

## Лабораторная работа os\_07. Вариант № 1/2012

**Задача:** создать программу *regexfile* на языке C, имеющую следующую форму запуска:

```
regexfile file opt
```

где

*file* – имя файла, содержимое которого надо передать команде *grep* в качестве стандартного ввода. *opt* – дополнительный параметр, передаваемый программе;

В качестве параметра *opt* выступает шаблон поиска, передаваемый команде *grep*, например *\*the\**, *k[oaи]t*.

Программа должна запустить указанную команду с необходимыми параметрами с помощью системных вызовов *fork* и *exec*. Перед загрузкой указанной программы следует перенаправить ее стандартный ввод или вывод.

Перед вызовом *fork* программа должна вывести свои PID и PPID. Потомок перед любыми действиями должен вывести свои PID и PPID.

Родительский процесс после завершения потомка должен вывести информацию о том, как завершился потомок, используя макросы, определённые в *sys/wait.h*.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке.

Программа должна быть снабжена комментариями, отражающими ход ее работы, а также содержать название лабораторной работы и сведения об авторе программы.

## Лабораторная работа os\_07. Вариант № 2/2012

**Задача:** создать программу *showmsdoc* на языке C, имеющую следующую форму запуска:

```
showmsdoc file opt
```

где

*file* – имя файла, содержимое которого надо передать команде *catdoc* в качестве стандартного ввода. *opt* – дополнительный параметр, передаваемый программе;

В качестве параметра *opt* выступает название выходной кодовой страницы, использующейся командой *catdoc* в ключе *-d*, например *koi8-u*.

Программа должна запустить указанную команду с необходимыми параметрами с помощью системных вызовов *fork* и *exec*. Перед загрузкой указанной программы следует перенаправить ее стандартный ввод или вывод.

Перед вызовом *fork* программа должна вывести свои PID и PPID. Потомок перед любыми действиями должен вывести свои PID и PPID.

Родительский процесс после завершения потомка должен вывести информацию о том, как завершился потомок, используя макросы, определённые в *sys/wait.h*.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке.

Программа должна быть снабжена комментариями, отражающими ход ее работы, а также содержать название лабораторной работы и сведения об авторе программы.

## Лабораторная работа os\_07. Вариант № 3/2012

**Задача:** создать программу *filefield* на языке C, имеющую следующую форму запуска:

```
filefield file opt
```

где

*file* – имя файла, содержимое которого надо передать команде *cut* в качестве стандартного ввода. *opt* – дополнительный параметр, передаваемый программе;

В качестве параметра *opt* выступает номер поля, передаваемый команде *cut* в ключе *-f*, например 3. Дополнительно следует передать параметр *-d:*.

Программа должна запустить указанную команду с необходимыми параметрами с помощью системных вызовов *fork* и *exec*. Перед загрузкой указанной программы следует перенаправить ее стандартный ввод или вывод.

Перед вызовом *fork* программа должна вывести свои PID и PPID. Потомок перед любыми действиями должен вывести свои PID и PPID.

Родительский процесс после завершения потомка должен вывести информацию о том, как завершился потомок, используя макросы, определённые в *sys/wait.h*.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке.

Программа должна быть снабжена комментариями, отражающими ход ее работы, а также содержать название лабораторной работы и сведения об авторе программы.

## Лабораторная работа os\_07. Вариант № 4/2012

**Задача:** создать программу *ls\_out* на языке C, имеющую следующую форму запуска:

```
ls_out file opt
```

где

*file* – имя файла, в который надо записать стандартный вывод команды

*ls. opt* – дополнительный параметр, передаваемый программе;

В качестве параметра *opt* выступает имя каталога, список файлов в котором надо получить. Дополнительно следует передать параметры *-l* и *-a*.

Программа должна запустить указанную команду с необходимыми параметрами с помощью системных вызовов *fork* и *exec*. Перед загрузкой указанной программы следует перенаправить ее стандартный ввод или вывод.

Перед вызовом *fork* программа должна вывести свои PID и PPID. Потомок перед любыми действиями должен вывести свои PID и PPID.

Родительский процесс после завершения потомка должен вывести информацию о том, как завершился потомок, используя макросы, определённые в *sys/wait.h*.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке.

Программа должна быть снабжена комментариями, отражающими ход ее работы, а также содержать название лабораторной работы и сведения об авторе программы.

