

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ»**

Факультет безопасности информационных технологий

**Дисциплина:
“Операционные системы”**

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Выполнил:

Студент гр. N3249

Шарифуллин Ильдан Айдарович



Проверил:

Савков С.В.

Санкт-Петербург
2022г.

Задание:

1. Написать программу forkbomb для Linux и Windows.
2. Составить график числа процессов в ОС.
3. Проанализировать, как ОС реагирует на forkbomb.

Ход работы:

1. Написать программу forkbomb для Linux и Windows.

Для обеих ОС я использовал самые простые программы запуска самих себя.

fork.bat:

```
%0|%0
```

f.c:

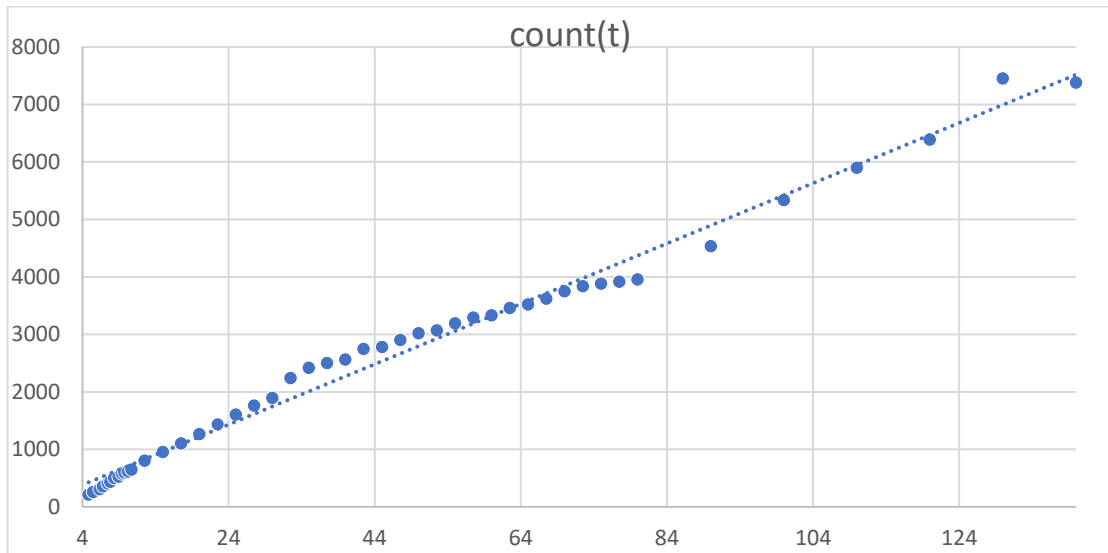
```
#include <unistd.h>
#include <time.h>
#include <stdio.h>
#include <errno.h>

int main(void) {
    int a;
    while(1) {
        a = fork();
        printf("errno value: %d.\n", errno);
        usleep(2800000);
    }
}
```

2. Составить график числа процессов в ОС.

