

Задания к работе №2 по дисциплине “Операционные системы и системное программирование”.

Все задания реализуются на языке программирования C (стандарт C99 и выше).

Реализованные приложения не должны завершаться аварийно.

Все ошибки, связанные с операциями взаимодействия с файлами, должны быть обработаны; все открытые файлы должны быть закрыты.

1. Реализуйте консольное приложение для Unix-систем, в котором выводятся некоторые атрибуты текущего процесса: идентификаторы текущего и родительского процесса, идентификатор группы вызывающего процесса, реальный идентификатор владельца, реальный идентификатор группы владельца и эффективные идентификаторы владельца и группы владельца. Продемонстрируйте работу реализованного приложения.

2. 1. В теле функции main() напишите следующие инструкции:

```
int pid;
pid=fork();
printf(“%d\n”, pid);
return 0;
```

Объясните наблюдаемый вывод вашего приложения.

2. В теле функции main() напишите следующие инструкции:

```
fork();
printf(“Hi\n”);
fork();
printf(“Hi\n”);
return 0;
```

Объясните наблюдаемый вывод вашего приложения.

3. В теле функции main() напишите следующие инструкции:

```
if (fork() || fork()) fork();
printf("forked! %d", fork());
return 0;
```

Объясните наблюдаемый вывод вашего приложения.

4. В теле функции main() напишите следующие инструкции:

```
if (fork() && (!fork()))
{
    if (fork() || fork()) fork();
}
printf("52\n");
return 0;
```

Объясните наблюдаемый вывод вашего приложения.

3. Реализуйте приложение для поиска строки в наборе текстовых файлов, список путей к которым передаётся в другом текстовом файле, путь к которому является аргументом командной строки. Задачу поиска строки в файле выполняйте в отдельном процессе (для каждого файла должен быть запущен свой процесс). В случае успешного поиска необходимо вывести на консоль пути к файлам, в которых поисковая строка была обнаружена, и для каждого из них количество вхождений исходной строки в файл; в случае неудачного поиска (строка не найдена ни в одном из файлов) необходимо вывести на консоль сообщение об отсутствии исходной строки *str* и запустить контролируемую fork-бомбу, дерево процессов в которой идеально сбалансировано по высоте и состоящее из *strlen(str)* процессов.
4. (Таненбаум, Гл. 4, задание 46) Реализуйте собственную версию программы *ls*. Ваша версия должна принимать в качестве аргументов командной строки значения *recmin* и *recmax* (целые неотрицательные числа) и далее произвольное количество путей к каталогам. Для каждого пути к каталогу необходимо вывести в стандартный поток вывода информацию (имя файла, расширение файла, адрес на диске) обо всех файлах, содержащихся в этом каталоге и его подкаталогах с глубиной рекурсивного обхода в диапазоне [*recmin...recmax*] по одной строке на каждый файл. При выводе каждое поле должно форматироваться в соответствии с его типом. Для каждого файла указать в списке только первый дисковый адрес файла. Выводимая информация о каждом файле должна быть выведена строго единожды.