

Operating Systems. Homework 3

[Operating Systems](#)

Александр Васюков | БПИ235

Задание

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.

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

Программа на Github: <https://github.com/vasyukov1/HSE-FCS-SE-2-year/tree/main/Operating%20Systems/Homeworks/03>

Вычисление n-го числа Фибоначчи

Вычисление числа Фибоначчи. Проверка переполнения, если сумма двух чисел станет меньше одного из слагаемых.

```
uint64_t fibonacci(unsigned int n) {
    uint64_t a = 0, b = 1;
    for (unsigned int i = 2; i <= n; ++i) {
        b = a + b;
        if (b < a) {
            fprintf(stderr, "Переполнение при вычислении Fibonacci(%u)\n",
n);
            exit(EXIT_FAILURE);
        }
        a = b - a;
    }
    return a;
}
```

Вычисление факториала числа n

Подсчёт факториала через цикл. Проверка на переполнение: если число при умножении на i меньше предыдущего числа, то произошло переполнение.

```

uint64_t factorial(unsigned int n) {
    uint64_t fact = 1;
    for (unsigned int i = 1; i <= n; ++i) {
        uint64_t next_fact = fact * i;
        if (next_fact < fact) {
            fprintf(stderr, "Переполнение при вычислении Factorial(%u)\n",
n);
                exit(EXIT_FAILURE);
            }
        fact = next_fact;
    }
    return fact;
}

```

Вывод содержания каталога

2. Открытие директории и проверка на ошибки.
3. Чтение элементов.
4. Закрытие директории.

```

void print_directory() {
    DIR *dir = opendir(".");
    if (!dir) {
        perror("Ошибка при открытии каталога");
        exit(EXIT_FAILURE);
    }

    struct dirent *entry;
    printf("\nСодержимое текущего каталога:\n");
    while ((entry = readdir(dir)) != NULL) {
        printf("%s\n", entry->d_name);
    }

    closedir(dir);
}

```

Основная программа

5. Проверка, что дан аргумент.
6. Создание дочернего процесса для вычисления факториала и проверка на отсутствие ошибок при создании.
7. Создание дочернего процесса для вывода содержания каталога и проверка на отсутствие ошибок при создании.
8. Вычисление числа Фибоначчи.
9. Ожидание завершения всех процессов.

```

int main(int argc, char *argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Нужен 1 аргумент!");
        return EXIT_FAILURE;
    }

    unsigned int n = atoi(argv[1]);

    pid_t pid1, pid2;
    pid1 = fork();
    if (pid1 < 0) {
        perror("Ошибка при применении fork()");
        return EXIT_FAILURE;
    }

    if (pid1 == 0) {
        printf("Процесс Factorial: PID = %d, PPID = %d\n", getpid(),
getppid());
        uint64_t fact = factorial(n);
        printf("Factorial(%u) = %llu\n\n", n, fact);
        exit(EXIT_SUCCESS);
    }

    pid2 = fork();
    if (pid2 < 0) {
        perror("Ошибка при применении fork()");
        return EXIT_FAILURE;
    }

    if (pid2 == 0) {
        printf("Процесс Directory: PID = %d, PPID = %d\n", getpid(),
getppid());
        print_directory();
        exit(EXIT_SUCCESS);
    }

    printf("Процесс Fibonacci: PID = %d, PPID = %d\n", getpid(), getppid());
    uint64_t fib = fibonacci(n);
    printf("Fibonacci(%u) = %llu\n\n", n, fib);

    wait(NULL);
    wait(NULL);
    return 0;
}

```

Результат

```
● alexvasyukov@Alexanders-MacBook-Air Homeworks % gcc 03/hw3.c
⊗ alexvasyukov@Alexanders-MacBook-Air Homeworks % ./a.out
Нужен 1 аргумент!%
● alexvasyukov@Alexanders-MacBook-Air Homeworks % ./a.out 5
Процесс Factorial: PID = 64208, PPID = 64207
Factorial(5) = 120

Процесс Fibonacci: PID = 64207, PPID = 64108
Fibonacci(5) = 3

Процесс Directory: PID = 64209, PPID = 64207

Содержимое текущего каталога:
.
..
03
02
.DS_Store
a.out
hw_01.pdf
● alexvasyukov@Alexanders-MacBook-Air Homeworks % ./a.out 10
Процесс Fibonacci: PID = 64218, PPID = 64108
Процесс Factorial: PID = 64219, PPID = 64218
Fibonacci(10) = 34

Factorial(10) = 3628800

Процесс Directory: PID = 64220, PPID = 64218

Содержимое текущего каталога:
.
..
03
02
.DS_Store
a.out
hw_01.pdf
```

```
● alexvasyukov@Alexanders-MacBook-Air Homeworks % ./a.out 30
Процесс Factorial: PID = 64282, PPID = 64281
Процесс Fibonacci: PID = 64281, PPID = 64108
Переполнение при вычислении Factorial(30)
Fibonacci(30) = 514229
```

```
Процесс Directory: PID = 64283, PPID = 64281
```

```
Содержимое текущего каталога:
```

```
.
..
03
02
.DS_Store
a.out
hw_01.pdf
```

```
⊗ alexvasyukov@Alexanders-MacBook-Air Homeworks % ./a.out 100
Процесс Factorial: PID = 64296, PPID = 64295
Переполнение при вычислении Factorial(100)
Процесс Fibonacci: PID = 64295, PPID = 64108
Переполнение при вычислении Fibonacci(100)
Процесс Directory: PID = 64297, PPID = 64295
```

```
Содержимое текущего каталога:
```

```
.
..
03
02
.DS_Store
a.out
hw_01.pdf
```