6.11 理发店问题

int main() {

srand(unsigned(time(nullptr)));

static constexpr int initSize = 20;

int barber = 1, customers = 0, avaliable = initSize;

std::condition\_variable cv;

std::mutex m1, m2;

std::thread work([&]{

while(1) {

std::unique\_lock<std::mutex> lck(m1);

cv.wait(lck, [&]{

if(customers > 0) {

std::cout << "有顾客, 理发师开始工作\n"; return true;

} else {

std::cout << "没有顾客, 理发师睡觉\n"; return false;

}

});

// 现在可以开始理发了

std::unique\_lock<std::mutex> lck2(m2);

std::cout << "理发师结束为一位顾客理发\n";

--customers;

++avaliable;

std::this\_thread::sleep\_for(std::chrono::seconds(rand() % 3 + 1));

}

});

std::thread comes[40];

for(int i = 0;i < 40; ++i) {

comes[i] = std::thread([&, i]{

std::unique\_lock<std::mutex> lck(m2);

if(avaliable <= 0) {

std::cout << "座位已满,　顾客 " << i << " 离开\n"; return;

}

// 有座位，来了一位顾客

std::cout << "顾客 " << i << " 号等待\n";

++customers;

--avaliable;

cv.notify\_all();

std::this\_thread::sleep\_for(std::chrono::seconds(rand() % 4 + 1));

});

}

for(int i = 0;i < 40; ++i) comes[i].join();

work.join();

return 0;

}