**3. Write a C program to create ‘n’ child processes. When all ‘n’ child processes terminates, Display total cumulative time children spent in user and kernel mode.**

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <unistd.h>

#include <sys/time.h>

#include <sys/resource.h>

#define NUM\_CHILDREN 5

int main() {

pid\_t pid;

int status;

int num\_children = NUM\_CHILDREN;

struct rusage usage;

struct timeval start, end, user\_time, kernel\_time;

double total\_user\_time = 0.0, total\_kernel\_time = 0.0;

gettimeofday(&start, NULL); // Start time

for (int i = 0; i < num\_children; ++i) {

pid = fork();

if (pid < 0) {

perror("fork");

exit(EXIT\_FAILURE);

} else if (pid == 0) {

// Child process

printf("Child %d with PID %d is running.\n", i + 1, getpid());

sleep(2); // Simulate some work in the child process

exit(EXIT\_SUCCESS);

}

}

// Parent process

while ((pid = wait(&status)) > 0) {

if (getrusage(RUSAGE\_CHILDREN, &usage) != -1) {

total\_user\_time += (double)usage.ru\_utime.tv\_sec + (double)usage.ru\_utime.tv\_usec / 1000000;

total\_kernel\_time += (double)usage.ru\_stime.tv\_sec + (double)usage.ru\_stime.tv\_usec / 1000000;

}

}

gettimeofday(&end, NULL); // End time

// Calculate the total elapsed time

double elapsed\_time = (end.tv\_sec - start.tv\_sec) + ((end.tv\_usec - start.tv\_usec) / 1000000.0);

printf("\nTotal cumulative time spent by children in user mode: %.6f seconds\n", total\_user\_time);

printf("Total cumulative time spent by children in kernel mode: %.6f seconds\n", total\_kernel\_time);

printf("Total elapsed time: %.6f seconds\n", elapsed\_time);

return 0;

}