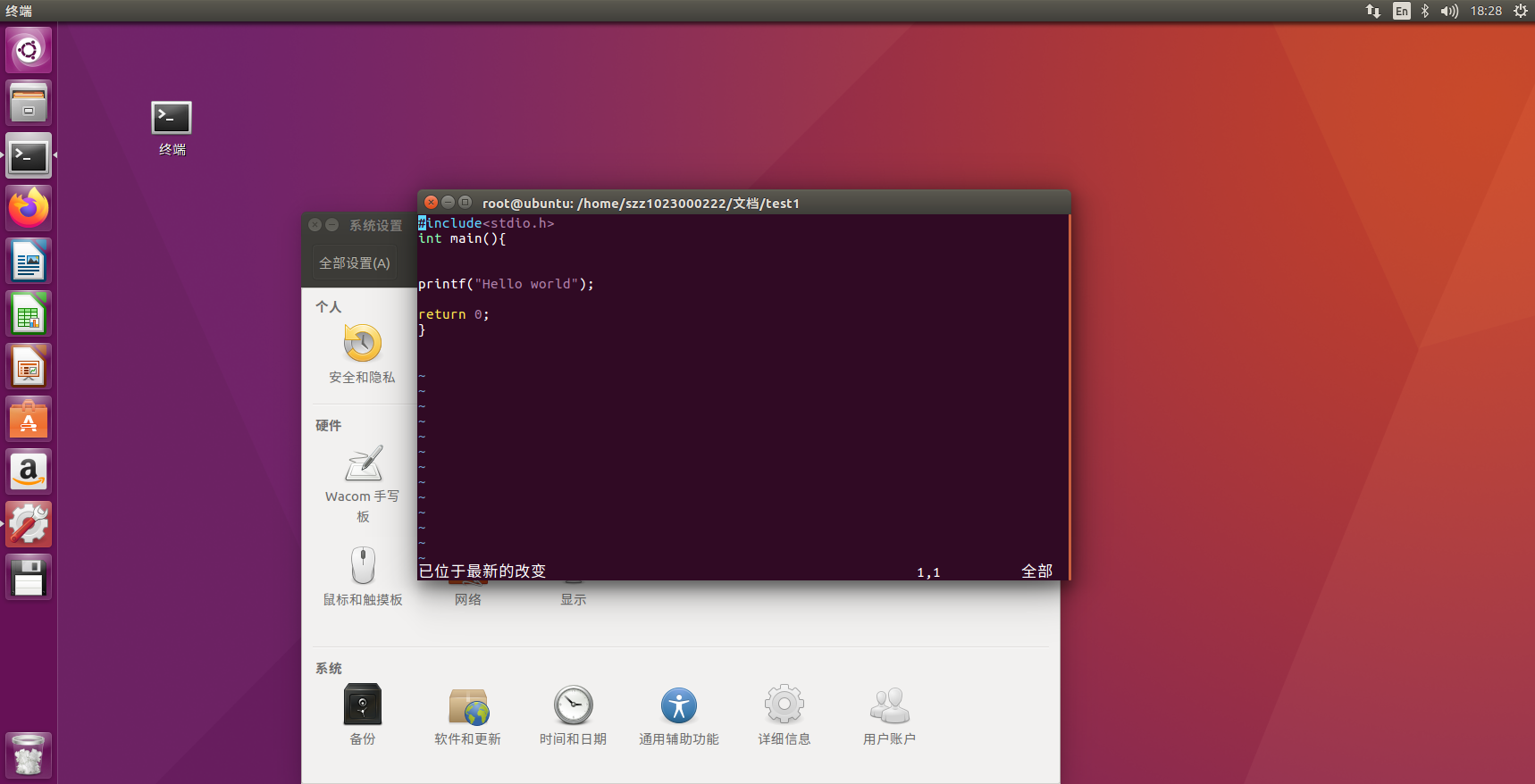
1.苏忠朝 102300222 szz



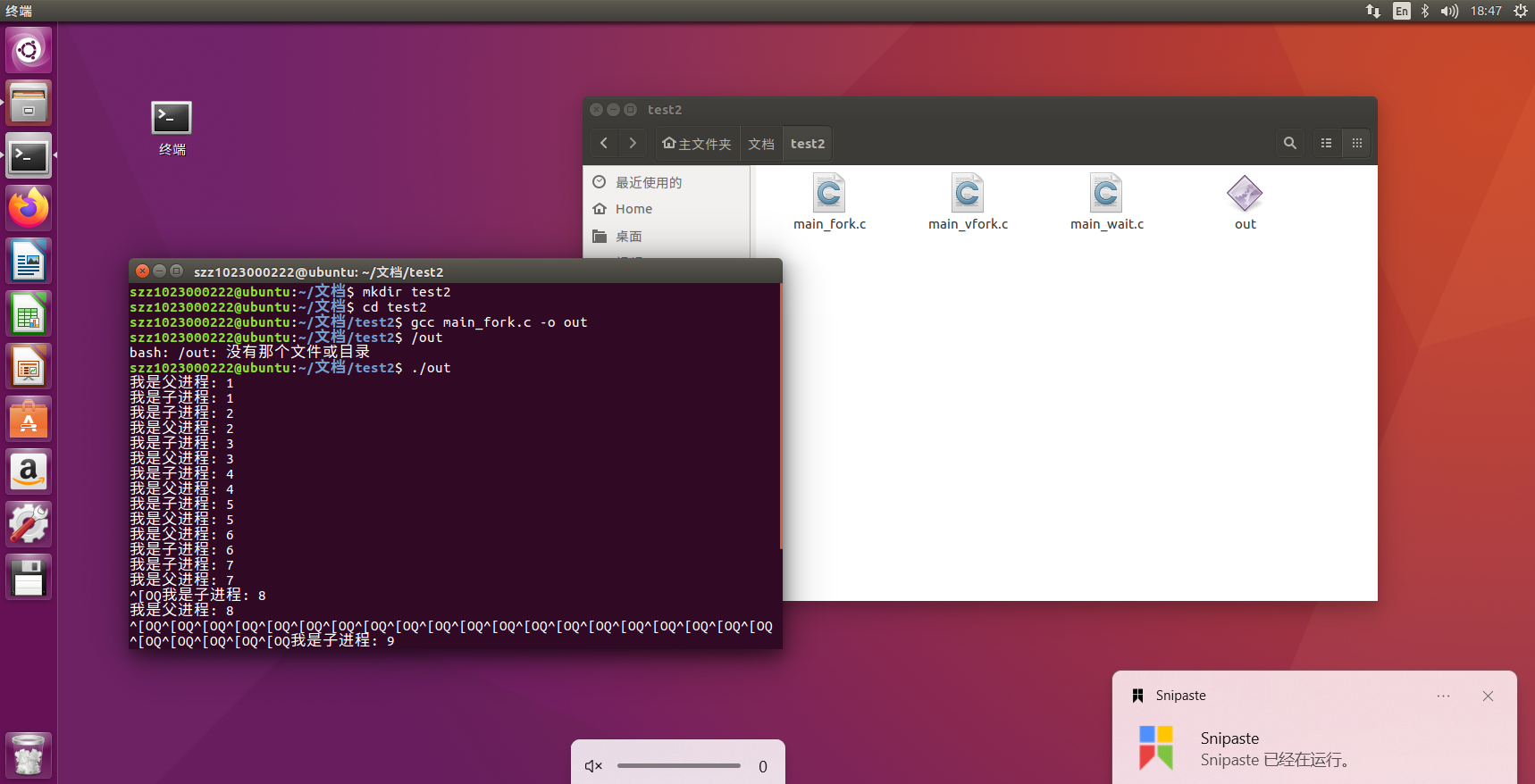
2.vim界面



3.运行main\_fork.c

Gcc main\_fork.c -o out

./out



题目：并行计算前20个质数的和

题目描述：编写一个 C 程序，将从2开始的前20个质数存储到数组中，然后使用 vfork()或fork() 创建5个子进程，每个子进程计算一部分质数的和。父进程等待所有子进程完成后，将所有子进程的计算结果相加得到最终结果。（注意必须使用子进程进行并行计算，并要求输出最终的结果）

我没有用通道，用的vfork而不是fork。

如果用fork, 每个子进程都有自己的独立地址空间。也就是说，子进程对 sums[i] 的修改不会影响到父进程的 sums[]，所以父进程中的 sums[] 数组仍然是初始化为 0，最终总和是 0。  
fork():

创建一个父进程的完全副本，包括数据段、堆栈段等。

父子进程拥有独立的地址空间，互不影响。

子进程可以通过写操作修改自己的数据而不影响父进程的数据。

vfork():

不会复制父进程的地址空间。相反，父子进程共享同一个地址空间，直到子进程通过 exec 系列函数执行新的程序或调用 \_exit() 退出。

子进程对任何数据的修改都会直接影响到父进程（因为它们共享内存），因此在子进程中不能进行复杂的操作，以免破坏父进程的状态。  
fork内存开销很大

图形用户界面, 文本

AI 生成的内容可能不正确。

图形用户界面, 文本

AI 生成的内容可能不正确。

运行结果:unbuntu有中文乱码

一共20个素数，5个进程，每个进程计算4个素数的和。

子进程Vfork共用同一块内存，父进程在循环中wait(NULL),每当一个子进程结束，就把sums[i]加到total上，最终total等于639

