

1、 解释Linux内核源代码的结构，以及根目录下的makefile文件。

内核源代码目录在“/usr/src/kernels”中（如果有多个内核，这里会有多个文件夹）主要目录解释如下

- (1) arch：这个子目录包含了此核心源代码所支持的硬件体系结构相关的核心代码
- (2) ipc：此目录包含了核心的进程间通讯代码
- (3) include：这个目录包括了核心的大多数include文件。另外对于每种文件支持的体系结构分别有一个子目录
- (4) init：此目录包含核心启动代码
- (5) mm：此目录包含所有的内存管理代码。与具体硬件体系结构相关的内滚管理代码位于arch/*/mm目录下，如对应于x86的就是arch/i386/mm/fault.c
- (6)

2、 查找GCC使用的库文件和头文件存放的路径。

```
[151020216@localhost ~]$ rpm -qa gcc
/usr/bin/c89
/usr/bin/c99
/usr/bin/cc
/usr/bin/gcc
/usr/bin/gcov
/usr/bin/protoize
/usr/bin/unprotoize
/usr/bin/x86_64-redhat-linux-gcc
/usr/lib/gcc
/usr/lib/gcc/x86_64-redhat-linux
/usr/lib/gcc/x86_64-redhat-linux/4.4.4
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/crtbegin.o
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/crtbeginS.o
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/crtbeginT.o
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/crtend.o
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/crtendS.o
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/crtfastmath.o
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/crtprec32.o
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/crtprec64.o
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/crtprec80.o
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/libgcc.a
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/libgcc_eh.a
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/libgcc_s.so
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/libgcov.a
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/libgomp.a
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/libgomp.so
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/libmudflap.a
/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/libmudflap.so
```

3、 编写一个C程序，其功能是用户向系统输入一个简单的问候，然后系统以字母反序将该问候显示出来。要求：编写makefile文件。

xx.c：

```
[151020216@localhost ~]$ cat xx.c
#include<stdio.h>
#include<string.h>

int main()
{
    char *str;
    gets(str);
    int n=0;
    n=strlen(str);

    for(;n>=0;n--)
    {
        printf("%c",str[n]);
    }
    return 0;
}
```

```
[151020216@localhost ~]$ █
```

Makefile文件内容:

```
[151020216@localhost ~]$ cat makefile
demol : xx.o
        gcc xx.o -o demol
xx.o : xx.c
        gcc -c xx.c -o xx.o
clean:
        rm -fr *.o demol
[151020216@localhost ~]$ █
```

Make执行结果:

```
[151020216@localhost ~]$ make
gcc -c xx.c -o xx.o
gcc xx.o -o demol
xx.o: In function `main':
xx.c:(.text+0x10): warning: the `gets' function is dangerous and should not be
sed.
[151020216@localhost ~]$ ./demol
abc
cba[151020216@localhost ~]$ █
```

4、编写一个C程序，其功能是从键盘输入三个整数，然后系统输出最大值、最小值和平均值。要求：使用函数功能及makefile文件。

(1) 新建yy.c文件，命令为vim yy.c 文件内容如图

```

#include<stdio.h>
int main()
{
    int num[3]={1,2,3};
    int i=3;
    int max=0;
    int min=0;
    double avg=0.0;
    int sum=0;

    scanf(" %d%d%d",&num[0],&num[1],&num[2]);
    max=num[0];
    min=num[0];
    for(i=0;i<3;i++)
    {
        if(num[i]>max)
        {
            max=num[i];
        }
        if(num[i]<min)
        {
            min=num[i];
        }
        sum+=num[i];
    }
    avg=sum/3.0;
    printf("max:%d\n",max);
    printf("min:%d\n",min);
    printf("avg:%lf\n",avg);
    return 0;
}

```

(2) 新建makefile文件

```

[151020216@localhost Desktop]$ cat makefile
demo2 : yy.o
        gcc yy.o -o demo2
yy.o :yy.c
        gcc -c yy.c -o yy.o
clean:
        rm -fr *.o demo2
[151020216@localhost Desktop]$

```

(3) 编译，测试结果如图所示

```

[151020216@localhost Desktop]$ make
gcc -c yy.c -o yy.o
gcc yy.o -o demo2

```

```
[151020216@localhost Desktop]$ ./demo2
10 25 30
max:30
min:10
avg:21.666667
[151020216@localhost Desktop]$
```

然后执行 make clean 命令会删除掉生成的.o 文件