

操作系统 lab4 实验报告

231880485 李嘉睿

231880485@smail.nju.edu.cn

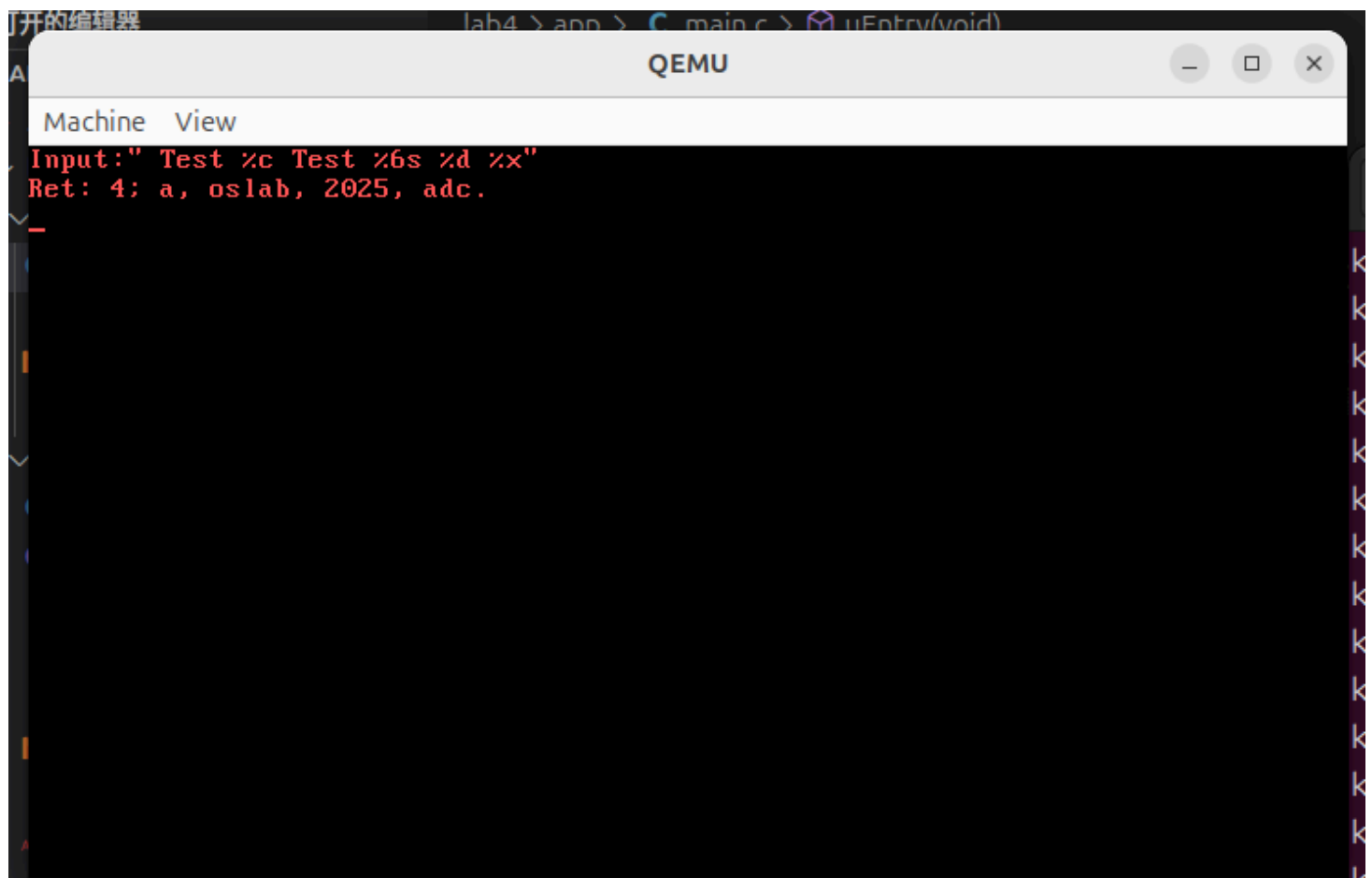
1.实验进度

我完成了所有内容.

