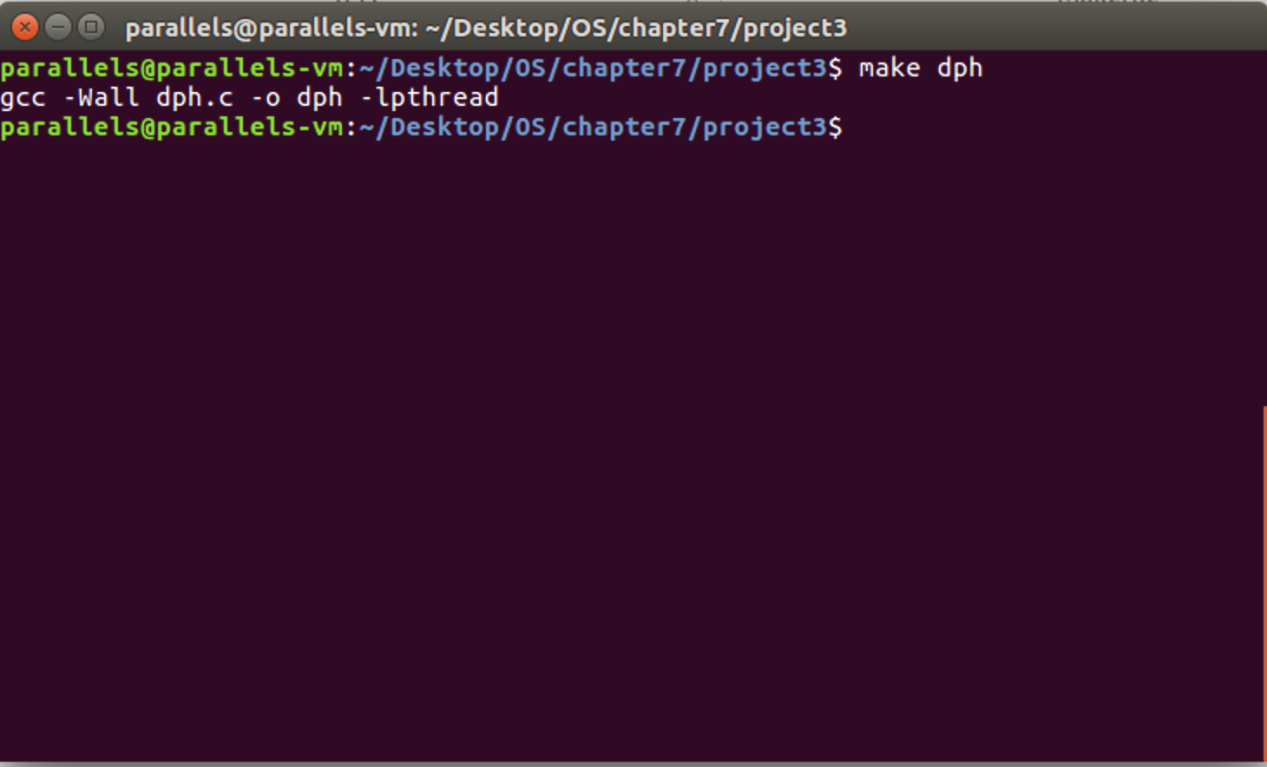
