1. 实验目标

在xv6上实现用户程序xargs，即从标准输入中读取行并 为每行运行一次 指定的命令，且将该行作为命令的参数提供。请将代码写在user/xargs.c中。

1. 设计思路

xargs的功能是将标准输入转为程序的命令行参数。可配合管道使用，让原本无法接收标准输入的命令可以使用标准输入作为参数。

根据lab中的使用例子可以看出，xv6的xargs每次回车都会执行一次命令并输出结果，直到ctrl+d时结束；而linux中的实现则是一直接收输入，收到ctrl+d时才执行命令并输出结果。

所以借鉴之前Primes程序的设计思路，我们依然沿用“合作进程”的设计，即每次从管道读取命令后都传递给子进程（合作进程）进行执行，执行完毕后再将结果回传给父进程，并由父进程通过管道输出。

1. 代码

#include "kernel/types.h"

#include "user/user.h"

int main(int *argc*, char \**argv*[]) {

    int p[2];

    pipe(p);*// 开启具名管道，对应于命令行的管道操作符*

    char buf[1024] = {"\0"};

    char \*params[128]; *// 暂存命令行参数*

    int index = 0;

    int k;

    for (int i = 1; i < argc; i++) {

        params[index++] = argv[i];

    }

    while ((k = read(0, buf, 1024)) > 0) { *// fd=0代表管道操作符的读端*

        char tmp[1024] = {"\0"};

        params[index] = tmp;

        for (int i = 0; i < strlen(buf); i++) {

            if (buf[i] == '\n') {*// 读参数完毕，开始执行*

                if (fork() == 0) { *// 创建合作子进程*

                    exec(argv[1], params);*// 执行*

                }

                wait(0);  *// 父进程等待子进程执行完毕，并释放*

            } else {

                tmp[i] = buf[i]; *// 未遇到终止符，读入参数*

            }

        }

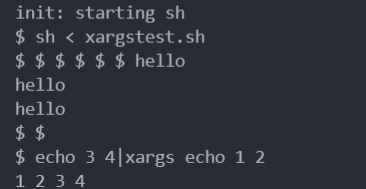
    }

    exit(0);

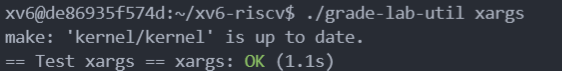
}

1. 测试

在qemu中测试：



自动化测试：



实验一完整测试：

