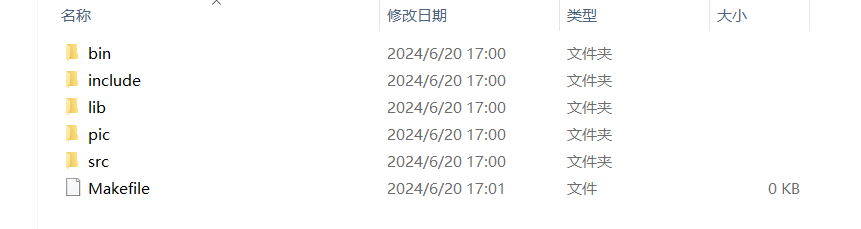
**一、工程目录管理**

在写一个很复杂的项目时，不可能将所有的代码都写在一个.c文件里面，为了方便管理和维护，可以使用工程目录管理

创建如下文件夹：



bin：存放可执行程序以及一些配置文件

include：用于存放自定义的头文件

lib：存放库文件

pic：存放图片等（以练习为例）

src：存放源代码

Makefile：管理工具，编译整个工程

all:

    arm-linux-gcc ./src/\*.c -o ./bin/target -I ./include -lpthread

**二、头文件**

头文件的内容：

1、函数的声明（普通函数、静态函数、内敛函数lnline）

2、宏定义 #define

3、结构体、联合体、枚举的声明

4、其他头文件

6、变量的声明

**三、线程**

线程：线程是任务调度的基本单位

**线程API：线程创建**

头文件：

#include <pthread.h>

函数原型：

int pthread\_create(pthread\_t \*thread, const pthread\_attr\_t \*attr,

void \*(\*start\_routine) (void \*), void \*arg);

参数列表：

thread：线程的ID

attr：线程属性，NULL

start\_routine：函数指针，指向线程的任务函数

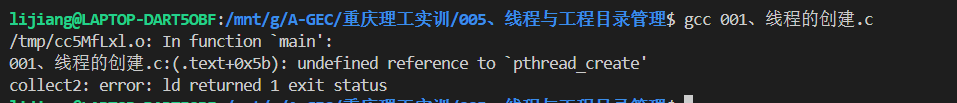
arg：线程的参数

返回值：

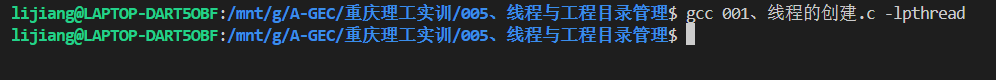
成功：0

失败：错误码

线程编译的时候报错



解决办法：连接线程库，-lpthread



示例代码：

#include <stdio.h>

#include <pthread.h>

#include <unistd.h>

// 线程的任务函数

void\* task1(void\* arg)

{

    while (1)

    {

        printf("任务1进行中\n");

        sleep(1);

    }

}

int main(int argc, char const \*argv[])

{

    pthread\_t A;

    pthread\_create(&A, NULL, task1, NULL);  // 创建一个线程，任务函数位task1，没有参数

    while (1)

    {

        printf("主任务进行中\n");

        sleep(1);

    }

    return 0;

}

**线程的销毁：**

头文件：

#include <pthread.h>

函数原型：

int pthread\_cancel(pthread\_t thread);

示例代码：

#include <stdio.h>

#include <pthread.h>

#include <unistd.h>

// 线程的任务函数

void\* task1(void\* arg)

{

    while (1)

    {

        printf("任务1进行中\n");

        sleep(1);

    }

}

// 线程的任务函数

void\* task2(void\* arg)

{

    while (1)

    {

        printf("任务2进行中\n");

        sleep(1);

    }

}

int main(int argc, char const \*argv[])

{

    pthread\_t A, B;

    pthread\_create(&A, NULL, task1, NULL);  // 创建一个线程，任务函数位task1，没有参数

    pthread\_create(&B, NULL, task2, NULL);  // 创建一个线程，任务函数位task2，没有参数

    int n;

    while (1)

    {

        scanf("%d", &n);

        if (n == 1)

            pthread\_cancel(A);

        if (n == 2)

            pthread\_cancel(B);

    }

    return 0;

}

**线程的传参：**

#include <stdio.h>

#include <pthread.h>

#include <unistd.h>

struct canshu

{

    int a;

    char c;

    double d;

};

// 线程的任务函数

void\* task1(void\* arg)  // int a   char c   double d

{

    // int a = \*(int \*)arg;

    // int\* a = (int \*)arg;

    // \*a = 20;    // 修改实参的值

    struct canshu canshu = \*(struct canshu\*)arg;

    while (1)

    {

        // printf("a = %d\n", \*a);

        // sleep(1);

        printf("a = %d\n", canshu.a);

        printf("c = %c\n", canshu.c);

        printf("d = %lf\n", canshu.d);

        sleep(1);

    }

}

int main(int argc, char const \*argv[])

{

    // int a = 10;

    struct canshu canshu1;

    canshu1.a = 10;

    canshu1.c = 'A';

    canshu1.d = 3.14;

    pthread\_t A;

    pthread\_create(&A, NULL, task1, &canshu1);  // 创建一个线程，任务函数位task1

    while (1)

    {

        // printf("a = %d\n", a);

        // sleep(1);

    }

    return 0;

}