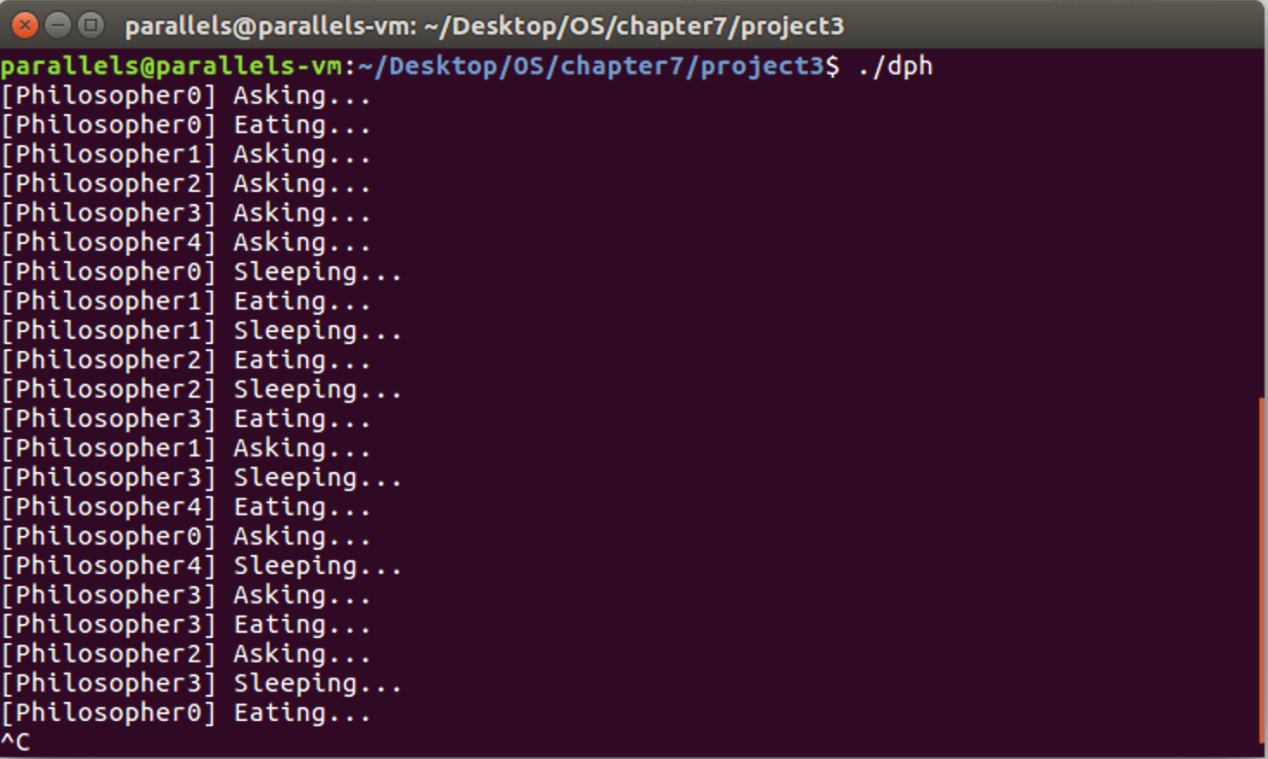
1.

Make dph

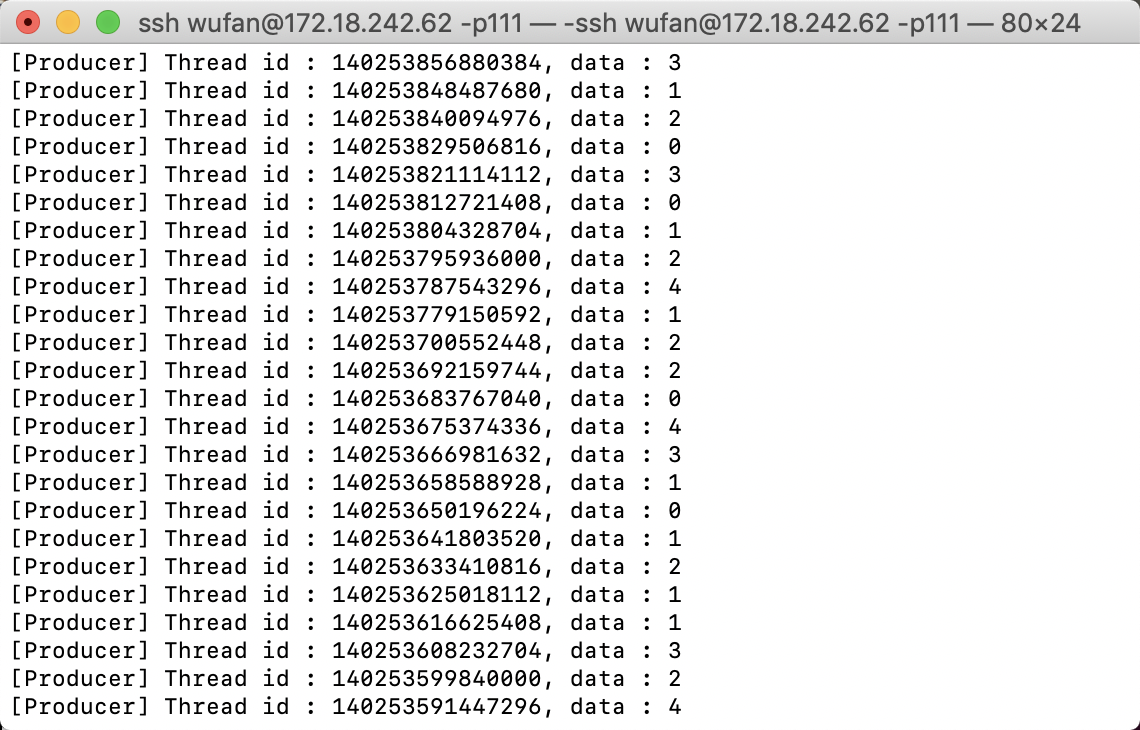


./dph

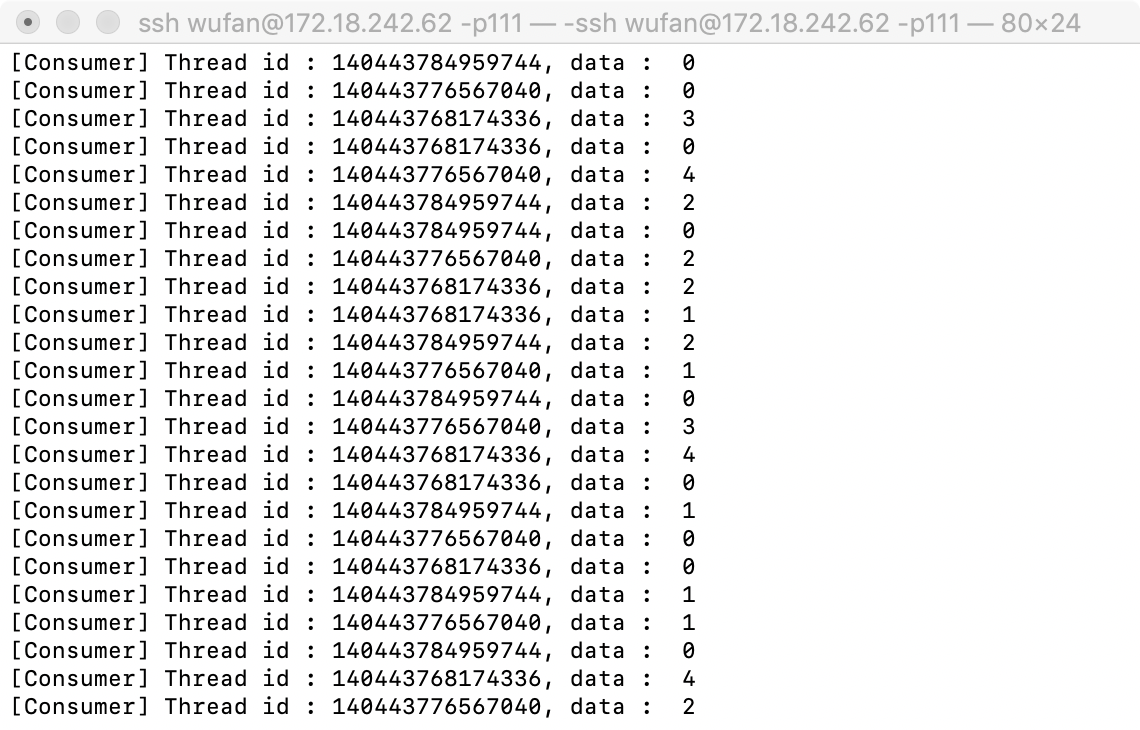


2.

生产者进程



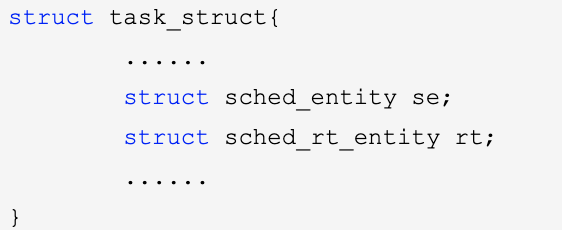
消费者进程



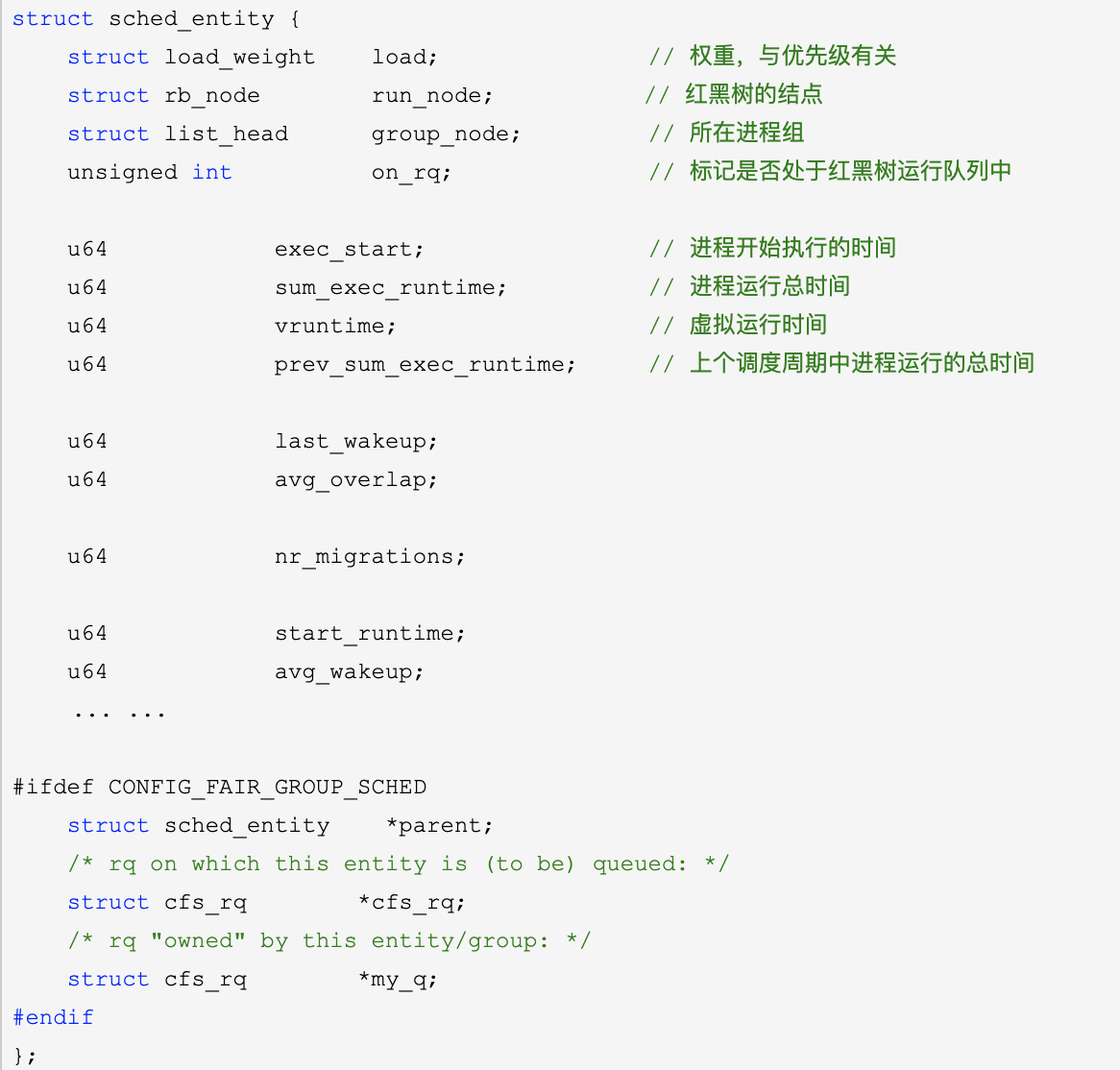
3.



