

Задание №3.

Неименованные каналы

20 сентября 2016 г.

Мощным инструментом стандартных оболочек является возможность построения конвейеров из консольных команд с помощью перенаправления ввода-вывода.

Пример. Подсчитать число уникальных строк.

```
$cat file | sort | uniq | wc -l
```

Задание: модифицируйте программу из 2-го задания, чтобы оболочка позволяла строить конвейер из двух программ с их параметрами. Стоит отметить, что в стандартных оболочках нет необходимости ставить пробелы вокруг символа '|', который считается разделителем слов.

Указания: Данный функционал работает на основе системного вызова `pipe`, который создаёт неименованный канал, состоящий из 2-х файловых дескрипторов. Один на запись, а другой на чтение.

Пример использования.

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>

int main() {
    int pipefd[2];

    if (-1 == pipe(pipefd)) {
        perror("pipe");
        return -1;
    }

    pid_t pid1 = fork();
    if (pid1 == 0) {
        close(pipefd[0]);
        /* Forward STDOUT_FILENO to pipefd[0] */
    }
}
```

```

        if (-1 == dup2(pipefd[1], STDOUT_FILENO)) {
            perror("dup2");
            return EXIT_FAILURE;
        }
        if (-1 == execlp("cat", "cat", "/usr/share/dict/words", NULL)) {
            perror("execlp");
            return EXIT_FAILURE;
        }
    }

    pid_t pid2 = fork();
    if (pid2 == 0) {
        close(pipefd[1]);
        /* Forward STDIN_FILENO from pipefd[1] */
        if (-1 == dup2(pipefd[0], STDIN_FILENO)) {
            perror("dup2");
            return EXIT_FAILURE;
        }
        if (-1 == execlp("wc", "wc", "-l", NULL)) {
            perror("execlp");
            return EXIT_FAILURE;
        }
    }
    close(pipefd[0]);
    close(pipefd[1]);
    wait(NULL);
    wait(NULL);

    return EXIT_SUCCESS;
}

```