1. 在/usr/src/linux/include/unistd.h 中添加 #define \_\_NR\_call 87 (这样就添加了一个名字叫做call的系统调用，系统调用号为87)，然后再在文件下面添加该函数原型。比如：int call（int flag）；（具体的可以根据你的实现来改）
2. /usr/src/linux/include/linux/sys.h 中添加 extern int sys\_call()，然后再在那个sys\_call\_table中添加sys\_call，注意顺序必须与系统调用号一致
3. 在/usr/src/linux/kernel/system\_call.s中将nr\_system\_calls由86改大，可以改为100都行（确保你的nr\_system\_calls值比你最大的系统调用号还要大）
4. 在/usr/src/linux/kernel/sched.c中修改调度算法，原先的调度函数是schedule()，你可以在其中添加新的调度算法，也可以自己重新写一个函数，看你心情；你同时可以把你添加的系统调用在这里实现，比如 int call（int flag）{；；；return0 ；}。如果你不想，那你也可以在sys.c中去实现你的新添加的系统调用。如果你还不想，你也可以随便建一个.c文件，在里面实现，不过这样你必须修改Makefile文件（具体怎么改参照百度，我忘了）
5. 回到根目录，输入指令cp /usr/src/linux/include/unistd.h /usr/include/unistd.h（目的是为了更新你gcc的函数库）
6. 进入usr/src/linux 目录下，输入 make clean（清除你原先的编译信息），然后输入make（编译你的内核代码），然后编译通过后输入 dd bs=8192 if=Image of=/dev/fd0 来生成镜像文件，然后点击reset，这样系统就修改好了（是不是很简单）
7. 编写用户程序测试，充分发挥你的才智吧！！！！

Bochs 常用指令详见实验二的指导书，这里就不说了