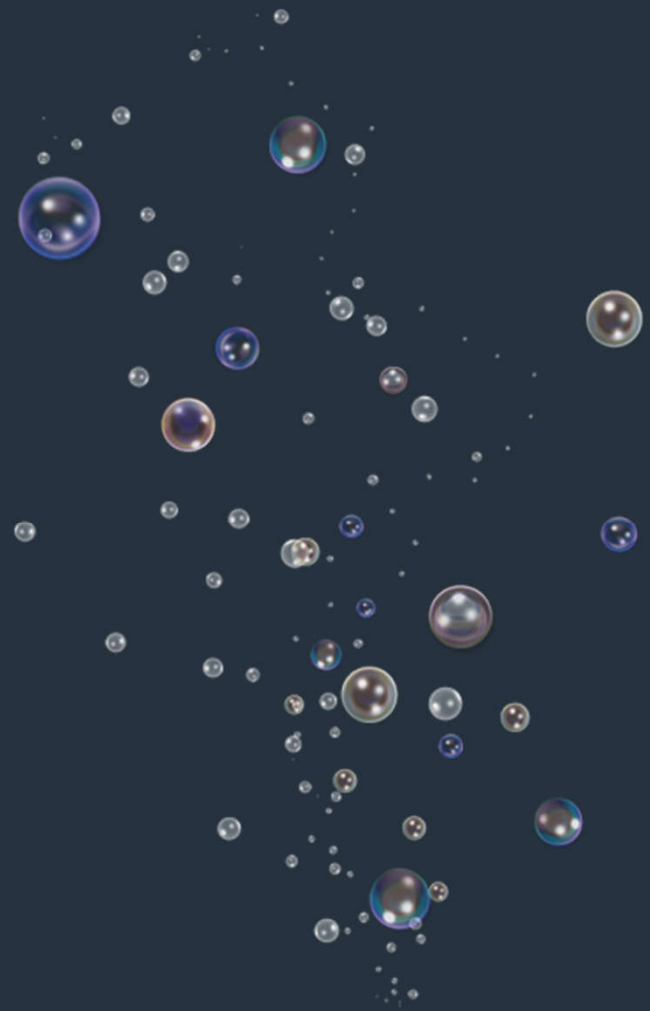


gcc的使用

大连理工大学 赖晓晨

使用gcc

- gcc –v　查看gcc版本
- gcc f1.c –o f2　(gcc f1.c)

举例:

```
#include <stdio.h>
int main()
{
    printf("hello world\n");
    return 0;
}
```

hello.c

用gcc编译多源文件程序

```
//mul1.c
#include <stdio.h>
#include   “my.h”
main()
{
        printf("hello world\n");
        f();
}
```

```
//mul2.c
#include <stdio.h>
f()
{
        printf(  “in f()\n”  );
}
```

```
//my.h
void f();
```

```
lai@dell lai> gcc mul1.c mul2.c –o m.out
```

mul1.c/mul2.c/my.h

用gcc编译多源文件程序

```
//mul1.c
#include <stdio.h>
#include   “my.h”
main()
{
        printf("hello world\n");
        f();
}
```

```
//mul2.c
#include <stdio.h>
f()
{
        printf(  “in f()\n”  );
}
```

```
//my.h
void f();
```

```
lai@dell lai> gcc mul1.c mul2.c –o m.out
```

用一条gcc命令编译多源文件程序的缺点：

每个文件都要重新编译

gcc的警告与提示

- gcc包含完整的出错检查和警告提示功能，可以帮助程序员尽快找到错误的代码。
- gcc包含30多个警告和3个警告级别。

一个不好的例子

- main函数没有使用return语句返回int类型值
- 变量var未使用

```
gcc  -Wall bad.c
```

```
#include <stdio.h>
void main(void)
{
        long long int var = 3;
        printf("this is not a standard c program!\n");
}
```

bad.c

gcc的警告选项

选 项 名	作 用
-Wcomment	如果出现注释嵌套则警告（/*后又出现/*）
-Wformat	如果传递给**printf**的参数与指定格式不匹配则警告
-Wmain	如果**main**的返回类型不是**int**或者调用**main**时参数不正确则警告
-Wparentheses	根据上下文推断，如果把（**n==10**）写作（**n=10**）则警告
-Wswitch	如果**switch**中少了一个或多个**case**分支（仅对**enum**适用）则警告
-Wunused	变量声明了但未使用，或**static**类型函数未被调用则警告
-Wuninitialized	使用的自动变量没有初始化则警告
-Wundef	如果在**#if**宏中使用了未定义的变量做判断则警告
-Winline	函数不能被内联则警告
-Wmissing-declarations	如果定义了全局函数但却没有在任何头文件中声明则警告
-Wlong-long	使用了**long long**类型则警告
-Werror	把所有警告转变为错误

