

项目 #3: TCP 套接字编程

ECE 650 – 2023 年春季

截止日期: 2023 年 3 月 2 日星期四晚上 11:59

一般说明

1. 您将单独处理此项目。
2. 本作业的代码应在基于 UNIX 的系统中开发和测试环境，特别是 login.oit.duke.edu 上提供的 Duke Linux 环境。
3. 您必须仔细遵守此作业规范，并按要求上交所有内容（并按照描述的正确格式）。由于班级规模较大，因此需要这样做才能提高评分效率。对于此任务，测试将自动进行。如果您没有按照提交材料（文件名、程序输出等）的确切说明进行操作，将被扣分。
4. 如果您在截止日期前在 Gradescope 上提交项目，您就有机会预览我们的自动评分器测试的程序结果。然后，您可以在截止日期前改进您的课程并重新评分。

注意：此作业包括编程部分（本文档）和书面部分（单独的文件）。一定要同时提交！

概述

在此作业中，您将开发一对程序，这些程序将交互以模拟游戏，如下所述。游戏很简单，但作业会让您亲身实践创建多进程应用程序、处理命令行参数、设置和监控进程之间的网络通信通道（使用 TCP 套接字）以及读取/写入进程之间的信息跨套接字的进程。

将被建模的游戏称为**热土豆**，其中有一些玩家快速将土豆从一个玩家扔给另一个玩家，直到随机点，游戏结束并且拿着土豆的玩家就是“它”。目标不是游戏结束时拿着土豆的那个人。

在此作业中，您将创建一个“玩家”进程环，这些进程将通过周围的土豆。因此，每个玩家进程都有一个左邻居和一个右邻居。还将有一个“ringmaster”进程启动每场比赛，报告结果，并关闭比赛。

您将从事的游戏遵循以下过程：

1. 首先，领班创建一个“马铃薯”对象，并用一定的跳数进行初始化，并将马铃薯发送给随机选择的玩家。
2. 每次玩家收到马铃薯时，玩家都会减少跳数并将玩家的 ID（这是分配给每个玩家的 ID 号，如下面的通信机制部分所述）附加到马铃薯。然后，这个拿着土豆的玩家将根据剩余的跳数选择下一步，如下所示：

A. 如果剩余跳数大于零，则玩家将随机选择一个邻居并将土豆发送给该邻居。

- b. 当跳跃计数器达到零时，游戏结束。拿着土豆的玩家将它发送给领班，表示游戏结束。ringmaster 将游戏的痕迹打印到屏幕上（使用附在土豆上的玩家身份），然后关闭游戏（通过向每个玩家发送一条消息，表明他们可以在游戏结束时关闭）。

任务是创建一个 ringmaster 进程和一些玩家进程，然后玩游戏并优雅地终止所有进程。您可以从交互式 shell 显式创建每个进程；但是，玩家进程必须在游戏结束时干净地退出，以响应 ringmaster 的命令。

根据以上描述，**指挥官**一方将负责以下事项：

1. 建立网络套接字连接并玩家数量并向每个玩家提供相关信息（详情请参见下面的通信机制部分）
2. 如上所述创建一个“土豆”对象
3. 随机选择一名玩家，将“土豆”发送给选中的玩家
4. 游戏结束时（当指挥官从“它”的玩家手中接过马铃薯时），在屏幕上打印马铃薯的踪迹

5.通过向每个玩家发送消息来关闭游戏

这**玩家**一方将负责以下事项：

1、建立三个网络socket连接进行通信：

A。玩家在左边

b. 玩家在右边

C。与指挥官

2. 继续收听三个频道，因为“土豆”可以到达这三个频道中的任何一个。请注意，命令和重要信息也可能会从指挥官那里接收到。

3.根据游戏规则妥善处理收到的一切。

通讯机制

在此作业中，您将使用 TCP 套接字作为 ringmaster 和播放器进程之间的通信机制。您的程序必须完全使用此处描述的命令行参数。ringmaster 程序被调用如下所示，其中 玩家人数必须大于 1 并且跳数必须大于或等于零且小于或等于 512（确保验证您的命令行参数！）。

