**操作系统大作业1**

**提交截止日：11月4日之前**

**总体要求**

在github上创建os-assignment1项目，项目包括三个题目的结果:

1. 哲学家就餐（dinning philosophers）问题：dph.c
2. 生产者消费者(producer-consumer)问题：prod.c和cons.c
3. Linux内核实验：（c）complete fair scheduler（CFS）内核代码阅读报告（b）添加系统调用：用户层测试程序mycall.c

该项目目录下，手工编写Makefile，通过命令行下的make all命令，可以同时产生上述文件对应的可执行文件：dph，prod, cons, mycall；通过单独的make 目标，可以产生单独的bin文件。比如make dph可以单独生成dph。

该目录下，同时存放1个word文件，按上述顺序包含实验内容报告，每个题目安排一大节。

**题目占分**：题目1：20分，题目2：30分，题目3(a)：30分，题目3（b）：20分. 对于编程题，结果正确和代码质量（逻辑流程清晰，适当添加注释）分别占70%和30%的分数。

1. **哲学家就餐问题**

参考课本（第十版）第7章project 3的要求和提示

1. 使用POSIX实现

5. 要求通过make，能输出dph文件，输出哲学家们的状态。打印结果，截屏放到作业报告中。

1. **生产者消费者问题**
2. 需要创建生产者和消费者两个进程（注意：不是线程），一个prod，一个cons，每个进程有3个线程。它们之间的缓冲最多容纳20个数据。
3. 每个生产者随机产生一个数据，打印出来自己的id（进程、线程）以及该数据；每个消费者取出一个数据，然后打印自己的id和数据。
4. 生产者和消费者这两个进程之间通过共享内存来通信，通过信号量来同步。
5. 生产者生成数据的间隔和消费者消费数据的间隔，按照负指数分布来控制，各有一个控制参数,
6. 运行的时候，开两个窗口，一个./prod ，另一个./cons ，要求测试不同的参数组合，打印结果，截屏放到作业报告里。
7. **Linux内核实验（Linux 4.0或以上）**
8. 以CFS的主要文件Fair.c为起点，浏览相关联的文件。理解Linux进程的基本结构、状态设置，CPU的调度基本架构，理解CFS调度算法的基本流程和主要数据结构。摘取关键代码片段，用自己的的方式描述出来。不要求理解每一条语句，但需要陈述主要脉络。此外，单独回答以下问题：
9. 简述进程优先级、nice值和权重之间的关系
10. CFS调度器中的vruntime的基本思想是什么？是如何计算的？何时得到更新？其中的min\_vruntime有什么作用？
11. 添加一个内核系统调用，重新编译内核，启动后运行screenfetch命令（可能需要安装），截屏显示结果，需要显示出运行主机的内核版本、CPU等信息（注意：每个同学在自己的机器上编译，这些信息会有所差异，以此作为同学们的作业区分。）

编写用户层程序mycall.c调用该调用，要求打印出当前进程的调度信息（如下图所示），通过dmesg可以查看。实现时，可以通过current访问sched\_entity的数据成员。

