1. 为什么MPI\_Status里还要记录rank和tag? (1分) 提示：看教材《MPI与OPENMP并行程序设计：C语言版》9.3节找答案
2. 写一段代码，在一个8核系统上，让主线程绑定到0号CPU，子线程绑定到最后一个CPU。(2分)
3. 讲义第39页的循环，如果改成在内层循环使用omp并行，如何修改？两种方式各有何利弊？(2分)
4. rank用来表示实际接受的信息来自哪个发送方，tag可以表示实际接受的信息的标识。

#define \_GNU\_SOURCE

#include<pthread.h>

#include <stdio.h>

#include <stdlib.h>

#define handle\_error(msg) \

do {printf(msg); exit(EXIT\_FAILURE); } while (0)

void \* childFun(void \*arg)

{

cpu\_set\_t mask;

cpu\_set\_t get;

pthread\_t tid = pthread\_self();

CPU\_ZERO(&mask);

CPU\_SET(7, &mask);

if(pthread\_setaffinity\_np(tid, sizeof(mask), &mask) != 0)

handle\_error("error: child thread band CPU7 failed\n");

CPU\_ZERO(&get);

if(pthread\_getaffinity\_np(tid, sizeof(get), &get) != 0)

handle\_error("error: child thread get affinity failed\n");

for(int i = 0; i < CPU\_SETSIZE; ++i)

if(CPU\_ISSET(i, &get))

printf("child thread affinity is: %d\n", i);

}

int main()

{

cpu\_set\_t mask;

pthread\_t tid = pthread\_self();

pthread\_t child\_tid;

CPU\_ZERO(&mask);

CPU\_SET(0,&mask);

if(pthread\_setaffinity\_np(tid, sizeof(mask), &mask) == -1)

handle\_error("error: main thread band CPU0 failed\n");

int s = pthread\_getaffinity\_np(tid, sizeof(mask), &mask);

if(s != 0)

handle\_error("error: main thread get affinity failed\n");

printf("main thread affinity is: %d\n", s);

if(pthread\_create(&child\_tid, NULL, childFun, NULL) == -1)

Handle\_error("error: child thread failed to create\n");

pthread\_join(child\_tid, NULL);

}