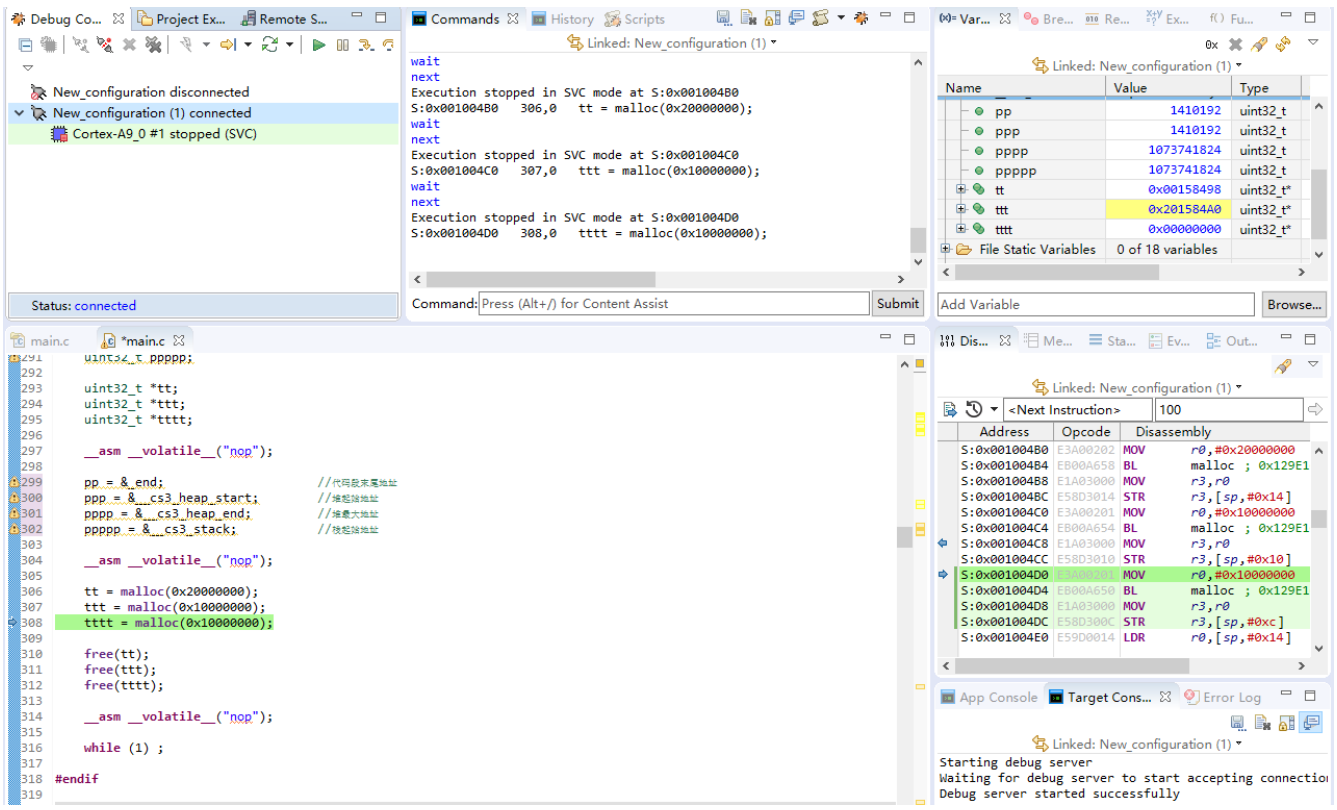


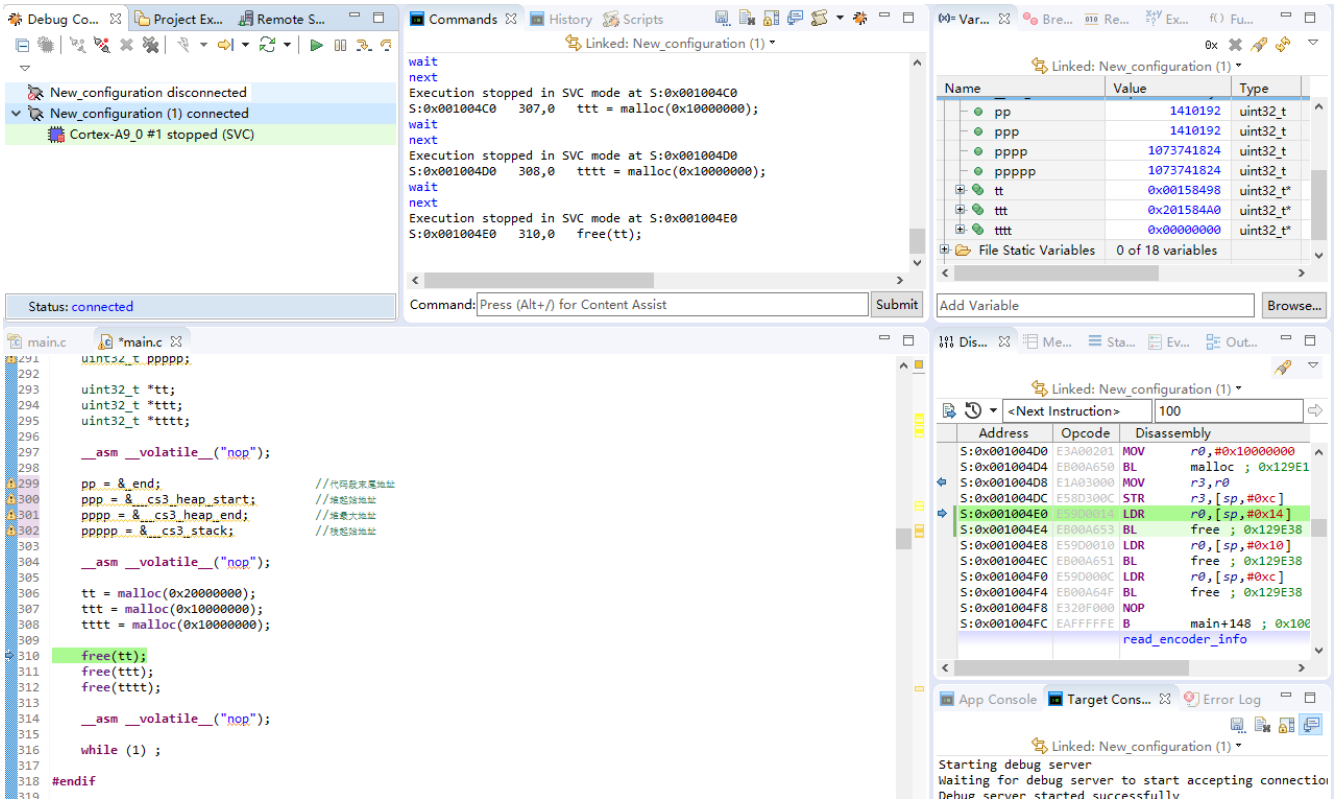

上图也是释放未初始化的三个指针tt,ttt,tttt。第一次释放就跑飞，因为第一个指针就不是0。

