Write a c/c++ program to implement copy one diretory with multi-threads

Target

1. Write a c/c++ program with multi-threads
    1. 多线程拷贝目录及子目录
2. To implement copy one diretory and it's subdiretories with multi-threads
3. GCC
4. Test directory: 使用最新的Linux Kernel来测试(从www.kernel.org下载最新的linux内核)
    1. https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.10.10.tar.xz
    2. extract linux-6.10.10.tar.xz to linux-6.10.10 directory,
    3. and copy linux-6.10.10 directory to linux-6.10.10bak directory
5. Verify that the directory copy is correct
6. Compare the advantages and disadvantages of two different solutions: multi-process and multi-threaded

Tools

Install GCC Software Colletion

```
sudo apt-get install build-essential
```

How to use GCC

- gcc and make

比较两个目录是否内容相同diff

务必加上r参数

```
diff -r DirA DirB
```

get the total time of program execution

```
$ time pwd
/mnt/test2linux
```

```
real    0m0.000s
user    0m0.000s
sys 0m0.000s


$ time tar xvJf  linux-6.10.10.tar.xz

real    0m28.554s
user    0m7.738s
sys 0m3.554s
```

structure of directory

```
struct dirent
{
    ino_t d_ino; //d_ino 此目录进入点的inode
    ff_t d_off; //d_off 目录文件开头至此目录进入点的位移
    signed short int d_reclen; //d_reclen _name 的长度, 不包含NULL 字符
    unsigned char d_type; //d_type d_name 所指的文件类型 d_name 文件名
    har d_name[256];
};

opendir()
readdir()
closedir()
```

posix thread

```
#include <pthread.h>
pthread_create()
```

posix mutex互斥量

```
#include <pthread.h>
pthread_mutex_t mutex =PTHREAD_MUTEX_INITIALIZER;
int pthread_mutex_init(pthread_mutex_t *restrict mutex,
                       const pthread_mutexattr_t *restrict attr);
int pthread_mutex_destroy(pthread_mutex_t *mutex);
int pthread_mutex_lock(pthread_mutex_t *mutex);
int pthread_mutex_trylock(pthread_mutex_t *mutex);
int pthread_mutex_unlock(pthread_mutex_t *mutex);
```

实例:

```
pthread_mutex_t mutex;

void * thread_run(void *arg)
{
    pthread_mutex_lock(&mutex);
    //////TODO
    XXXXXXXXX

    pthread_mutex_unlock(&mutex);
    return 0;
}

int main(int argc, char *argv[])
{
    pthread_t thread1, thread2;
    pthread_mutex_init(&mutex, 0);
    pthread_create(&thread1, NULL, thread_run, 0);
    pthread_create(&thread2, NULL, thread_run, 0);
    pthread_join(thread1, 0);
    pthread_join(thread2, 0);
    pthread_mutex_destroy(&amp;mutex);
    return 0;
}
```

How to do

write a c program with multi-threads to implement copy one directory and it's subdiretories, and the program also verifies the result

1. 对比三种情况的拷贝效率:

- 单进程
- 多进程
- 多线程
- 系统自带的命令:cp

2. 建议借鉴生产者消费者问题的思路

- 建立一个待拷贝文件的数组
- 一个线程负责多源目录及文件，并负责创建目标目录，以及将文件放到待拷贝文件的数组中
- 另外一些线程负责拷贝具体文件
- 线程之间需要对临界资源进行互斥访问

1. Example of traverse one directory

```
#include <dirent.h>
#include <unistd.h>
#include <stdlib.h>

int main()
```

```
{
    DIR * dir;
    struct dirent * ptr;
    /*open dir*/
    dir = opendir("/home");
    /*read dir entry*/
    while((ptr = readdir(dir)) != NULL)
    {
        printf("d_name : %s", ptr->d_name);
        if (ptr->d_type==DT_DIR){
            printf("\tDir");
        }
            printf("\n");
    }
    /*close dir*/
    closedir(dir);
    exit(0);
}
```

2. Example of multi-threads

```
#include <pthread.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
void *ThreadFunc(void *arg)
{
    static int count = 1;
    printf ("Create thread %d\n", count);
    count++;
}
main(void)
{
    int     err;
    pthread_t tid;
    while (1)
    {
           err= pthread_create(&tid, NULL, ThreadFunc, NULL);
           if(err != 0){
               printf("can't create thread: %s\n",strerror(err));
           break;
           }
          usleep(2000);
          pthread_join(tid, 0);
          break ;
    }
}
```

```
gcc pthread_test.c  -o pthread_test  -lpthread
./pthread_test
```