包括scanf,信号量,哲学家问题

2.实验结果

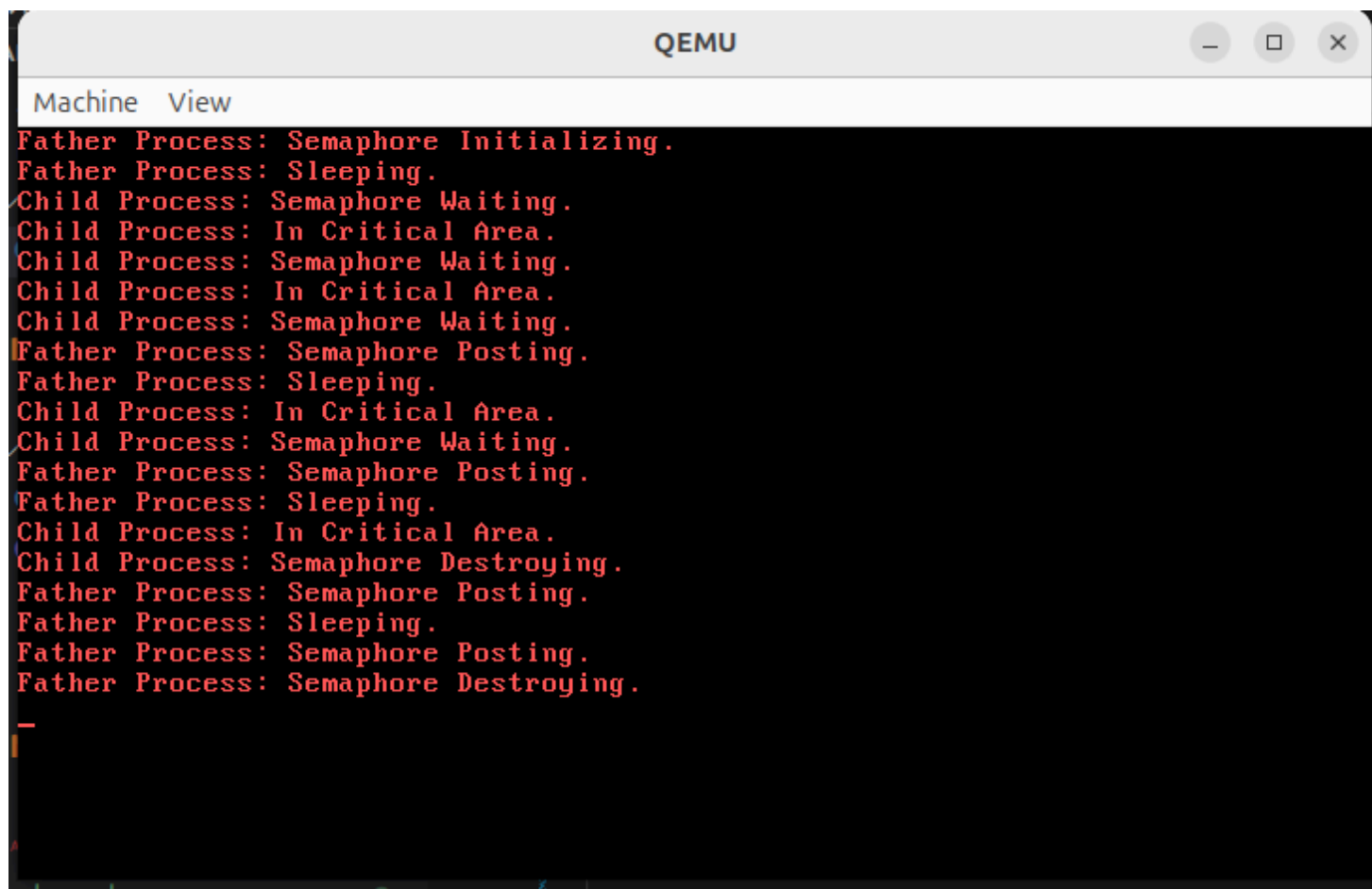
2.1 scanf()函数



The screenshot shows a QEMU terminal window with a title bar that includes the text "QEMU" and standard window control buttons. The terminal content displays the output of a program using scanf(). The input string is "Test %c Test %6s %d %x", and the return value is 4, indicating that four arguments were successfully read. The arguments are 'a', 'oslab', '2025', and 'adc'.

```
Machine View
Input: "Test %c Test %6s %d %x"
Ret: 4: a, oslab, 2025, adc.
```