**结论：**不能释放未初始化指针，如果未初始化指针的值不巧好为0而直接释放就会跑飞，因为这不是一个合法的申请得到的指针，两个版本释放未初始化指针行为一致。

# 三、申请超限

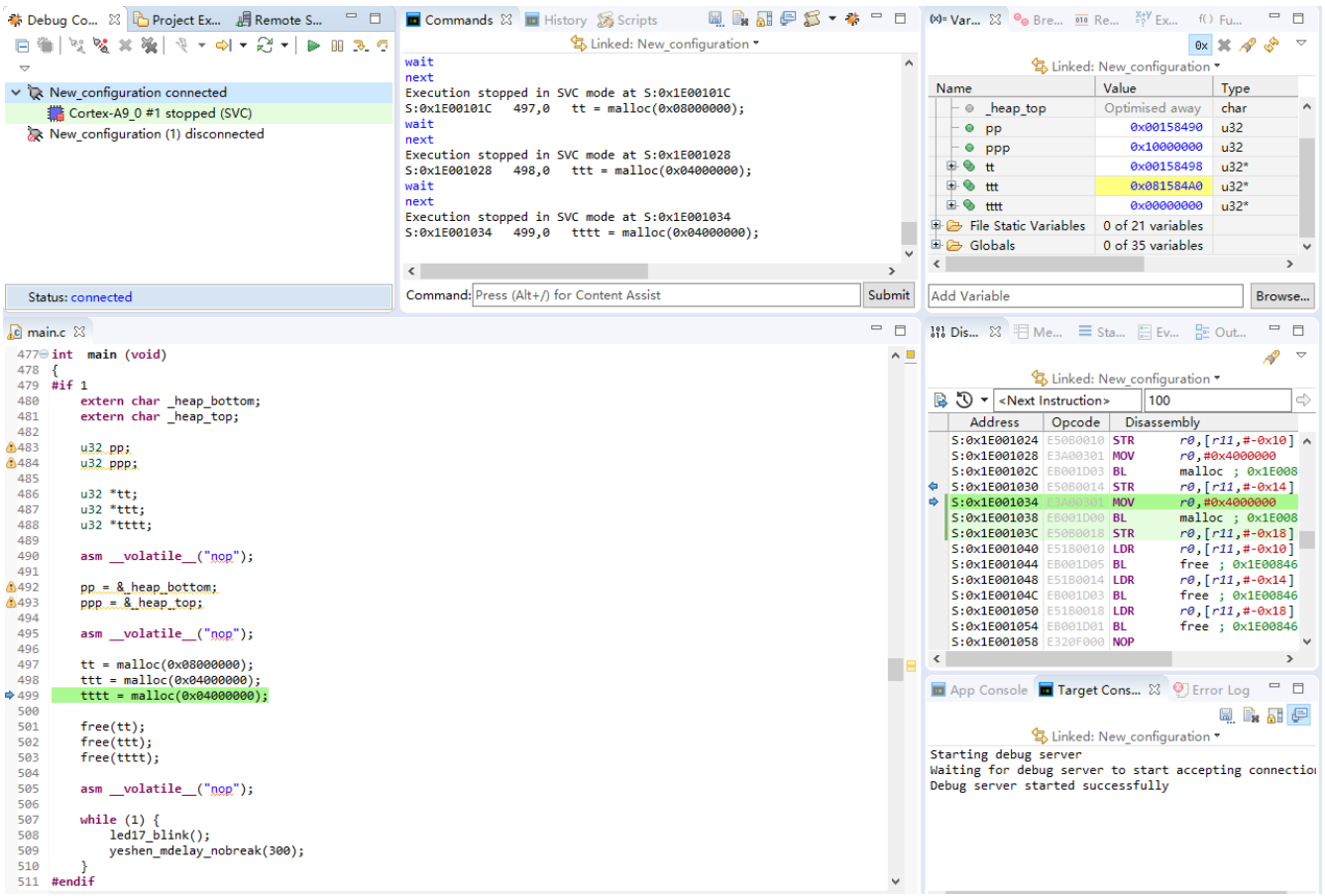
原版：