gcc的库依赖

- 函数库：头文件（.h）+库文件（.so、.a）
- 系统能够识别的默认路径：
  - 头文件：/usr/include
  - 库文件：/usr/lib
- 使用非默认路径的文件需在编译时指定路径
  - 头文件用“-I”
  - 库文件用“-L”

gcc的库依赖

- 使用非默认路径的文件需在编译时指定路径
  - 头文件用“-I”
  - 库文件用“-L”，-l后的参数指明要寻找库“libkk.so”

```
gcc k.c –I /home/lai/include –o k
gcc k.c –L /home/lai/lib –lkk –o k
```

gcc的优化选项

- 代码优化指的是编译器通过分析源代码，找出其中尚未达到最优的部分，然后对其重新进行组合，目的是改善程序的执行性能。
- gcc代码优化采用“-On”选项
  - -O、 -O1：同时减小代码的长度和执行时间
  - -O2：包括-O的功能，以及指令调度等调整工作
  - -O3：包括-O2的功能，以及循环展开等工作

代码优化举例

```
#include <stdio.h>
int main()
{
        double counter;
        double result;
        double temp;
        for(counter=0; counter<4000.0*4000.0*4000.0/20.0+2030;
                                                counter+=(5-3+2+1)/4 )
        {
                temp=counter/1239;
                result=counter;
        }
        printf("the result is %1f\n",result);
        return 0;
}
```

optimeze.c

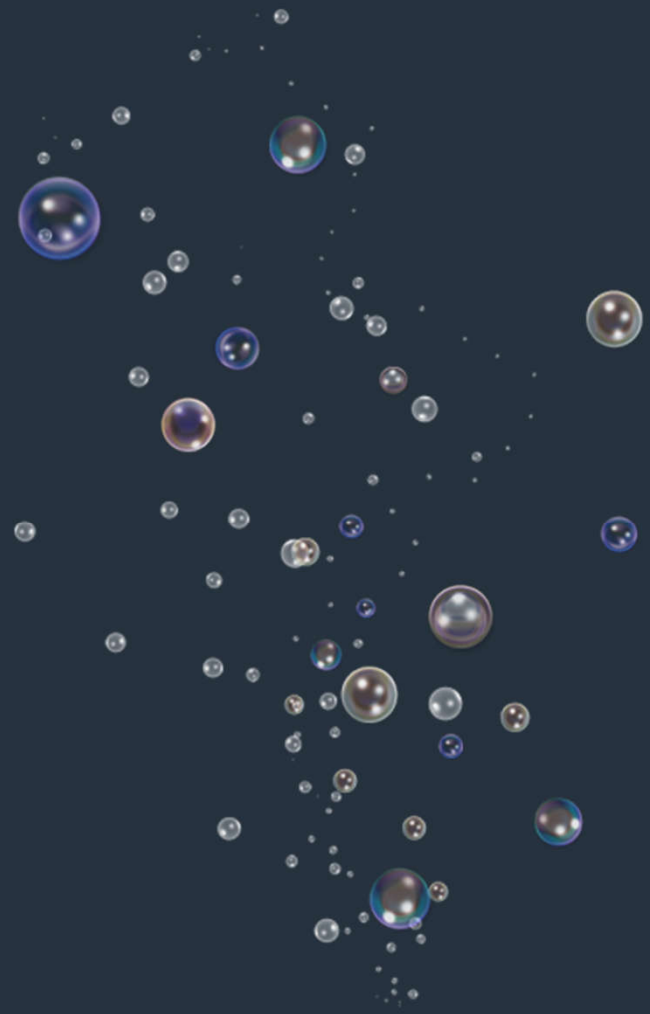
例optimize.c

- 分别使用以下选项编译程序。

```
gcc optimize.c
gcc –O optimize.c
```

- 运行程序，找到运行时间，观察差异

```
time ./a.out
```



嵌入式软件设计

大连理工大学 赖晓晨