Lab4 report

PB20000328 杨博涵

实验目标

实现等大小分区管理和动态分区内存管理

源代码说明

使用空闲链表将不同的EMB块链接起来，用头插法和遍历等手段维护。合并EMB需要检测前后块的关系。

问题回答

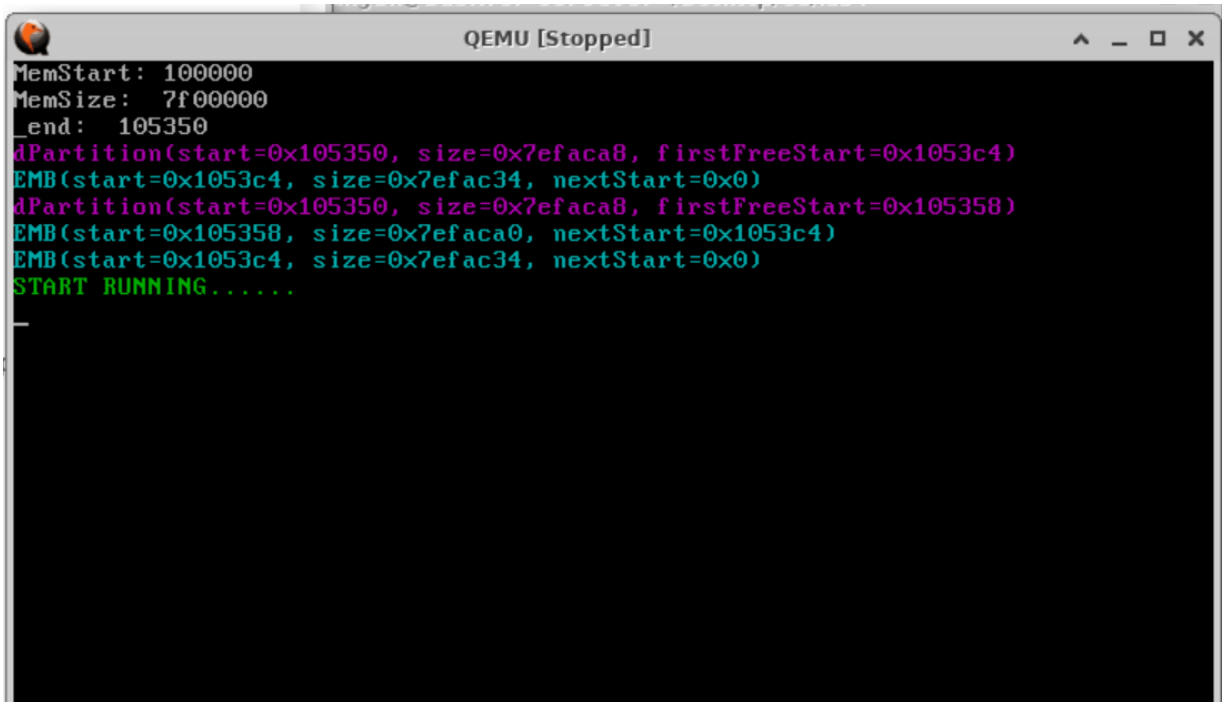
1. 请写出动态分配算法的malloc接口是如何实现的（即malloc函数调用了哪个函数，这个函数又调用了哪个函数...）

malloc调用了dPartitionAlloc函数，这个函数又调用了dPartitionAllocFirstFit函数，我们在dPartitionAllocFirstFit中实现分配算法

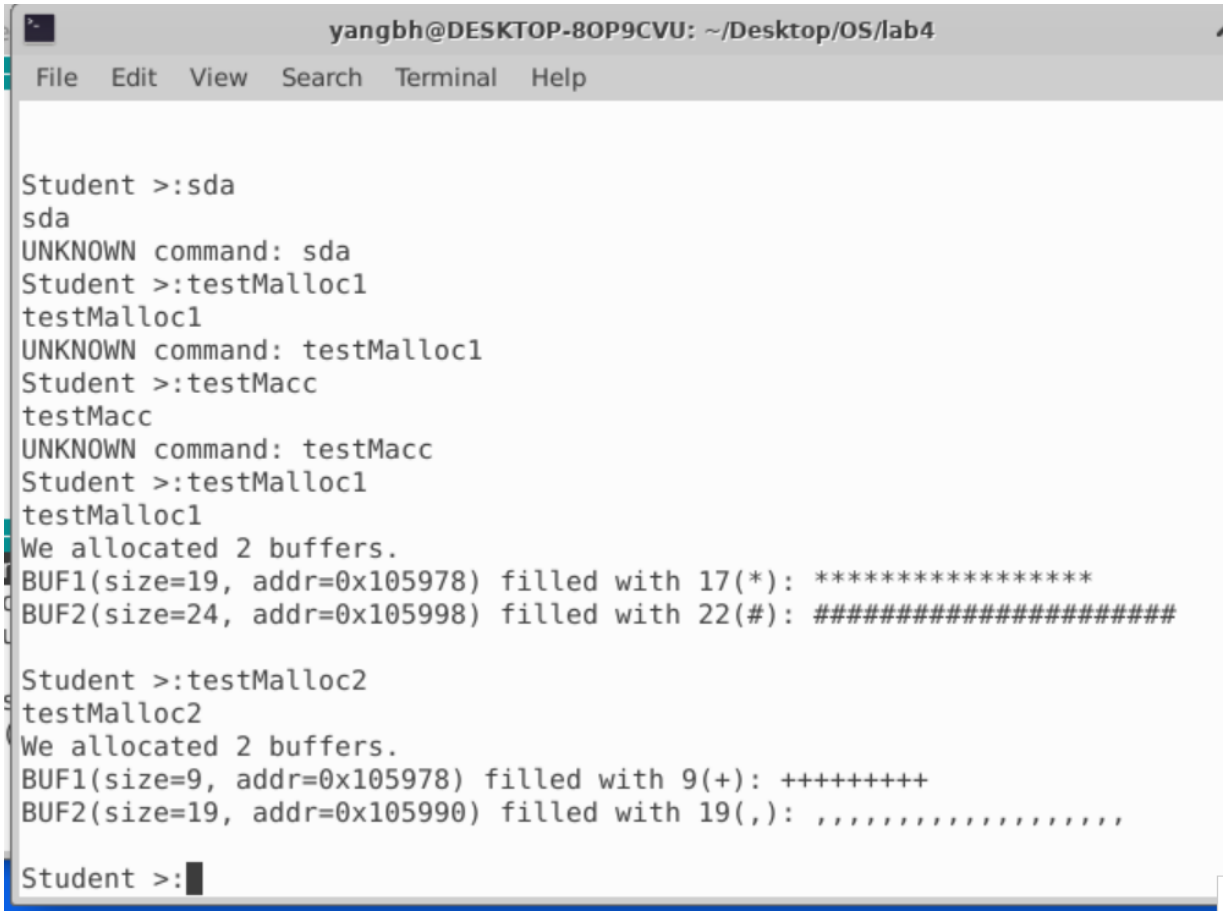
2. 运行 memTestCaseInit 那些新增的shell命令，会出现什么结果，即打印出什么信息（截图放到报告中）？是否符合你的预期，为什么会出现这样的结果。（详细地讲一两个运行结果，大同小异的可以从简）

最终没有把free函数debug完全，所以有一部分test是错误的，malloc则是正确的

init:



testMalloc:



testdP1:

```
We had successfully malloc() a small memBlock (size=0x100, addr=0x105978);
It is initialized as a very small dPartition;
dPartition(start=0x105978, size=0xf8, firstFreeStart=0x105980)
EMB(start=NKANKNOWN command: [AnextStart=0x0)
Student >:[A ==> -:B:C:- ==> -:C- ==> - .
tudent >:Alock A with size 0x10: success(addr=0x105988)!
dPartition(start=0x105978, size=0xf8, firstFreeStart=0x105998)
EMB(start=0x105998, size=0xd8, nextStart=0x0)
Alloc memBlock B with size 0x20: success(addr=0x1059a0)!
dPartition(start=0x105978, size=0xf8, firstFreeStart=0x1059c0)
EMB(start=0x1059c0, size=0xb0, nextStart=0x0)
Alloc memBlock C with size 0x30: success(addr=0x1059c8)!
dPartition(start=0x105978, size=0xf8, firstFreeStart=0x1059f8)
EMB(start=0x1059f8, size=0x78, nextStart=0x0)
Now, release A.
dPartition(start=0x105978, size=0xf8, firstFreeStart=0x105980)
EMB(start=0x105980, size=0xf0, nextStart=0x0)
Now, release B.
dPartition(start=0x105978, size=0xf8, firstFreeStart=0x105980)
EMB(start=0x105980, size=0xf0, nextStart=0x105998)
EMB(start=0x105998, size=0xffefa660, nextStart=0x0)
At last, release C.
dPartition(start=0x105978, size=0xf8, firstFreeStart=0x105980)
EMB(start=0x105980, size=0xf0, nextStart=0x105998)
EMB(start=0x105998, size=0xffefa660, nextStart=0x1059c0)
EMB(start=0x1059c0, size=0xffefa638, nextStart=0x0)
```