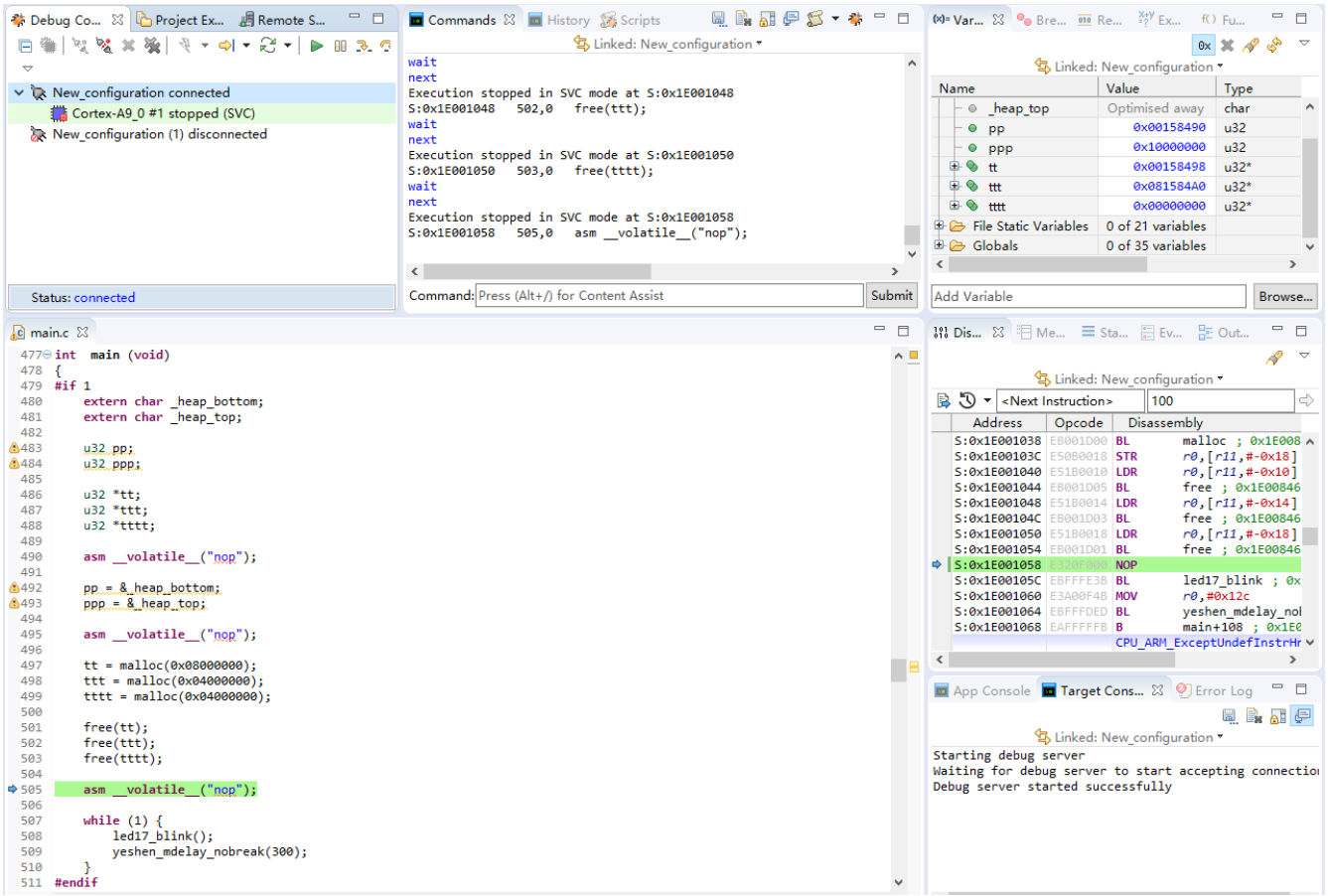





上图连续申请三块内存，申请前两块时没有超过堆的最大界限，到申请第三块时超限，可从图中看到，tttt为0，即申请超限后返回空指针，然后往后正常运行。

移植newlib版:



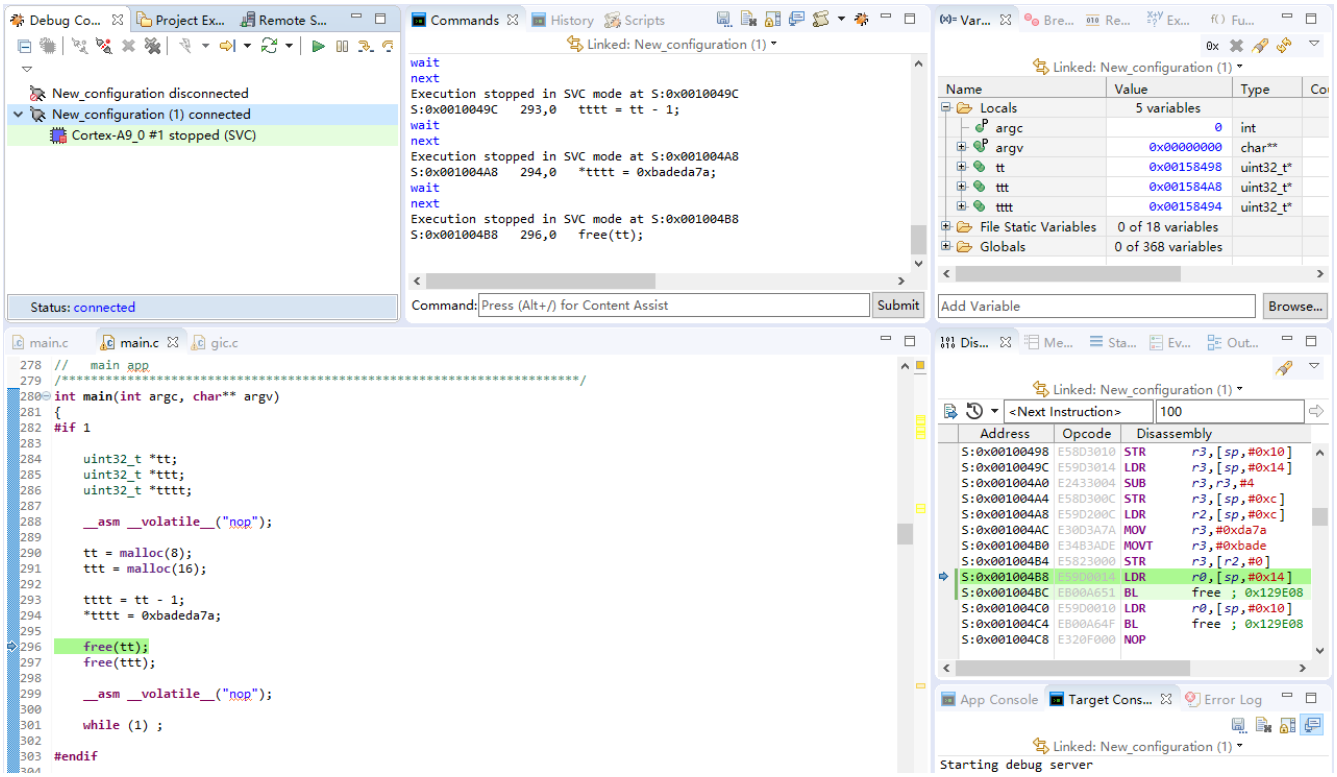


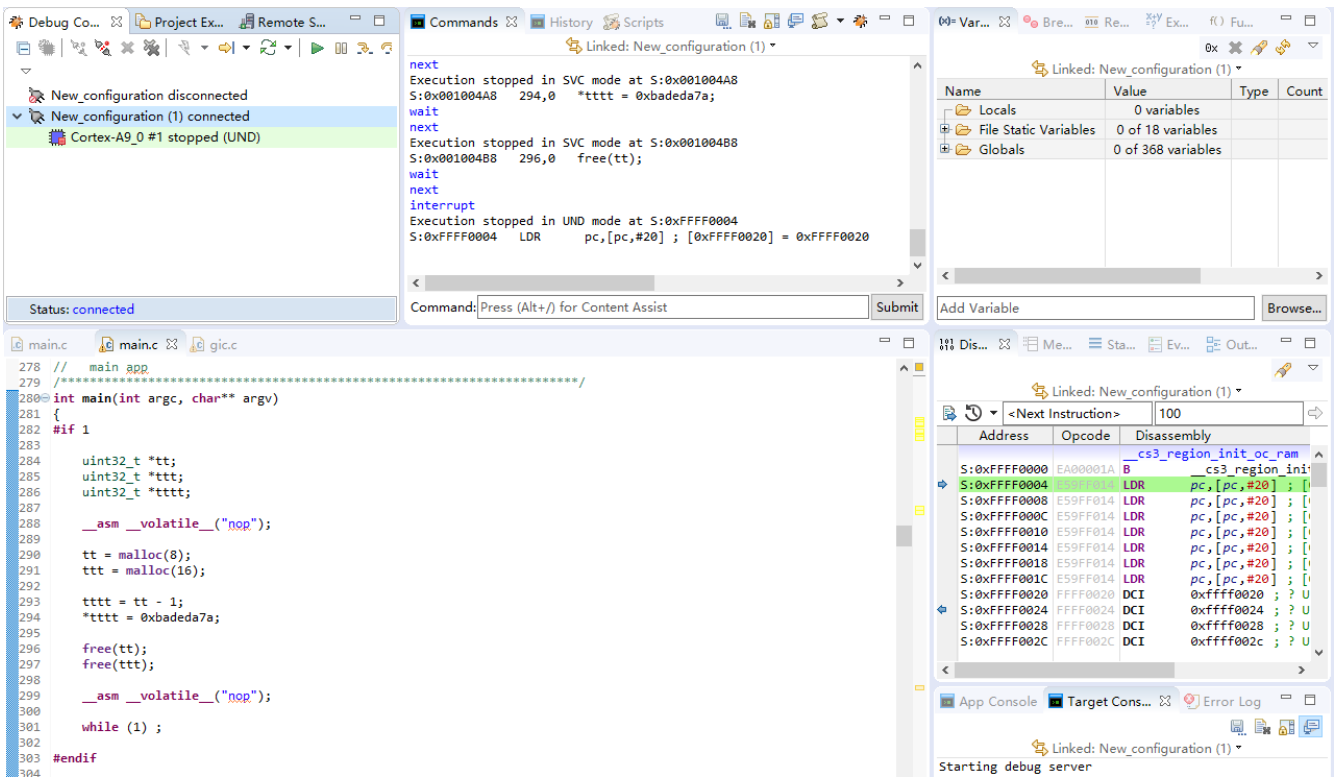

上图的现象和未移植newlib版一致。