请注意，这里的零是有效的跳数。在这种情况下，游戏必须创建进程环。创建环后，环长立即关闭游戏。

服务器程序被调用为：

```
ringmaster <port_num> <num_players> <num_hops> (示例： ./
ringmaster 1234 3 100)
```

播放器程序被调用为：

```
player <machine_name> <port_num> (例如： ./player vcm-
xxxx.vm.duke.edu 1234)
```

在哪里机器名是运行 ringmaster 进程的机器名称（例如 login-teer-03.oit.duke.edu），并且端口号是分配给 ringmaster 进程的端口号，它用于打开玩家连接的套接字。如果有 N 个玩家，每个玩家都有一个 ID $0, 1$ 个, 2 个, 到 $N-1$ 个。

作为设置游戏的一部分，领队可以提供每个玩家连接到他们的左右邻居所需的玩家 ID 和其他信息。玩家连接在环中，使得玩家的左邻居我是玩家 $i-1$ 正确的邻居是玩家 $i+1$ 。播放器 0 是玩家的右邻 $N-1$ 个, 和播放器 $N-1$ 个是玩家 0 的左邻居。

沟通机制资源：

请参阅我们关于 TCP 套接字的讲义和示例代码，以在 ringmaster 和玩家之间建立通信。你可以找到 [06 - tcp_example.zip](#) 在 sakai 资源下研究建立 TCP 通信的一般过程。

[Beej 的网络编程指南](#) 是本作业的极好参考资料。

您还会发现，您需要对来自 ringmaster 和 player 进程的一组文件描述符使用 “select” 调用，以便了解何时将某些信息写入一组套接字连接中的一个。您可能还会发现功能 “获取主机名” 和 “gethostbyname” 有助于建立环中相邻玩家之间以及玩家与领队之间的联系。“getsockname” 将有助于找到您的客户端程序必须监听的端口。

您还需要创建随机数（例如 0 到 N）。为此，您可以使用 兰德（） 称呼。您的代码应该首先播种随机数生成器：

```
srand((unsigned int)time(NULL)+player_id);
```

然后您可以使用以下方法生成一个介于 0 和 N-1 之间的随机数： `int random = rand() % N;`

输出：

您创建的程序必须严格遵循以下说明。如果您偏离预期，将会影响您的成绩。

下面介绍**全部**的输出指挥官程序。没有任何其他输出。

最初：

土豆 指挥官

球员 = <数字>

啤酒花= <数字>

与玩家建立连接后（即每个玩家应向 ringmaster 发送一些初始消息以指示其已准备就绪并可能提供有关该玩家的其他信息）： 玩家 <number> 已准备就绪

当向第一个随机选择的玩家发射土豆时：

准备开始游戏，发送土豆给玩家 <number>

当它拿回土豆时（游戏结束时）。球员号码。列表中没有空 跟踪是逗号分隔的列表格或换行符。的痕迹

土豆：

<n>,<n>,

示例 Ringmaster 输出：

土豆 指挥官

球员 = 3

啤酒花= 200

球员 1 准备比赛 球员 0 准备比赛 球

员 2 准备比赛

准备开始游戏，将土豆发送给玩家 2 土豆的踪迹：

2,1,2,0,2,1,0,2,...

下面介绍**全部**的输出玩家程序。没有任何其他输出。

在收到来自 ringmaster 的初始消息告诉玩家游戏中的玩家总数以及可能的其他信息（例如关于该玩家的邻居的信息）后： 在 <number> 个玩家中连接为 <number> 个玩家

将马铃薯转发给其他玩家时： 发送土豆给
<number>

当达到跳数时： 我就是

示例播放器输出：

玩家 0（在这个例子中不是最后一个玩家）： 作为玩家连接 3 个
玩家中有 0 个正在发送土豆给 2

发送土豆给 1 发送土豆给 1
发送土豆给 2 发送土豆给 2
发送土豆给 2。..