2.2 信号量系统调用



```
Machine View
Father Process: Semaphore Initializing.
Father Process: Sleeping.
Child Process: Semaphore Waiting.
Child Process: In Critical Area.
Child Process: Semaphore Waiting.
Child Process: In Critical Area.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Sleeping.
Child Process: In Critical Area.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Sleeping.
Child Process: In Critical Area.
Child Process: Semaphore Destroying.
Father Process: Semaphore Posting.
Father Process: Sleeping.
Father Process: Semaphore Posting.
Father Process: Semaphore Destroying.
```

2.3 哲学家进餐问题

策略是方案三的奇偶数,

Machine View

```
Philosopher 0: think
Philosopher 1: think
Philosopher 2: think
Philosopher 3: think
Philosopher 4: think
Philosopher 0: eat
Philosopher 3: eat
Philosopher 0: think
Philosopher 1: eat
Philosopher 3: think
Philosopher 4: eat
Philosopher 1: think
Philosopher 2: eat
Philosopher 4: think
Philosopher 0: eat
Philosopher 2: think
Philosopher 3: eat
Philosopher 0: think
Philosopher 1: eat
```

Machine View

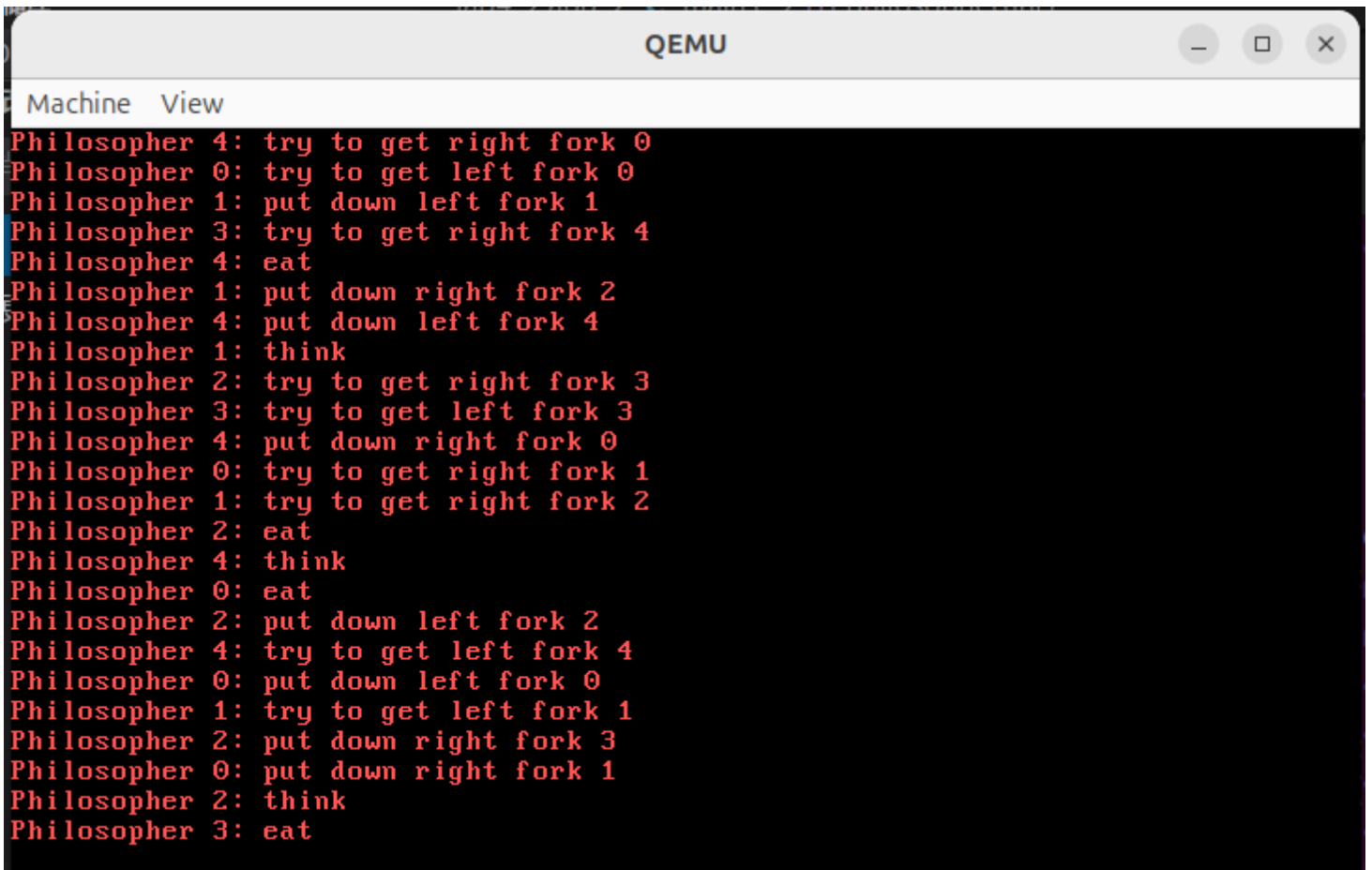
```
Philosopher 2: think
Philosopher 3: eat
Philosopher 0: think
Philosopher 1: eat
Philosopher 3: think
Philosopher 0: eat
Philosopher 1: think
Philosopher 2: eat
Philosopher 4: eat
Philosopher 0: think
Philosopher 1: eat
Philosopher 2: think
Philosopher 3: eat
Philosopher 4: think
Philosopher 0: eat
Philosopher 1: think
Philosopher 2: eat
Philosopher 3: think
Philosopher 0: think
Philosopher 4: eat
Philosopher 1: eat
Philosopher 2: think
Philosopher 3: eat
Philosopher 4: think
```

