

**Московский авиационный институт
(национальный исследовательский университет)**

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

Лабораторная работа №2 по курсу «Операционные системы»

Студент: Т.А.Габдуллин
Преподаватель: Е. С. Миронов
Группа: 80-206Б
Вариант: 1
Дата:
Оценка:
Подпись:

Лабораторная работа №2

Вариант: 3. Рекурсивное вычисление суммы последовательности от 1 до n, где каждый отдельный уровень рекурсии вычисляется в отдельном процессе.

2 Исходный код

```
#include <sys/wait.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>

#define P1_READ 0
#define P2_WRITE 1

int fd[2];

int sum(int n)
{
    pid_t pid;
    int buf, status, bufrear;

    if (n == 1) {
        return 1;
    }

    pid = fork();
    if (pid == 0) {
        buf = sum(n - 1) + n;
        close(fd[P1_READ]);
        if (write(fd[P2_WRITE], &buf, sizeof(buf)) == -1) {
            perror("write");
        }
        exit(0);
    } else if (pid < 0) {
        perror("fork");
    } else {
        if (waitpid(pid, &status, 0) == -1) {
            perror("waitpid");
        }
    }

    if (read(fd[P1_READ], &bufrear, sizeof(bufrear)) == -1) {
        perror("read");
    }
}
```

```
    return bufread;
}

int main(void)
{
    int n = 0;

    if (pipe(fd) == -1) {
        perror("pipe");
    }

    printf("Enter a sequence number: ");
    scanf("%d", &n);
    if (n < 0) {
        printf("Number must be > 0.\n");
    } else {
        printf("%d\n", sum(n));
    }

    return 0;
}
```

3 Консоль

```
timxag@KEKNOTE:~/Документы$ gcc -o o2 lab2.c
timxag@KEKNOTE:~/Документы$ ./o2
Enter a sequence number: 1
1
timxag@KEKNOTE:~/Документы$ ./o2
Enter a sequence number: 2
3
timxag@KEKNOTE:~/Документы$ ./o2
Enter a sequence number: 5
15
timxag@KEKNOTE:~/Документы$ ./o2
Enter a sequence number: 90
4095
```

4 Вывод

Мной была написана программа, выполняющая рекурсивный подсчет суммы последовательности от 1 до N, где N дается на ввод. Программа оказалась не слишком большой и сложной, но дала базовые представления о работе с процессами и взаимодействии их между собой в операционной системе Unix.