

Четвёртое задание по практикуму для самостоятельного программирования дома

Напишите программу, которая запускается без параметров. После своего запуска программа должна выдать пользователю приглашение ко вводу (например, знак «>», «\$» и т. д.) и считать строку, введённую пользователем. Считанную строку программа должна интерпретировать как команду Shell, которую она должна исполнить. Список основных операций, которые нужно реализовать, приведён в пункте 9.3 задачника Т. В. Руденко, раздел «Общая часть», при этом аргументы у команд в конвейере допускаются. Кроме основных, в программе должна быть реализована дополнительная операция: «+». Для её описания введём понятие мультипотока:

1. Мультипоток называется конечная последовательность потоков ввода/вывода.
2. Конкатенацией мультипотоков называется мультипоток, последовательность потоков которого получена конкатенацией последовательностей потоков конкатенируемых мультипотоков.
3. Если мультипоток поступает на вход некоторой программе, то её выход должен представлять конкатенацию мультипотоков, каждый из которых является выходом программы при подаче ей на вход очередного потока из последовательности входного мультипотока. Запуски программы для потоков из последовательности должны производиться *параллельно*.
4. Конвейер осуществляет передачу выходного мультипотока предыдущего звена на вход следующему звену.
5. Ввод с клавиатуры или из файла считается мультипоток, состоящим из одного потока.
6. Если мультипоток выводится на экран или в файл, то соответствующие потоки из его последовательности выводятся последовательно, следующий начинает выводиться после завершения предыдущего.
7. При исполнении выражения `pr1 + pr2` процессы `pr1` и `pr2` запускаются параллельно, каждому из них на вход подаётся исходный входной мультипоток всего выражения. Выходные мультипоток этих процессов конкатенируются и рассматриваются как выходной мультипоток всего выражения.
8. Приоритет операции «+» выше, чем у операции конвейера «|» и у операций перенаправления потока «<», «>», «>>», реализовывать скобки («(», «)») *не* нужно.

После обработки команды программа должна выдать пользователю новое приглашение ко вводу и повторять процедуру ввода команд до тех пор, пока пользователь не введёт ключевое слово «exit».

Примеры команд:

```
ls -l >> a.txt
cat < a.txt | grep ^d > out.txt &
calendar -t 1218 -A 0
cal -y 2022
ls + cat < a.txt | grep abc + cat > b.txt
```

В результате работы последней команды в файл `b.txt` запишется то же, что и при последовательности вызовов

```
ls < a.txt | grep abc > b.txt
cat < a.txt | grep abc >> b.txt
ls < a.txt | cat >> b.txt
cat < a.txt | cat >> b.txt
```

Требования:

1. В моменты, когда программа ждёт очередной команды пользователя, процессов-«зомби» быть не должно. Это, в частности, касается и фоновых процессов. Проверить это можно, приостановив выполнение программы (`Ctrl + Z`) и вызвав команду `ps` — отображение таблицы процессов.
2. Вызовы `exec` нужно производить, используя окружение системы.
3. Для определённости можно считать, что управляющие символы «|», «<», «>», «>>», «&», «+» отделены с обеих сторон пробелами. Аргументы команд не могут содержать пробелов и управляющих символов. Для конвейеров входной файл может быть указан только для первого процесса, а выходной — только для последнего.
4. Вывод фоновых процессов на экран должен игнорироваться, доступ к клавиатуре должен быть закрыт.
5. Процессы должны быть запущены ровно столько раз, сколько потоков содержит входной мультипоток. Например, в последнем примере вызов `ls` должен быть произведён ровно *один* раз.
6. Ошибки можно не обрабатывать.

Срок сдачи: 23:59, 16.12.2021.