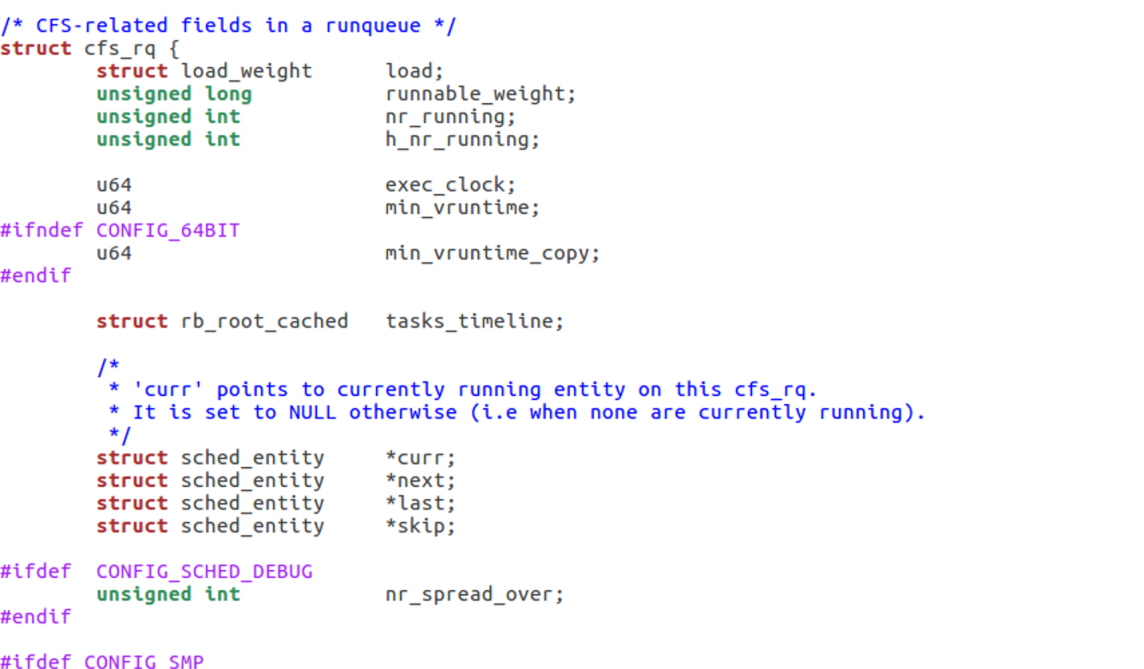
进程结构task\_struct



CFS调度器实体sched\_entity



CFS调度队列cfs\_rq



1. CFS调度流程：

调度器在sched\_entity中使用vruntime虚拟运行时间记录每个进程的运行时

调用 update\_curr() 函数，计算当前进程的执行时间，并将结果存放在delta\_exec

中，将 delta\_exec 作为参数传递给 \_update\_curr() 函数，该函数将根据当前可运

行的进程总数对运行时间进行加权运算；将权值与当前运行进程的 vruntime 相

加得到新的 vruntime 值。CFS挑选下一个进程运行时选择具有最小vruntime的。

进程队列使用红黑树数据结构。

2）进程优先级、nice值和权重之间的关系

Task\_struct 中的int prio, static\_prio, normal\_prio, unsigned int rt\_priority记录进程优先级，prio指的是任务当前的动态优先级，其值影响任务的调度顺序。Static\_prio静态优先级，进程被创建时产生的优先级，normal\_prio常规优先级，基于静态优先级和调度策略产生。Rt\_priority实时优先级，由进程的实时性决定。

Nice值即友好值

进程优先级 =（旧）进程优先级 + nice

nice值越小，进程的权重越大

1. vruntime为虚拟运行时间，min\_vruntime为实际运行时间

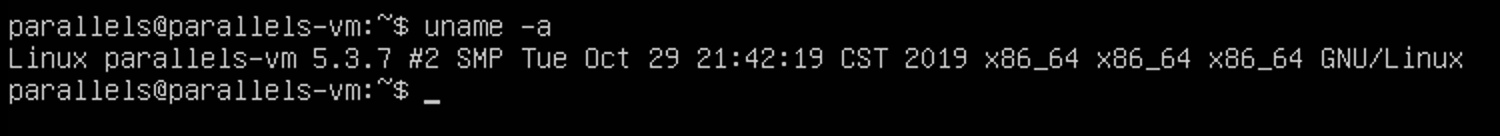
Vruntime = min\_vruntime \* (nice\_0\_weight / weight)

