

Домашнее задание №3. Мусаев Умахан Рашидович. БПИ234.

Условие:

До 8 баллов Разработать программу в которой родительский процесс вычисляет число Фибоначчи, а процесс-ребенок вычисляет значение факториала. В качестве исходного значения используется аргумент из командной строки. Использовать беззнаковую 64-разрядную целочисленную арифметику. Не забыть зафиксировать возникновение переполнения.

Опционально +2 балла

Для всех процессов вывести дополнительную информацию об их потомках и родителях. Организовать в программе дополнительно запуск процесса, который по завершении вычислений и выводе результатов выводит информацию о содержимом текущего каталога.

Решение:

Работа выполнена вместе с опциональной частью.

Файл с кодом программы содержится в архиве с отчетом

Задание было выполнено в виртуальной машине Ubuntu. Вот некоторые методы:

Метод нахождения числа Фибоначчи (родительский процесс):

```
//Фибоначчи.
uint64_t fibonacci(uint64_t n) {
    uint64_t a = 0, b = 1, c;
    if (n == 0) return a;
    for (uint64_t i = 2; i <= n; i++) {
        c = a + b;
        if (c < b) { // Переполнение
            cerr << "Обнаружено переполнение при вычислении числа Фибоначчи." << endl;
            exit(EXIT_FAILURE);
        }
        a = b;
        b = c;
    }
    return b;
}
```

Метод нахождения факториала числа (процесс-ребенок):

```
//Факториал.
uint64_t factorial(uint64_t n) {
    uint64_t result = 1;
    for (uint64_t i = 1; i <= n; i++) {
        uint64_t temp = result * i;
        if (temp / i != result) { // Переполнение
            cerr << "Обнаружено переполнение при вычислении факториала." << endl;
            exit(EXIT_FAILURE);
        }
        result = temp;
    }
    return result;
}
```

Метод вывода текущего каталога:

```
//Вывод каталога.
void list_directory() {
    DIR *dir;
    struct dirent *ent;
    if ((dir = opendir(".")) != nullptr) {
        while ((ent = readdir(dir)) != nullptr) {
            cout << ent->d_name << '\n';
        }
        closedir(dir);
    } else {
        perror("Ошибка при открытии каталога");
        exit(EXIT_FAILURE);
    }
}
```

Примеры работы программы:

При вставке 5:

```
umahan@50cent:~$ ./compute 5
Родительский процесс (PID: 2439)
Число Фибоначчи для 5 равно 5
Дочерний процесс (PID: 2440, Родительский PID: 2439)
Факториал числа 5 равен 120
Процесс для вывода содержимого каталога (PID: 2441, Родительский PID: 2439)
.bash_history
.profile
.bashrc
use_function_3.sh
.cache
if_function_2.sh
.landscape
.bash_logout
fibonnachifactorial.cpp
compute
.
use_function_2.sh
.sudo_as_admin_successful
.motd_shown
while_function_2.sh
while_function.sh
compute.cpp
.local
..
use_function.sh
create_file.sh
if_function.sh
umahan@50cent:~$ |
```

На экран также выводится PID (идентификатор процесса).

(в дальнейшем не буду скринить каталог).

Ввод 21:

```
umahan@50cent:~$ ./compute 21
Родительский процесс (PID: 2452)
Число Фибоначчи для 21 равно 10946
Дочерний процесс (PID: 2453, Родительский PID: 2452)
Обнаружено переполнение при вычислении факториала.
Процесс для вывода содержимого каталога (PID: 2454, Родительский PID: 2452)
.bash_history
.profile
.bashrc
use_function_3.sh
```

как мы видим, произошло переполнение при вычислении факториала.

Ввод 10000:

```
umahan@50cent:~$ ./compute 100000  
Родительский процесс (PID: 2459)  
Обнаружено переполнение при вычислении числа Фибоначчи.  
Дочерний процесс (PID: 2460, Родительский PID: 485)  
Обнаружено переполнение при вычислении факториала.  
umahan@50cent:~$ |
```

При вводе большого числа, происходит переполнение в обоих процессах.