

**Министерство образования и науки Российской Федерации**

**Федеральное агентство по образованию**

**Государственное образовательное учреждение высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Робототехника и комплексная автоматизация» (РК)**

**Кафедра «Системы автоматизированного проектирования» (РК6)**

****

**Отчет по лабораторной работе №4 по курсу**

**«Операционные системы»**

**Студент:** Петраков Станислав

**Группа:** РК6-56Б

**Преподаватель:** Грошев С.

Проверил:

Дата:

2021 год

**10 вариант**

**Задание:**

Модифицировать исходную программу так, чтобы происходило случайное перемещение символьной метки из центра экрана по 2-м ортогональным направлениям 4-мя процессами при ограничении на максимальное число передвижений. Когда процесс сделал предельное число передвижений – он завершается. Допустимое число перемещений передается в качестве аргумента командной строки.

**Исходный код:**

**#include <unistd.h>**

**#include <stdio.h>**

**#include <sys/timeb.h>**

**#include <stdlib.h>**

**#define WIDTH 81**

**#define HEIGHT 41**

**/\* clear standart text mode window \*/**

**void clrscr()**

**{**

**unsigned char esc[11];**

**/\* obtain set cursor to left-top window corner \*/**

**esc[0] = 27;**

**esc[1] = '[';**

**esc[2] = 'H';**

**write(1, esc, 3);**

**/\* obtain clear escape sequence \*/**

**esc[2] = '2';**

**esc[3] = 'J';**

**write(1, esc, 4);**

**return;**

**}**

**/\* position cursor in standart text window \*/**

**void gotoxy(int tx, int ty, char c)**

**{**

**unsigned char esc[16];**

**static unsigned char ystr[3]; /\* vertical cursor location \*/**

**static unsigned char xstr[3]; /\* horizontal cursor location \*/**

**int i; /\* ESC-sequence current index \*/**

**int j; /\* cursor location current index \*/**

**/\* convert cursor location to text format \*/**

**if ((tx > 99) || (ty > 99))**

**tx = ty = 99;**

**if ((tx < 1) || (ty < 1))**

**tx = ty = 1;**

**xstr[0] = xstr[1] = xstr[2] = '\0';**

**ystr[0] = ystr[1] = ystr[2] = '\0';**

**sprintf((char \*)xstr, "%d", tx);**

**sprintf((char \*)ystr, "%d", ty);**

**/\* obtain goto escape sequence \*/**

**esc[0] = 27;**

**esc[1] = '[';**

**i = 2;**

**j = 0;**

**while (ystr[j])**

**esc[i++] = ystr[j++];**

**j = 0;**

**esc[i++] = ';';**

**while (xstr[j])**

**esc[i++] = xstr[j++];**

**esc[i++] = 'H';**

**esc[i++] = '\b';**

**esc[i++] = ' ';**

**esc[i++] = c;**

**esc[i] = '\0';**

**write(1, esc, i);**

**return;**

**}**

**int main(int argc, char \*argv[])**

**{**

**if (argc < 1) {**

**printf("Usage: enter processes amount and distance\n");**

**exit(-1);**

**}**

**int x = 1;**

**int status;**

**int i;**

**int j = 0;**

**int PROCNUM = 4;**

**int \*pid = sbrk((PROCNUM + 1) \* sizeof(int));**

**char \*lead = sbrk((PROCNUM + 1) \* sizeof(char));**

**int dist = atoi(argv[1]);**

**int p;**

**char bell = '\007';**

**struct timeb tp[1];**

**int jump;**

**int live[PROCNUM];**

**for (int k = 0; k < PROCNUM; k++)**

**live[k] = 0;**

**ftime(tp);**

**srand(tp[0].millitm);**

**clrscr();**

**while (j < PROCNUM)**

**{**

**int seed = rand();**

**if ((pid[j] = fork()) == 0)**

**{**

**srand(seed);**

**usleep(PROCNUM - j);**

**while (live[j] < dist)**

**{**

**ftime(tp);**

**if ((tp[0].millitm % (j + 'A')) != j)**

**continue;**

**switch (j) {**

**case 0:**

**gotoxy(WIDTH/2 + 1, rand() % (HEIGHT/2) + 1, 'A' + j);**

**break;**

**case 1:**

**gotoxy(WIDTH/2 + 1, rand() % (HEIGHT/2) + HEIGHT/2 + 2, 'A' + j);**

**break;**

**case 2:**

**gotoxy(rand() % (WIDTH/2) + 1, HEIGHT/2 + 1, 'A' + j);**

**break;**

**case 3:**

**gotoxy(rand() % (WIDTH/2) + WIDTH/2 + 2, HEIGHT/2 + 1, 'A' + j);**

**break;**

**}**

**live[j]++;**

**for (i = 0; i < 1000000; i++) ;**

**}**

**exit('A' + j);**

**}**

**j++;**

**}**

**j = 0;**

**while ((p = wait(&status)) != (-1))**

**{**

**for (i = 0; i < PROCNUM; i++)**

**if (pid[i] == p)**

**lead[j++] = (char)((status >> 8) & '\377');**

**write(1, &bell, 1);**

**}**

**lead[j] = '\n';**

**sleep(1);**

**gotoxy(1, HEIGHT + 5, '\n');**

**write(1, lead, PROCNUM + 1);**

**exit(0);**

**}**