发送土豆给 1

类似于 Player 2

玩家 1（在这个例子中是最后一个玩家，即在最后一跳收到土豆）： 以玩家身份连接 3 个玩家中有 0 个正在发送土豆给 0

发送土豆给 2 发送土豆给 0
发送土豆给 0 发送土豆给 2
发送土豆给 2 发送土豆给
2。..

发送土豆给 0 我就是它

详细的提交说明

您提交的内容将包括源代码文件和您创建的 Makefile。您的文件应至少包含以下内容：

1. **指环王.c**–如上所述的 ringmaster 程序的源代码。
2. **播放器.c**–如上所述的播放器程序的源代码。
3. **生成文件**–将 ringmaster.c 和 player.c 分别编译为名为 ringmaster 和 player 的可执行程序的 makefile
4. **土豆.H**–包含马铃薯结构的源代码文件，可以在玩家之间以及玩家和 ringmaster 之间通过 TCP 套接字发送。

您将向 Gradescope 提交一个 zip 文件。

您应该创建一个名为热土豆首先，包含以下文件并压缩。ringmaster.c player.c potato.h Makefile 您可以提交您的程序以在 Gradescope 上进行预测试。

评分标准

建立服务器和播放器之间的连接：20% 编程部分的测试用

例：50%

写作部分：30%

常问问题

1. Q：可以用C++吗？

A：可以，但需要提供可编译的Makefile。

2. Q：游戏是否应该在不同机器上跨网络运行？

答：是的。网络套接字必须在网络上工作，因此，在不同的机器上运行 ringmaster 和 player 应该仍然允许正常运行的游戏。

3. 问：是否会针对 Valgrind 检查我们的代码是否存在内存错误？

答：不直接。但是，这些问题可能会在其他测试中作为实际问题出现。此外，始终建议编写无泄漏代码。

一些常见问题及其解决方法：

- 在套接字上发送未初始化的字节：使用内存集()初始化数据缓冲区。
- 忘记释放在堆上分配的对象：例如忘记 免费地址信息()后获取地址信息 ()，免费 () 后分配 ()，ETC。

如何使用 valgrind 捕捉以上问题：
valgrind -v --leak-check=full --track-origins=yes

4.问：如何使用获取地址信息（）？

答：查看手册页。确保遍历结果获取地址信息()用于绑定等，如手册页中所示。这是为了帮助克服无效主机名等的实际限制。

5. 问：我应该绑定哪个地址和端口来接受传入连接？

答：您绑定的地址将是客户端连接的地址。例如，**如果绑定到127.0.0.1，其他人只能通过连接到127.0.0.1来连接，这限制了客户端在同一台机器上。**

但是，如果您绑定到0.0.0.0 (又名INADDR_ANY),客户端可以使用任何地址连接到您，包括127.0.0.1和机器的公共IP（如果有）。因此，最通用的解决方案是绑定到地址0.0.0.0。如果您有要绑定的特定端口号，请使用该端口号。

