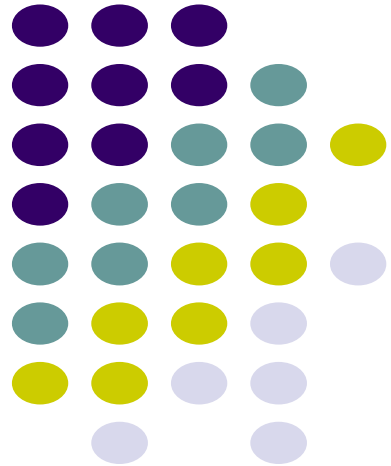
《计算机系统基础（四）：编程与调试实践》

基本实验工具的使用



基本实验工具的使用

基本gcc命令的使用

基本objdump命令的使用

基本gdb命令的使用

基本gcc命令的使用

基本gcc命令的使用

GCC是一套由GNU项目开发的编程语言编译器，可处理C语言、C++、Fortran、Pascal、Objective-C、Java等等。GCC通常是跨平台软件的编译器首选。gcc是GCC套件中的编译驱动程序名。

准备工作

1. 具备对Linux系统的了解，掌握Linux的常用命令

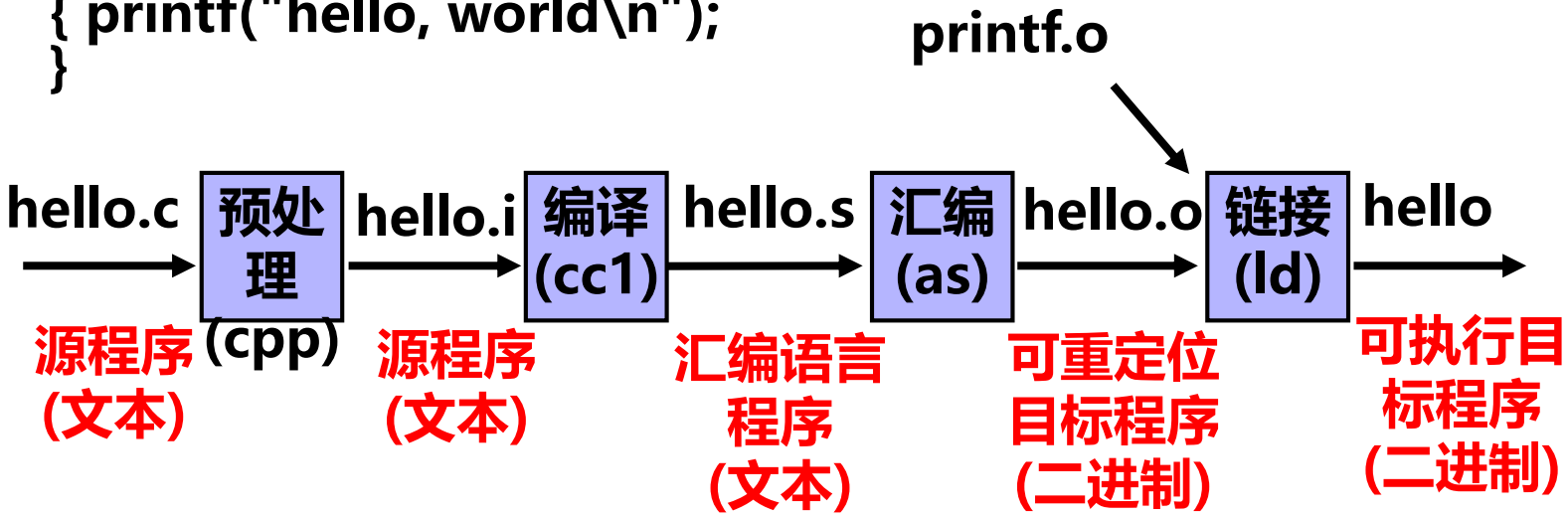
2. 若计算机是x86-64位系统，为了编译成IA-32指令集，

则请先运行下列命令：

```
sudo apt-get install build-essential module-assistant
sudo apt-get install gcc-multilib g++-multilib
```

基本gcc命令的使用

```
#include "stdio.h"
main( )
{ printf("hello, world\n");
}
```




```
gcc –E hello.c –o hello.i
gcc –S hello.i –o hello.s
gcc –c hello.s –o hello.o
gcc hello.o –o hello
```

```
gcc hello.c –o hello  //将hello.c直接编译成可执行目标文件hello
```

基本objdump命令的使用

gdbtest.c

```
#include "stdio.h"
void main( )
{  int x=3, y=5, z ;
   z=x+y;
   printf("z=%d\n",z);
   return;
}
```

```
gcc  -E -g -m32  gdbtest.c  -o  gdbtest.i
gcc  -S -g -m32  gdbtest.i  -o  gdbtest.s
gcc  -c –g -m32  gdbtest.s  -o  gdbtest.o
gcc  -O0  -m32  -g  gdbtest.c  -o  gdbtest
```

//gdbtest.o可重定位目标文件、gdbtest可执行目标文件

```
objdump –S  gdbtest.o>gdbtesto.txt
objdump –S  gdbtest>gdbtest.txt
```

