Задание 1. Пишем свою оболочку microsha.

Хорошая ли программа sh? A bash? Конечно, хорошие. Но большие. Ваша задача — написать программу, реализующую маааааленькое подмножество стандартных оболочек. Что она должна делать после своего запуска:

- Выдать приглашение к вводу, оно должно состоять из имени текущей директории и символа >, если она исполняется от непривилегированного пользователя или !, если от привилегированного.
- Считать стандартный ввод. Допускается любая длина входной строки, так что будьте аккуратны.
- Стандартный ввод разбивается на слова, разделённые пробелами или знаками табуляции.
- Имеется пять метасимволов, имеющие тот же смысл, что и в стандартных оболочках, >, <, |, * и ?.
- Символ | разбивает командную строку на компоненты конвейера. Как и в обычной оболочке каждая компонента представляет собой процесс, стандартный вывод которого соединяется со стандартным вводом следующего.
- Первые слова каждого компонента конвейере (даже один) имена команд. Команды могут быть внешними и внутренними. Внутренними командами являются:
 - cd [arg] перейти в директорию с именем arg, при отсутствии аргумента перейди в домашнюю директорию пользователя.
 - pwd вывести на стандартный вывод полное имя текущей директории.
 - time [cmd args] выполнить программу под именем cmd с аргументами args, после чего вывести на стандартный вывод ошибок три времени: время по обычным часам (walltime), время, которое запущенная программа провела в режиме системы и время, которое запущенная программа провела в режиме пользователя.

Остальные команды — внешние и должны пытаться запустить указанную программу с указанными аргументами. Если запуск происходит в компоненте конвейера, то метасимволы > и < недопустимы.

- Метасимволы * и ? в компоненте должны быть обработаны: каждая лексема, содержащая эти символы должна быть заменена на множество имён файлов, сопоставленных с ней.
- После запуска командной строки microsha должна дождаться завершения всех компонент и только после этого выдать приглашение на ввод очередной команды.

Чем пользоваться можно: по желанию C++ c STL (только базовые контейнеры — string, vector, deque), системными вызовами, pthread или <thread> и другими методами синхронизации.

Чем пользоваться нельзя: функциями fnmatch (она просто работает некорректно), regex, glob, <filesystem> и подобным из C++.

Последний срок сдачи задачи $N_1 - 20$ ноября.