就这样无限循环

如果加上叉子动态,输出结果如下:



```
Machine View
philosopher 0: think
philosopher 1: think
philosopher 2: think
philosopher 3: think
philosopher 4: think
philosopher 0: try to get left fork 0
philosopher 1: try to get right fork 2
philosopher 2: try to get left fork 2
philosopher 3: try to get right fork 4
philosopher 4: try to get left fork 4
philosopher 0: try to get right fork 1
philosopher 1: try to get left fork 1
philosopher 3: try to get left fork 3
philosopher 0: eat
philosopher 3: eat
philosopher 0: put down left fork 0
philosopher 3: put down left fork 3
philosopher 0: put down right fork 1
philosopher 3: put down right fork 4
philosopher 0: think
philosopher 1: eat
philosopher 3: think
philosopher 4: try to get right fork 0
```



```
Machine View
Philosopher 4: try to get right fork 0
Philosopher 0: try to get left fork 0
Philosopher 1: put down left fork 1
Philosopher 3: try to get right fork 4
Philosopher 4: eat
Philosopher 1: put down right fork 2
Philosopher 4: put down left fork 4
Philosopher 1: think
Philosopher 2: try to get right fork 3
Philosopher 3: try to get left fork 3
Philosopher 4: put down right fork 0
Philosopher 0: try to get right fork 1
Philosopher 1: try to get right fork 2
Philosopher 2: eat
Philosopher 4: think
Philosopher 0: eat
Philosopher 2: put down left fork 2
Philosopher 4: try to get left fork 4
Philosopher 0: put down left fork 0
Philosopher 1: try to get left fork 1
Philosopher 2: put down right fork 3
Philosopher 0: put down right fork 1
Philosopher 2: think
Philosopher 3: eat
```

3.修改的代码

- **app/main.c** :增加了哲学家问题
- **makefile**: 命令**objcopy**添加 -S -j .text
- **memory.h**: 增加了pcb数量, NR_SEGMENTS 改成16,把 MAX_DEV_NUM 和 MAX_SEM_NUM 改成6,适配5个哲学家
- **irqhandle.c**:
 - 完
成 keyboardhandle , syscallSemInit,syscallSemWait,syscallSemPost,syscallSemDestroy ,
syscallReadStdIn
 - 额外实现了 getpid 函数,不过没用上
- 添加了一个log()函数用于输出调试信息

4.思考和心得

4.1 app/main.c的函数

一个疑问,在main.c中声明实现的函数,为何即使没有在uMain()中调用,也会运行?

4.2 修改memory.h

一开始不太敢动几个常量,还以为是问题的限制条件,跟同学讨论了之后才决定应该大胆改掉