练习02

对于本练习,您将实施输入和输出重定向。我们将使用与
当你使用 `<` 或 `>` 作为命令的一部分时的 shell。

在我们的程序中,我们将 fork() 一个子进程。孩子将使用 execl() 来运行没有命令的 cat 命令
行参数。像这样运行时,cat 会从标准输入中读取它能读取的所有内容,并将所有内容写入
标准输出。然而,在启动 cat 之前,孩子将用读取的文件描述符替换标准输入
来自文件“input.txt”。同样,它会将标准输出替换为写入文件“output.txt”的文件描述符。
因此,当 cat 启动时,它可能认为它正在读取和写入终端,但它实际上是从 input.txt 文件复制到 output.txt 文件。这是典型的 fork()
和 execl() 用法。您创建一个新的子进程,
然后,在启动孩子应该执行的程序之前,您可以在其执行环境中进行调整
跑步。

我已经编写了 redirect.c 的部分实现。它创建一个子进程,然后让父进程等待子进程
结束。您可以添加有趣的部分。您需要添加代码才能执行以下操作。我留下了一些评论,比如
“// ...”在代码中,你需要添加一些东西。

- 我们需要打开输入文件并用读取自的文件描述符替换孩子的标准输入
  “input.txt”代替。这通常分三步完成。创建孩子后,让它使用 open()
  系统调用打开文件“input.txt”进行读取。这会给孩子一个文件描述符,从
  “输入.txt”。然后,使用 dup2() 系统调用关闭孩子的标准输入的旧文件描述符(有一个
  peprocessor 常量,STDIN_FILENO) 并将其替换为另一个文件描述符,该文件描述符从
  “输入.txt”。 dup2() 调用会同时完成这两件事。您将需要查看其文档以
  看看如何使用它。现在,关闭打开“input.txt”时得到的原始文件描述符;与 dup2()
  调用,您只是复制了该文件描述符以替换标准输入,因此您不再需要原始文件
  从 open() 返回的描述符。
- 按照类似的步骤打开“output.txt”进行写入并替换孩子的标准输出文件描述符
  使用写入“output.txt”的文件描述符。
- 在更改了用于标准输入的文件后,使用 execl() 在子进程中运行 cat 命令,并且
  输出。请记住,如果成功,execl() 将不会返回。尽管如此,放置错误消息或
  在调用 execl() 调用之后调用 fail(),以帮助您在启动 cat 时遇到问题时诊断问题。
  (这显示了为错误消息、标准错误有一个单独的输出流是很好的一个原因。在你之后
  将标准输出重定向到一个文件,你不能再用它打印到终端,但你仍然有标准错误,它仍然会进入终端。)

一旦你的程序运行起来,它不应该打印任何输出,但它应该成功地将文件“input.txt”复制到一个

新文件“output.txt”。完成后，上交已完成的 redirect.c 程序的源代码。提交
文件到名为 EX02 的 Gradescope 作业。