其中nice\_0为nice值为0时的权重

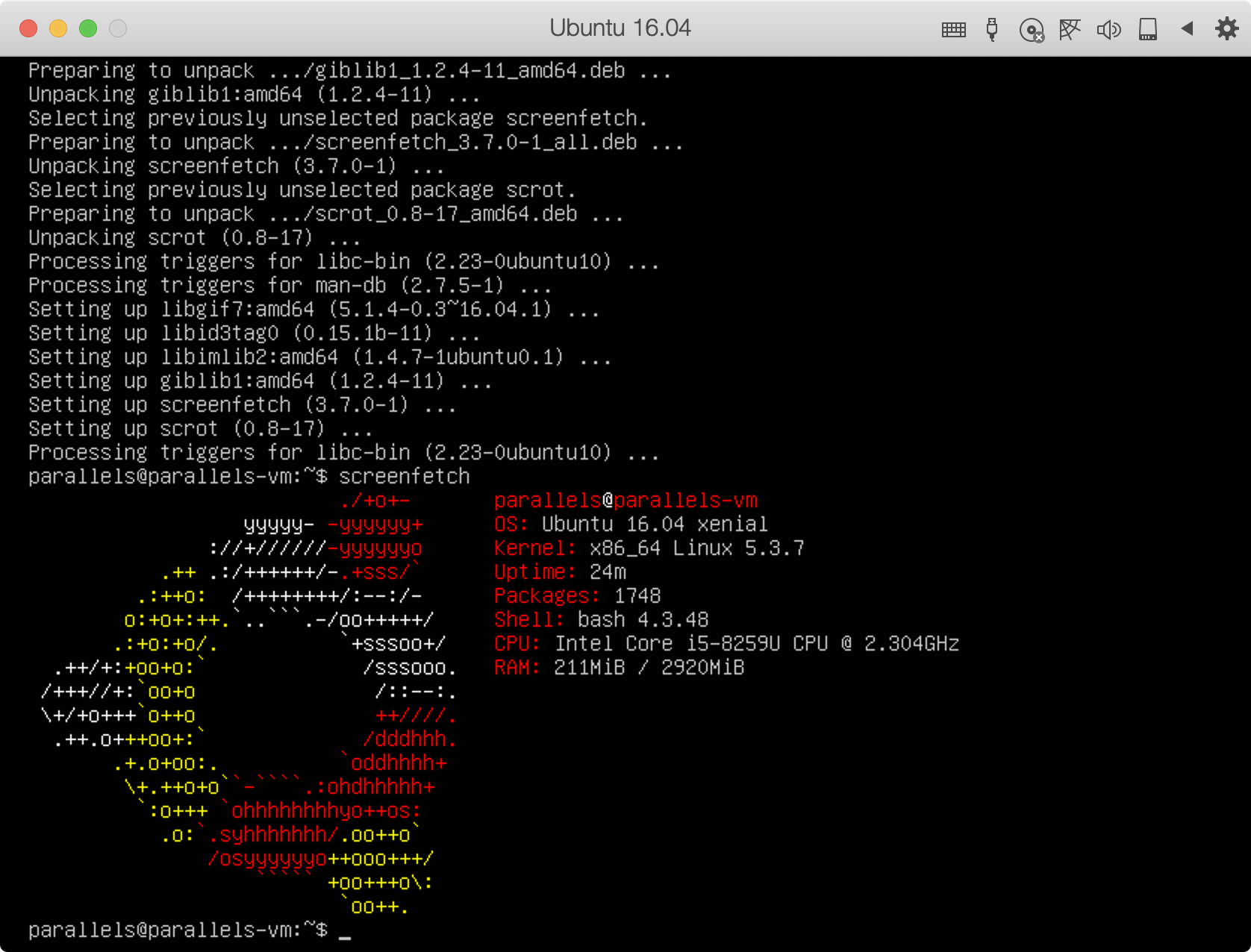
创建进程，加入就绪队列时更新vruntime

B)

