编译与调试命令小结

1. .c 文件的编译：

从.c 到.i:　　gcc -E hello.c -o hello.i

从.i 到.s　　gcc -S hello.i -o hello.s

从.s 到.o　　gcc -c hello.s -o hello.o

从.o 到.out　　gcc hello.o -o hello.out（或者不加 out）

--注意区分大小写！--

2. 单纯汇编文件的编译：

注意 1 中的编译指令是从.c 文件开始的，如果不是从.c 开始，那么缺乏 main 函数，gcc 就会报错。例如课堂演示时的单纯汇编文件我们都是用 as(汇编)来进行编译并使用 ld 进行链接的，示例如下：

as -g 1001.s -o 1001.o　（参数-g 是添加 gdb 调试所需信息）

ld 1001.o -o 1001.out　（因为是纯汇编代码，没有库的链接）

如果是需要链接库函数的程序，需要在 ld 指令中添加参数，目前不要求掌握。

3. GDB 调试命令

以 gdb -q 1001.out 进入调试环境，使用参数 q 的目的是避免显示过多的版本信息

第一步：在你希望程序执行时停下的指令行处设下断点，程序执行至断点停止，便于逐步调试。命令为 break，常用方式是：

break 行号（行号可以通过 list 命令查看全部指令得到）

或者

break 关键字　例如 break *_start 表示在出现_start 关键字的地方设置断点。break 也可简写为 b。

第二步：使用 run 命令（简写 r）运行程序，程序在断点处停下，并显示下一条即将执行的指令。

第三步：查看寄存器的值。
如果是查看所有寄存器，使用 info reg 命令（简写 i r）；
如果查看单一寄存器，使用 print $eax （简写 p %eax,注意这样打印出来是十进制，如果要规定查看数据的进制，p 后接 /x 表示十六进制，/t 表示二进制等，具体查阅 GDB 手册）

第四步：查看内存的值
命令为 “exam（简写 x） /_ _ _ 地址”
第一个空格是需要查看的数量；第二个空格是查看数据的单位；第三个空格是查看数据的格式。地址可以是实际的地址值，也可以是存有地址值的寄存器（$eax），也可以是变量名表示的地址（&value）
例如 x/4bx 0x08408056 表示从地址 0x08408056 开始查看 4 个字节（b）即 8056、8057、8058、8059 四个地址中的数据，并且是以十六进制（x）格式查看

第五步：使用 next（简写 n）或者 step 指令（简写 s）执行下一条指令

调试时根据即将执行的指令内容，先查看当前相关变量、寄存器、内存的值，执行下一条后，再看目标操作数的值是否发生了期望中的改变，获得了指令执行的结果，从而达到调试目的。