**结论：**申请超限后返回空指针，不跑飞。两个版本行为一致。

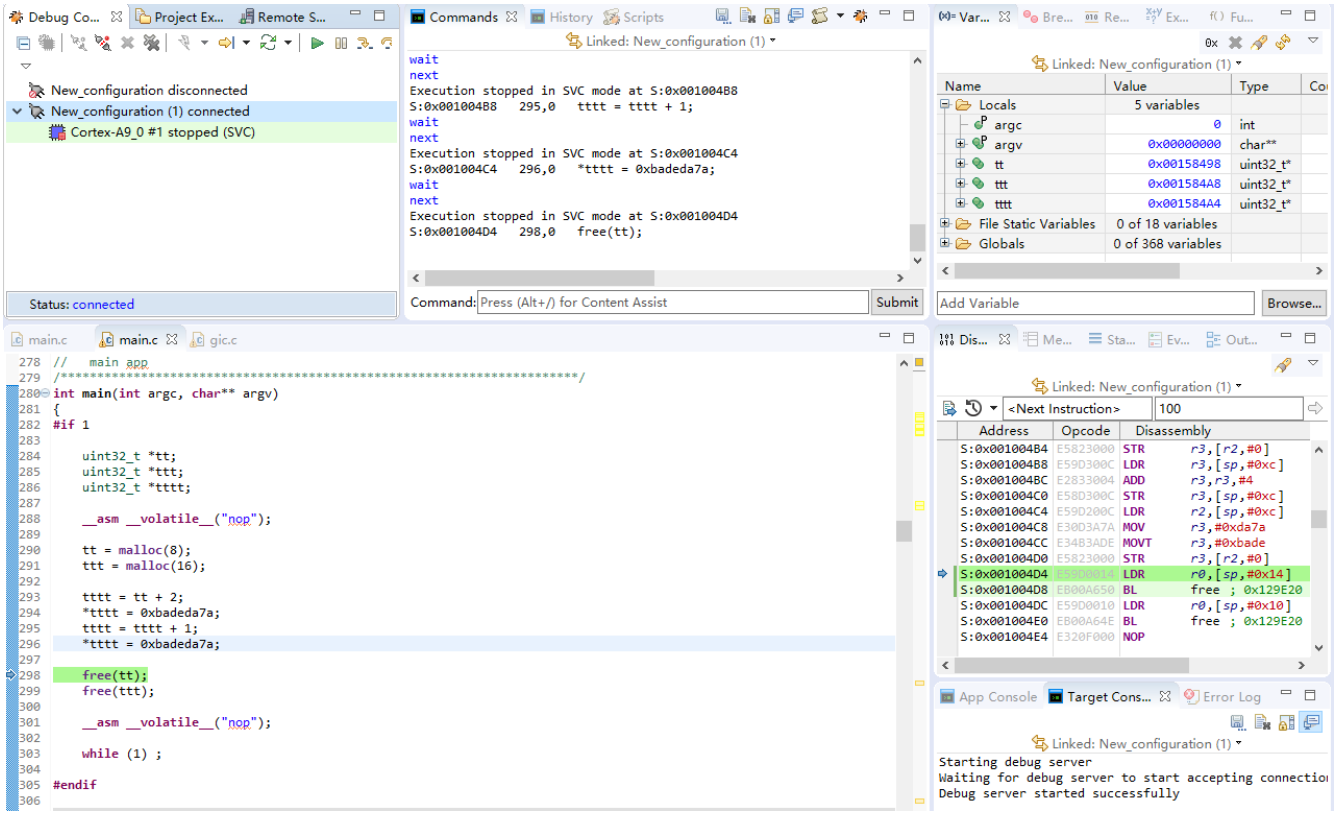
# 四、成功申请内存后，越界写然后释放

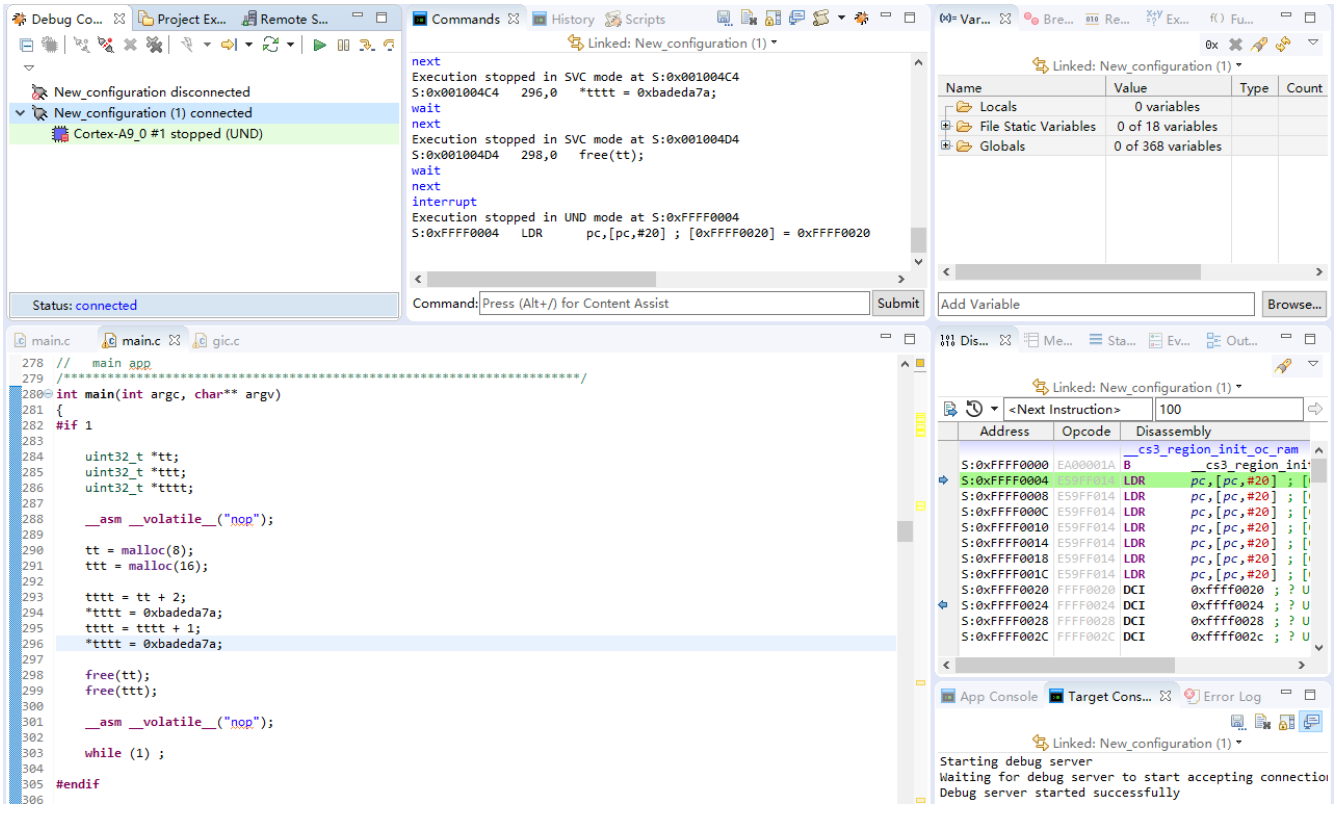
原版：



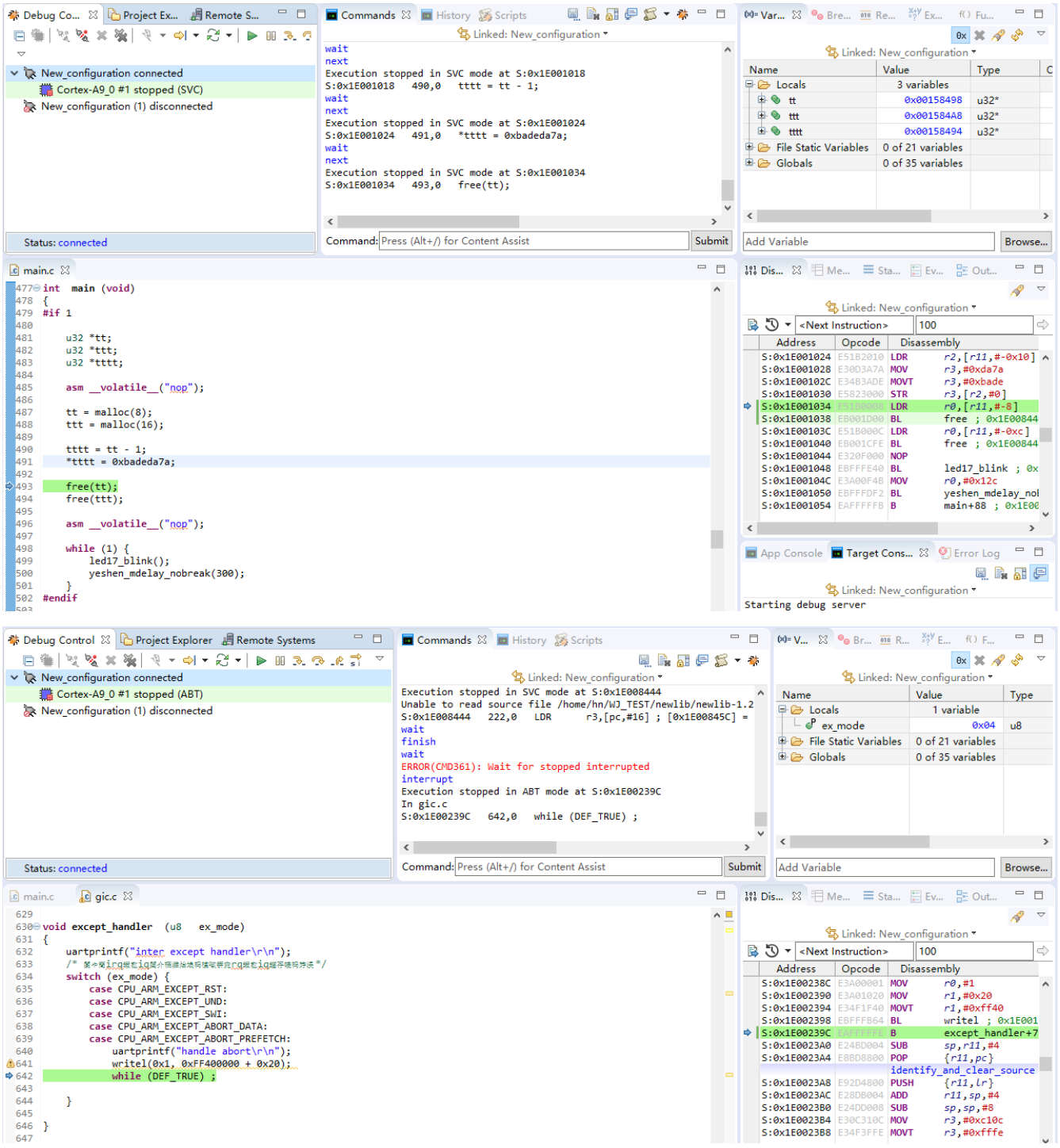

上图展示的是成功申请到一块内存后然后越界向前写一个字，然后跑飞。



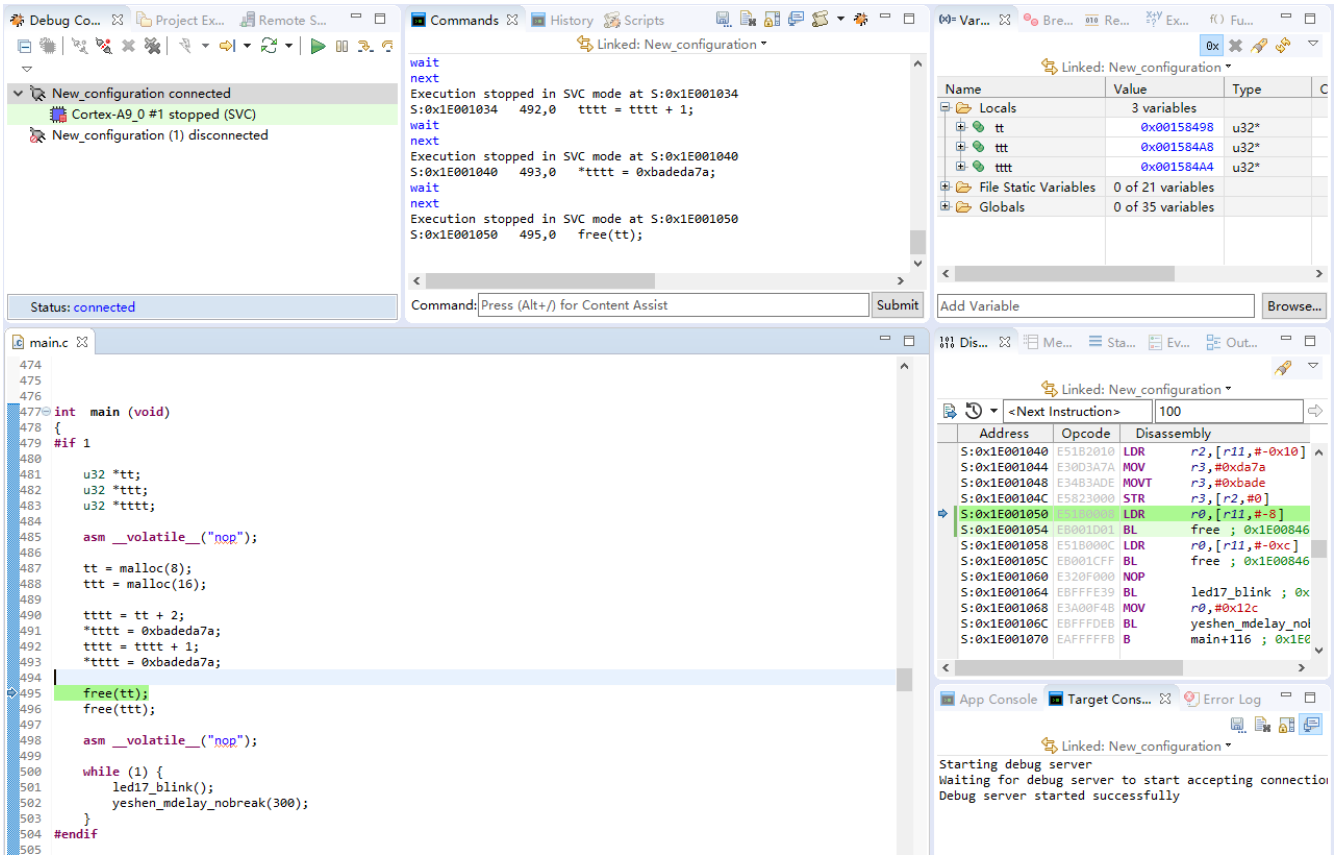



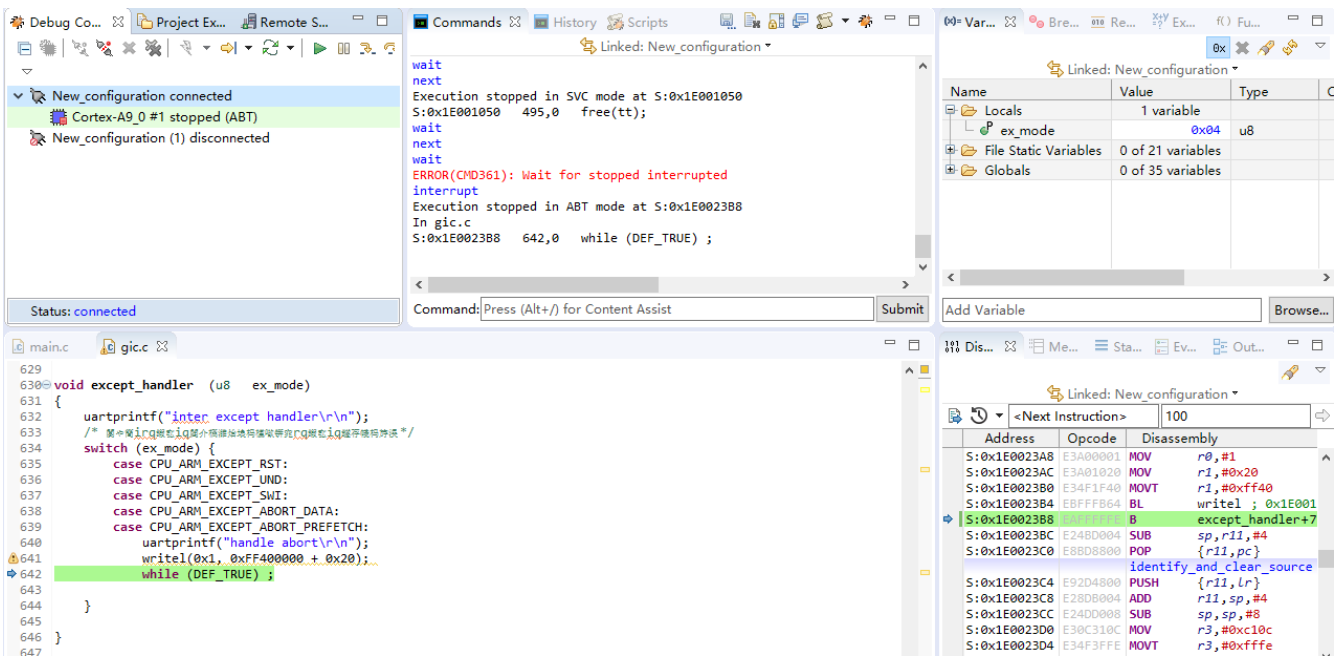

上图展示的是成功申请一块内存然后越界往后写内存，最后跑飞。

移植newlib版：




上图展示的是成功申请一块内存然后越界向前写内存，最后跑飞。




上图展示的是成功申请一块内存然后越界往后写内存，最后跑飞。

**结论：**成功申请内存后不能越界写，不然会跑飞。两个版本行为一致。