否则，如果您在
sin_port调用 bind 时，操作系统会为您分配一个端口。之后，您可以使用 getsockname()查看套接字绑定到哪个端口。

6. Q：如何知道自己的IP地址？

答：命令ip地址可以显示您每个网络接口的IP地址。

就项目3而言，指挥和玩家都不必这样做。玩家可以通过解析在参数中给出的主机名来知道ringmaster的地址。领队也可以知道玩家的地址；看下一个问题。

7. 问：如果客户端连接到我，我怎么知道它的IP地址？

答：接受系统调用将填写地址参数，这是一个套接字地址包含客户端IP地址的结构。

8. 问：我的笔记本电脑无法接受传入连接。

答：您的笔记本电脑可能没有其他人可以识别的公共IP地址和/或主机名。解决这些问题超出了作业3的范围。不过，您可能对以下阅读材料感兴趣：

[网络地址解读](#),[转发端口](#),[非军事区](#)。

出于此任务的目的，最好在具有公共IP和/或主机名的机器上测试您的程序。VCM是一个不错的选择，我相信你至少应该能够获得2个VM。

9. 问：我的系统对我可以接受的传入套接字连接数有最大限制。这个可以吗？

答：是的。在接受ringmaster的呼叫，您可能只能支持 1020 传入连接。这也可以使用检查 `ulimit -n`。您不必修改或担心这一点。

10.问：我在乎什么发送（）和接收（）返回？

答：是的。请参阅两者的手册页。此外，请确保处理错误条件和套接字关闭情况，例如 -1 表示错误，0 表示接收时套接字关闭，等等。有关更多详细信息，请参见手册页。

11.问：什么是选择（）？我在乎吗？

答：看男人选择。选择（）是一个允许您查看哪些文件描述符已准备好向您发送内容的函数。由于此游戏涉及每个玩家都与另外两个玩家相连，并且您可能会从任何一方获得土豆，因此您需要使用此功能来检查从何处接收数据。

另外，请确保重置读文件这个函数每次调用select之后使用的结构，它只包含当前正在向您发送数据的套接字的fds。

12.问：发送（）无法发送我的完整消息。

答：这不是错误。这就是 `send()` 的设计方式。在不使用额外库的 C/C++ 中，基本的发送（）不保证消息中指定的所有字节

缓冲区大小将被发送。成功时，这些调用返回

字节发送。看[发送手册页](#)。

您可能需要使用一些循环来确保发送整个消息缓冲区。

13.问：接收（）未能收到完整的消息。

答：同上。但是，您可以使用消息等待如果您确定需要接收的字节数。

14. Q: socket buffer应该发送什么数据结构？

A: 套接字缓冲区取一个字节流。有一个协议可以发送一个字符串的大小后跟一个字符串，即缓冲区中的字符流，这可能就足够了。大小会让您知道要接收多少数据，因为这些通信的长度可能不同。随意玩耍并发明自己的协议。与字符串相比，在套接字缓冲区上发送结构需要做更多的工作。

15. Q: 跳数可以为0吗？

答：是的。“零是有效的跳数。在这种情况下，游戏必须创建进程环。创建环后，环长立即关闭游戏。” - 如通讯机制所述

注意：在这种情况下没有打印啤酒花的痕迹。

16. Q: 什么是第一跳？

A: 第一跳是从领队到第一个玩家，随机选择。

17. 问：最后一跳发生了什么？ / 游戏应该什么时候结束？

A：当马铃薯在最后一跳传递给玩家时，游戏结束。然后应将马铃薯的完整踪迹发送给指挥官——这不算一跳。游戏结束，所有进程必须关闭/完成。进程退出的顺序无关紧要。

18. Q: 当我尝试使用同一个端口连续运行 ringmaster 时，尽管我结束了程序，但仍提示端口正在使用中。我应该怎么办？

A: 需要一段时间才能再次将端口设置为空闲。您可以通过使用带有 SO_REUSEADDR 选项的 setsockopt() 来让操作系统知道立即将其设置为释放。看[参考](#)。

19. Q: 我可以在我的代码中添加一些时间延迟来同步吗？

答：不，你可能不会。必须根据需要使用良好的逻辑步骤系列和任何循环来实现同步，以确保连接成功。基于时间的连接是任意的且不可接受的解决方案。这也会导致您的代码的评分时间失控，并可能导致处罚。

20. 问：我是否必须处理连接到我的播放器/ringmaster 并发送随机消息的恶意/任意进程？

A: 不需要。您可以设计自己的通信协议并遵守它。我们不会测试您如何处理游戏中的任意连接而不是您自己的玩家等。

21. Q：游戏中途掉线、掉网等问题有规定如何处理吗？

答：不，您可以选择如何处理这些情况。然而，静默失败是一种糟糕的软件工程实践。