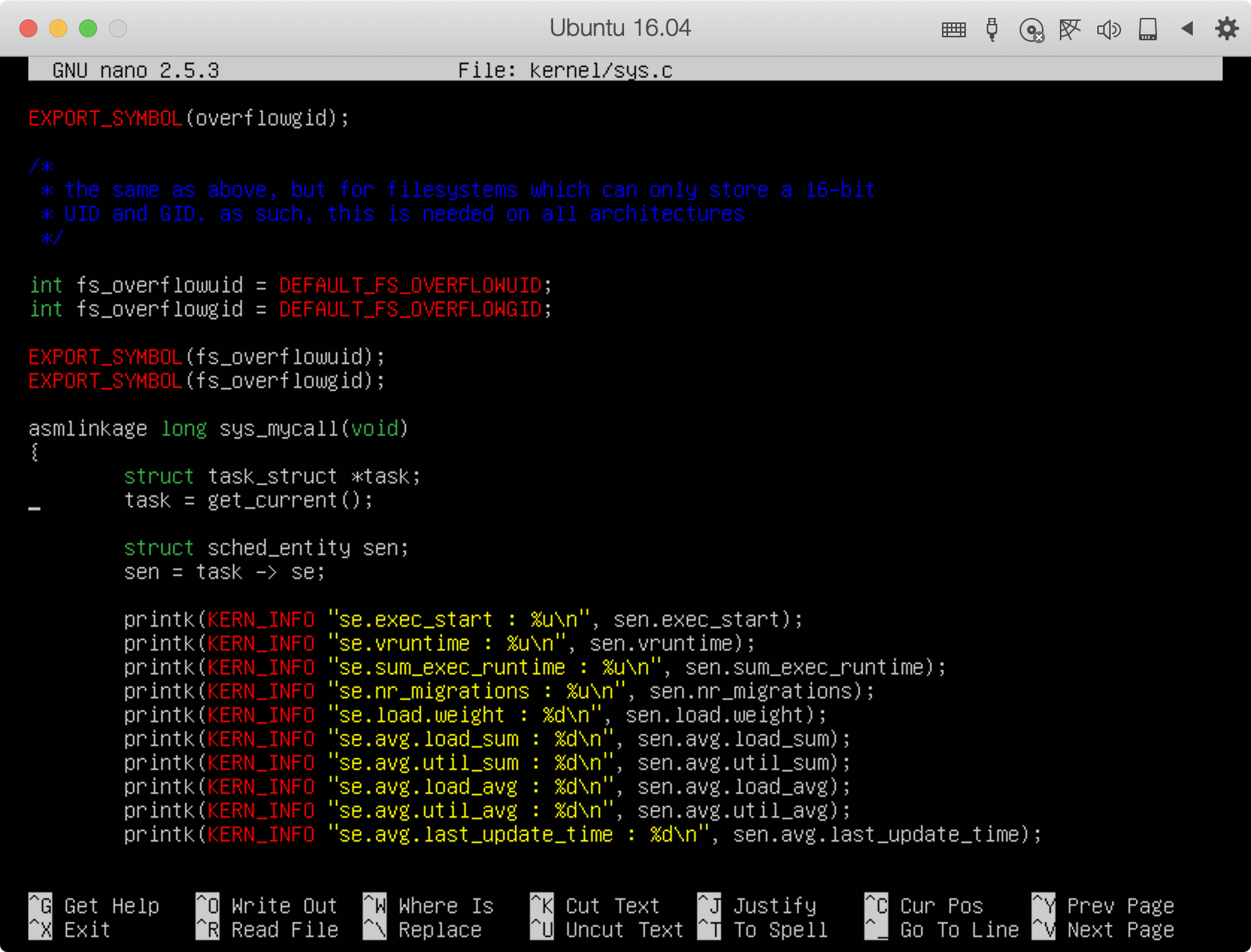
编译安装最新linux-5.3.7内核



运行screenfetch命令



在kernel/sys.c文件中添加系统调用



运行mycall.c程序测试系统调用，dmesg命令显示结果