## Лабораторная работа os\_07. Вариант № 5/2012

**Задача:** создать программу *proc\_list* на языке C, имеющую следующую форму запуска:

```
proc_list file opt
```

где

*file* – имя файла, в который надо записать стандартный вывод команды *ps*. *opt* – дополнительный параметр, передаваемый программе;

В качестве параметра *opt* выступает какая-либо из опций команды *ps*, например *ax*, *Tug*.

Программа должна запустить указанную команду с необходимыми параметрами с помощью системных вызовов *fork* и *exec*. Перед загрузкой указанной программы следует перенаправить ее стандартный ввод или вывод.

Перед вызовом *fork* программа должна вывести свои PID и PPID. Потомок перед любыми действиями должен вывести свои PID и PPID.

Родительский процесс после завершения потомка должен вывести информацию о том, как завершился потомок, используя макросы, определённые в *sys/wait.h*.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке.

Программа должна быть снабжена комментариями, отражающими ход ее работы, а также содержать название лабораторной работы и сведения об авторе программы.

## Лабораторная работа os\_07. Вариант № 6/2012

**Задача:** создать программу *lsnetinfo* на языке C, имеющую следующую форму запуска:

```
lsnetinfo file opt
```

где

*file* – имя файла, в который надо записать стандартный вывод команды *netstat*. *opt* – дополнительный параметр, передаваемый программе;

В качестве параметра *opt* выступает какая-либо из опций команды *netstat*, например *-r*, *-n*.

Программа должна запустить указанную команду с необходимыми параметрами с помощью системных вызовов *fork* и *exec*. Перед загрузкой указанной программы следует перенаправить ее стандартный ввод или вывод.

Перед вызовом *fork* программа должна вывести свои PID и PPID. Потомок перед любыми действиями должен вывести свои PID и PPID.

Родительский процесс после завершения потомка должен вывести информацию о том, как завершился потомок, используя макросы, определённые в *sys/wait.h*.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке.

Программа должна быть снабжена комментариями, отражающими ход ее работы, а также содержать название лабораторной работы и сведения об авторе программы.

## Лабораторная работа os\_07. Вариант № 7/2012

**Задача:** создать программу *localedate* на языке C, имеющую следующую форму запуска:

```
localedate file opt
```

где

*file* – имя файла, в который надо записать стандартный вывод команды *date*. *opt* – дополнительный параметр, передаваемый программе;

В качестве параметра *opt* выступает имя локальных установок времени, которые необходимо передать программе *date* в качестве переменной окружения `LC_TIME`, например `POSIX`, `uk_UA.KOI8-U`.

Программа должна запустить указанную команду с необходимыми параметрами с помощью системных вызовов *fork* и *exec*. Перед загрузкой указанной программы следует перенаправить ее стандартный ввод или вывод.

Перед вызовом *fork* программа должна вывести свои PID и PPID. Потомок перед любыми действиями должен вывести свои PID и PPID.

Родительский процесс после завершения потомка должен вывести информацию о том, как завершился потомок, используя макросы, определённые в `sys/wait.h`.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке.

Программа должна быть снабжена комментариями, отражающими ход ее работы, а также содержать название лабораторной работы и сведения об авторе программы.

## Лабораторная работа os\_07. Вариант № 8/2012

**Задача:** создать программу *orderfile* на языке C, имеющую следующую форму запуска:

```
orderfile file opt
```

где

*file* – имя файла, содержимое которого надо передать команде *sort* в качестве стандартного ввода. *opt* – дополнительный параметр, передаваемый программе;

В качестве параметра *opt* выступает какая-либо из опций команды *sort*, например *-u*, *-r*.

Программа должна запустить указанную команду с необходимыми параметрами с помощью системных вызовов *fork* и *exec*. Перед загрузкой указанной программы следует перенаправить ее стандартный ввод или вывод.

Перед вызовом *fork* программа должна вывести свои PID и PPID. Потомок перед любыми действиями должен вывести свои PID и PPID.

Родительский процесс после завершения потомка должен вывести информацию о том, как завершился потомок, используя макросы, определённые в *sys/wait.h*.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке.

Программа должна быть снабжена комментариями, отражающими ход ее работы, а также содержать название лабораторной работы и сведения об авторе программы.