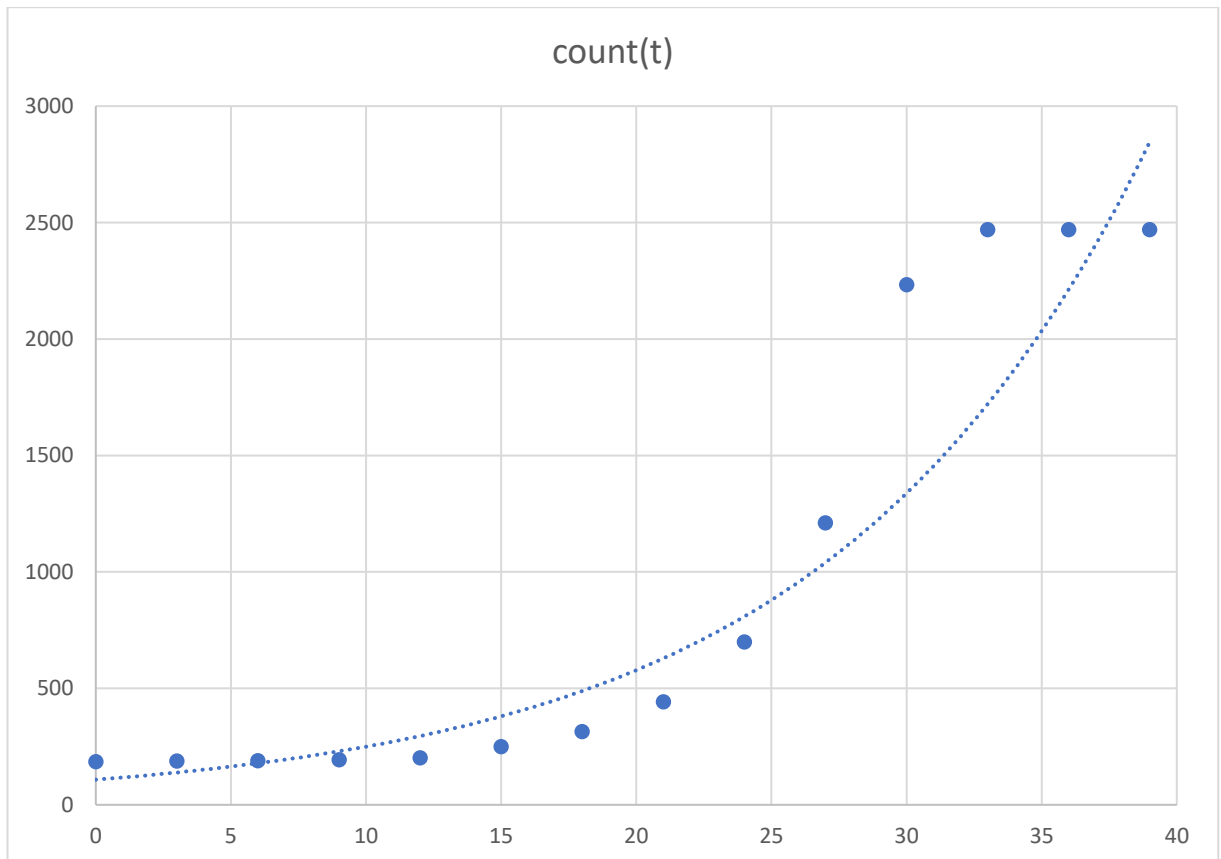
Для составления графиков я захватывал экран с помощью OBS studio, а потом по раскадровке видео выписывал в xls файл количество процессов (взятых из запущенного диспетчера задач) на определенных моментах запуска программы.

График для ОС Windows:



Из графика очевидно, что количество процессов растёт до $t=130$ с, далее начинает убывать или остаётся прежним.

График для ОС Linux:

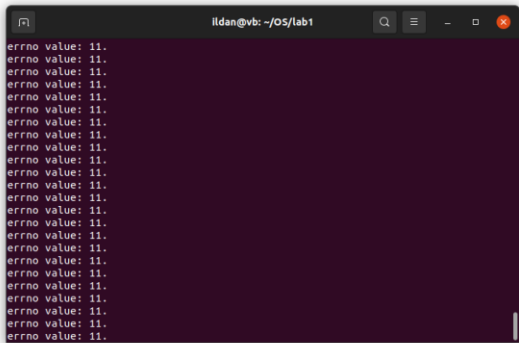
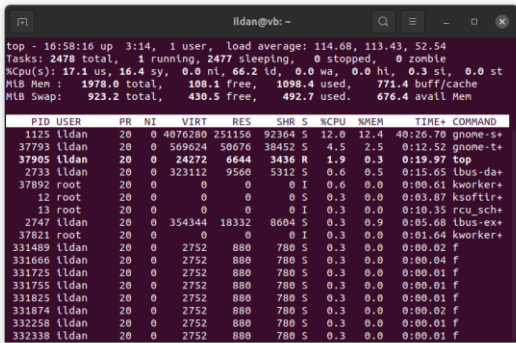


На данном графике заметно, что количество процессов возрастает по экспоненте до достижения своего максимума (2470 процессов), далее запуск новых процессов не происходит, а предыдущие не завершаются. До завершения программы так и стоят 2470 процессов.

3. Проанализировать, как ОС реагирует на forkbomb.

