Упражнение 05
Управление на процеси

1. Пренасочване на стандартния вход, стандартния изход и стандартния изход за грешки

При стартирането на процес той предварително е свързан с 3 комуникационни канала :

- stdin (стандартен вход – файлов дескриптор 0),
- stdout (стандартен изход -файлов дескриптор 1),
- stderr (стандартен изход за грешки- файлов дескриптор 2).

Стандартните потоци на процесите могат да бъдат пренасочвани във файлове:

1.1. **Стандартният вход** (stdin) се пренасочва чрез "**<**" към файл, откъдето процесът чете

Примери: ***write*** `идентификатор < letter`

wc -c `< f1 ( wc -c f1 → разлика в изхода !!! )`

cat `< f1 еквивалентно с cat f1`

когато използваме cat без аргументи, командата чете низове от стандартния вход и ги принтира на стандартния изход. Горния случай вход за cat става файла f1, а изхода си е стандартния изход. Командата извежда съдържанието на файла.

Има процеси, нечетящи от файла на ст. вход:

ls `< f1`
who `< f1`
date `< f1`

1.2. **Стандартния изход** (stdout) се пренасочва с **">"** към файл. Ако файлът не съществува се създава, иначе се презаписва. Ако не искаме файла да се презаписва, а да добавяме към края му, се използва "**>>".**

cat > f1 - стандартния вход е клавиатурата, а стандартния изход става файлът f1, т.е. пишем последователно във файла f1. Това е бърз начин за създаване на файл.

ps aux >> f1 - стандартния изход за ps aux става файла f1, като сме указали информацията да се добави към края на съдържанието му.

можем да пренасочим и двата потока едновременно

cat < f1 > f2 - stdin за cat е f1, а stdout е f2, т.е. симулирахме cp командата

Има и процеси, неизвеждащи на стандартния изход

cd,
chmod,
kill

1.3 **Стандартния изход за грешки 2**- >, >>

Ако изпълним командата **`find / -name f1 > ff`** виждаме че stdout е пренасочен към файла ff, а на екрана ни излизат само съобщения за грешки. Стандартния поток за грешки се пренасочва чрез файловия му дескриптор - 2
За да пренасочим грешките в друг файл:

`find / -name f1 > ff 2>errors.txt` - така имаме файл ff, който съдържа желаните резултати и файл errors.txt, който съдържа всички грешки от изпълнението на find.

Понякога се налага освобождаване от изхода или изхода за грешки– извеждане във фиктивно у-во/ файл (/dev/null)

grep низ файл 2 > /dev/null

Горните примери съответно могат да изглеждат по този начин:

`cat 0<f1` - принтираме съдържанието на f1
`cat 1>f1` - записваме низове, подадени от стандартния вход в f1
`ps aux 1>>f1` - записваме информацията от ps aux в края на f1
Дескрипторите на стандартния вход/0/ и изход/1/ за по подразбиране и не се налага да ги пишем.

2. Конвейер между процеси.

Конвейерът е последователност от процеси, свързани чрез техните потоци, така че стандартния изход от единия процес става стандартен вход на следващия процес.

Синтаксисът е следният: `command1 | command 2 за две команди`

Пример: Искаме да преброим колко потребителя работят активно в системата. Допускаме, че всеки потребител е на един логически терминал. Това може да стане по следния начин:

```
who > temp
wc -l < temp
rm temp
```

Вместо да създаваме временен файл можем да свържем стандартния изход на who със стандартния вход на wc:

```
who | wc -l
```

`За три команди - command1 | command 2 | command 3`
Пример: Искаме да преброим на колко логически терминала работи потребител с указан акаунт
who | grep s81xxx | wc -l

`За повече команди - command1 | command 2 | command 3 | .....`

3. Стартиране на процес във фонов режим

Понякога като стартирате процес, терминала ви става "завладян" от него и не можете да продължите работата си с него, а като затворите терминала убивате процеса. Това налага използването на фонов режим за процес.

За да стартирате процес във фонов режим просто пишете **&** в края на командата.

Пример1: man & Когато изпълните man без амперсант влизате в ръководството. С амперсанта терминала ви остава свободен, като се изписва само pid-то на стартиралия процес.
За да спрете процес във фонов режим използвате kill и pid на процеса.

Пример 2: Напишете програма на C, включваща безкраен цикъл, без печат. Пуснете я за изпълнение във фонов режим и я убийте от друг терминал отворен към сървера.