//反汇编，-S 保留C语句，> 保存到文件

基本gdb命令的使用

GDB调试的基本步骤只有6步

步1：启动GDB调式工具，加载要执行的目标文件

(1) gdb 可执行目标文件　　//启动GDB调试工具，并加载程序

(2) gdb　　//启动GDB调试工具

file 可执行目标文件　　//加载程序

步2：设置断点

break main　　//在main函数的入口处设置断点

break gdbtest.c:3　　//源程序gdbtest.c的第3行处设置断点

步3：启动程序运行

run　　//程序会在断点处停下

步4：查看程序运行时的当前状态

步5：继续执行下一条指令或语句

si　　//执行一条机器指令

s　　//执行一条c语句

步6：退出调试

quit

步4和步5根据自己的需要不断地、交替地执行，达到对程序执行过程的跟踪。

基本gdb命令的使用

步4：查看程序运行时的当前状态

程序的当前断点位置：i r eip (或 i r)

通用寄存器的内容：　i r eax ebx ecx edx (或 i r)

存储器的单元内容：　x/8xb 0xffffd2bc

x/2xw 0xffffd2bc

栈帧信息

当前栈帧范围：　i r esp ebp　　//esp栈顶指针和ebp栈底指针

当前栈帧字节数：y=R[ebp]-R[esp]+4 //不是命令，是计算方法

显示当前栈帧内容：x/yxb $esp

x/zxw $esp　　//z=y/4

基本gdb命令的使用-举例

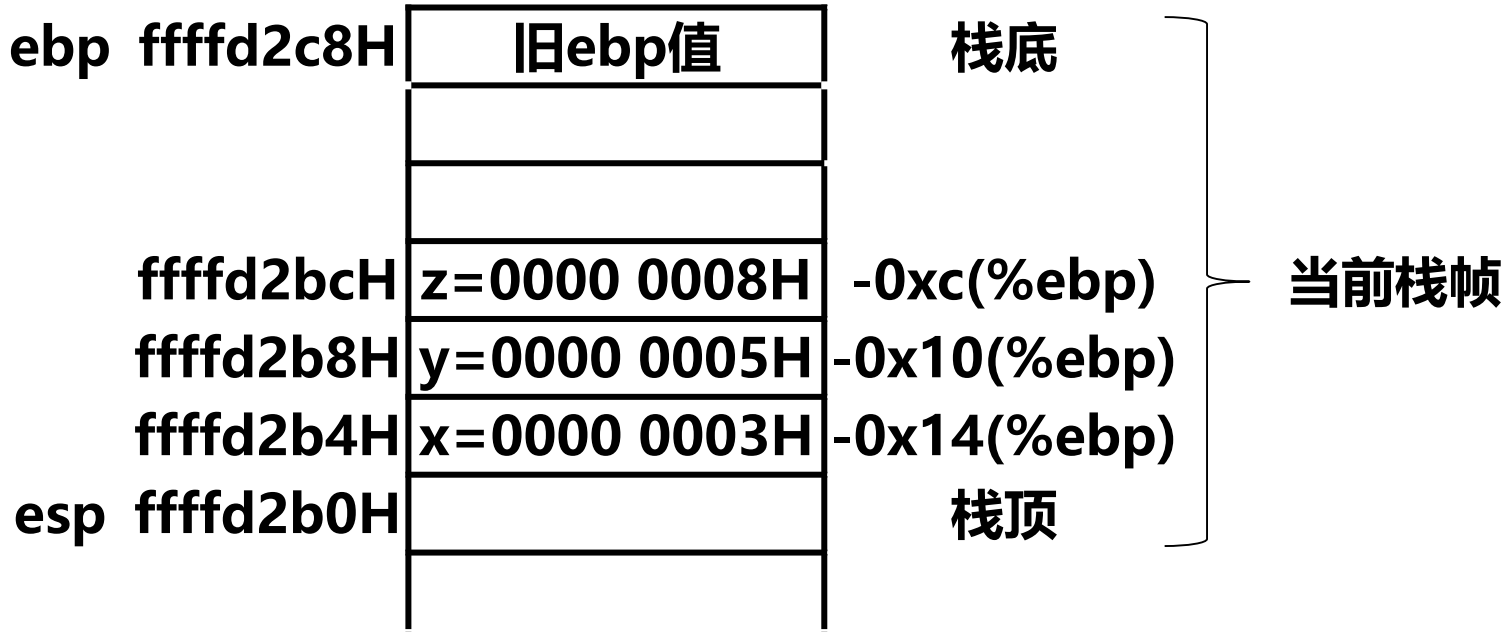
C源程序	可执行目标文件	反汇编文件
gdbtest.c	gdbtest	gdbtest.txt

步1：gdb gdbtest　　//启动GDB调式工具，进入gdbtest的调试环境
步2：break main　　//在main函数处设置断点
　　break gdbtest.c:3　　//在gdbtest.c的第3行设置断点
步3：run　　//启动运行程序
步4：si（或s)　　//执行一条指令（或c语句）
步5：i r　　//查看各寄存器的内容
步6：i r eip　　//查看eip寄存器的内容
步7：i r esp ebp　　//查看esp和ebp寄存器的内容
步8：x/yxb $esp　　//按字节显示当前栈帧内容
　　//y=R[ebp]-R[esp]+4
　或 x/zxw $esp　　//按4字节显示当前栈帧内容
　　//z=(R[ebp]-R[esp]+4)/4
　重复执行步4~步8，观察程序运行时的各种数据变化。
步9：quit　　//退出调试

基本gdb命令的使用

从2到8的自然数有几个？ 2 3 4 5 6 7 8 答案：7个

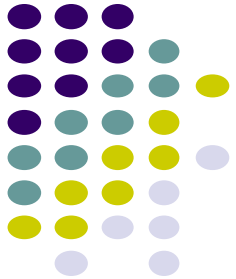
8-2=6 8-2+1=7



当前栈帧的字节数：

R[ebp]-R[esp]+4=ffffd2c8H-ffffd2b0H+4

=28个字节



谢谢！