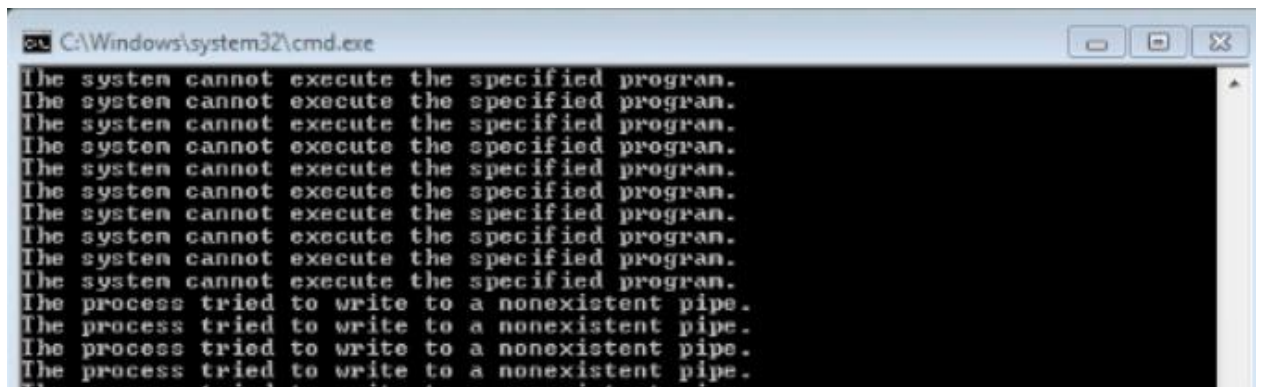
В случае с линуксом, ОС запустила максимально возможное количество процессов, а далее поддерживала количество процессов на таком уровне. При этом значение errno после запуска функции fork стало равным 11, что означает "fork failed: Resource temporarily unavailable". То есть запуск новых процессов не происходил из-за того, что у ОС не осталось ресурсов для запуска новых процессов.

```
1 #include <unistd.h>
2 #include <time.h>
3 #include <stdio.h>
4 #include <errno.h>
5
6 int main(void) {
7     int a;
8     while(1) {
9         a = fork();
10        printf("errno value: %d.\n", errno);
11        usleep(2000000);
12    }
13 }
```

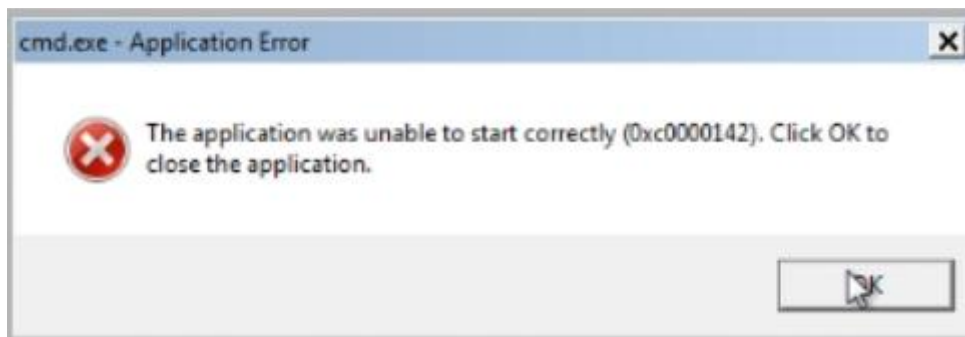


ОС позволила завершить работу программу, после чего все процессы тут же завершились. ОС вернулась к стандартному режиму работы.

В случае с windows, ОС постепенно запускала все больше и больше процессов, пока не заполнила ими все ресурсы. В это же время в консоль начали поступать следующие сообщения:



Также ОС выдавала ошибки:



После достижения максимума количество процессов начало колебаться в диапазоне ± 80 процессов. На данном этапе работать с ОС и даже завершить запущенную программу уже не представлялось возможным.

Вывод: завершить самописную форкбомбу на линуксе не составляет труда, в отличие от Windows, в котором нужно заранее иметь триггеры для “обезвреживания” бомбы. При этом, разные ОС выводили ошибки о том, что запуск новых процессов невозможен из-за недостатка ресурсов.

Приложение:

