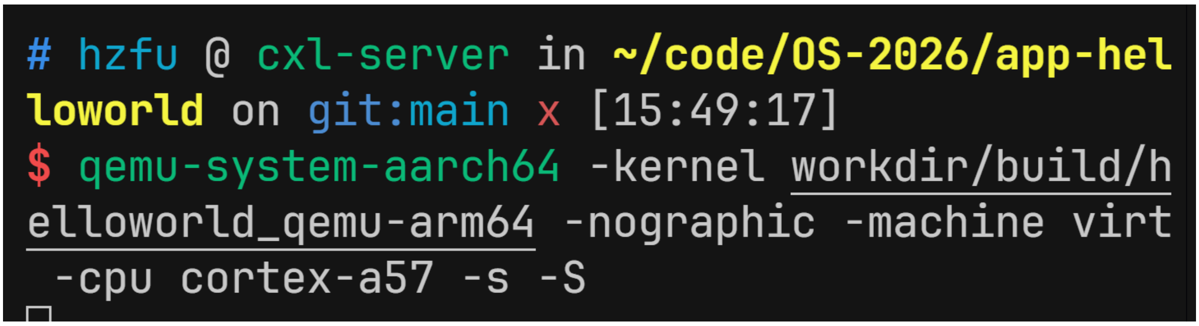
将错误示例代码复制到

OS-2026/app-helloworld/workdir/unikraft/lib/ukallocbbuddy/bbuddy.c

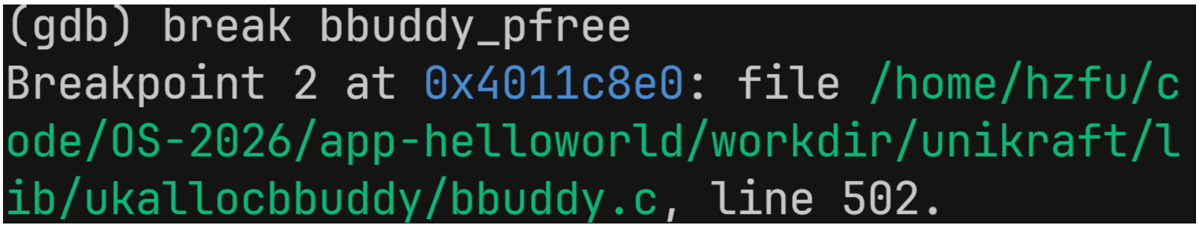
重新编译并运行



```
# hzfu @ cxl-server in ~/code/OS-2026/app-helloworld on git:main x [15:49:17]
$ qemu-system-aarch64 -kernel workdir/build/helloworld_qemu-arm64 -nographic -machine virt -cpu cortex-a57 -s -S
```

用gdb调试

1. 在bbuddy_pfree函数上打断点，进入其内部

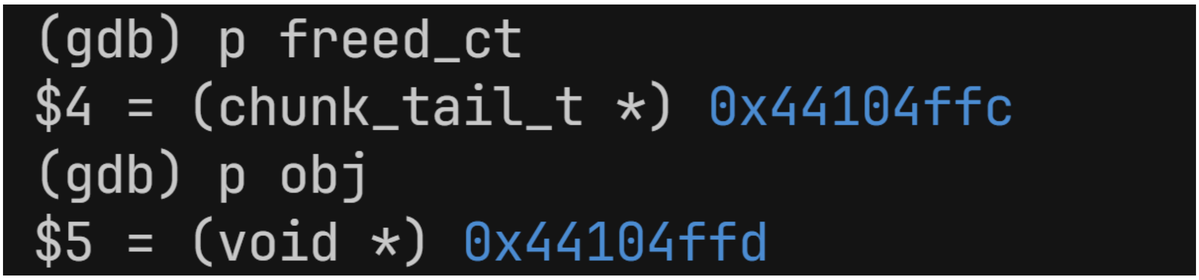


```
(gdb) break bbuddy_pfree
Breakpoint 2 at 0x4011c8e0: file /home/hzfu/code/OS-2026/app-helloworld/workdir/unikraft/lib/ukallocbbuddy/bbuddy.c, line 502.
```

2. 打印出obj和freed_ct 的初始值和循环后的值

发现错误1：obj = (char *) freed_ct + 1;使得页面计算错误

更新 obj 指针的正确方式应该是：在当前 obj 的地址基础上，增加一个页面的大小 (1UL << __PAGE_SHIFT)



```
(gdb) p freed_ct
$4 = (chunk_tail_t *) 0x44104ffc
(gdb) p obj
$5 = (void *) 0x44104ffd
```

3. 错误2：freed_ch->next->pprev = &freed_ch->next;中b->free_head[0]本身可能为NULL，这时候就会导致空指针解引用，引发崩溃。

```
if (b->free_head[0]) // 增加空指针检查
    b->free_head[0]->pprev = &freed_ch->next;
```