Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра ПМиК

Расчетно-графическое задание

по дисциплине

«Операционные системы реального времени»

Выполнил: студент 4 курса

ИВТ, гр. ИП-013

Копытина Т.А.

Проверил: старший преподаватель кафедры ПМиК

Милешко Антон Владимирович

Новосибирск 2023г

Оглавление

Задание

1. Основные характеристики системы. Сравните время запуска (создания) нити и время активизации с помощью импульса заранее созданной нити.
2. Особенности реализации операционной системы и аппаратуры. Определите среднюю неточность задержки и диапазон изменения неточности при использовании функции delay().

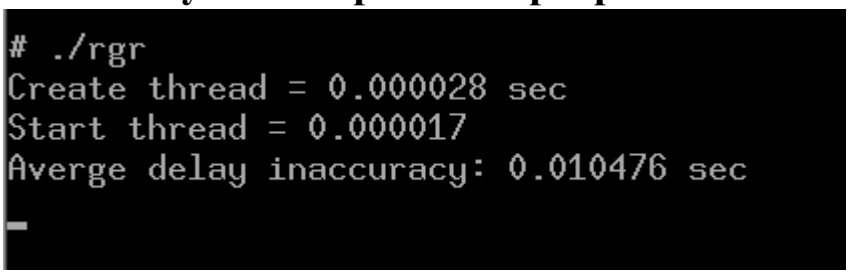
Выполнение работы

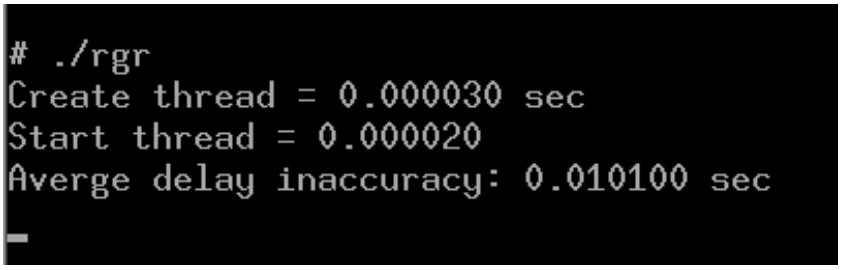
1. Для выполнения первого задания были использованы функции: pthread_create, для отслеживания времени был выбран ClockCycles, MsgReceivePulse, MsgSendPulse, pthread_join(), delay().
    - Функция pthread_create(pthread_t* thread, const pthread_attr_t* attr, void* (*start_routine)(void*), void* arg) создаёт нить на которой выполняется указанная при создании функция.
    - int ClockCycles() – функция из библиотеки sys/neutrino.h, возвращает текущее значение 64-битного счетчика циклов процессора. Чтобы вычислить количество прошедших секунд необходимо сначала вычислить количество циклов в секунду у системы с помощью SYSPAGE_ENTRY(qtime)->cycles_per_sec и уже поделить наши циклы на это количество.
    - int pthread_join(pthread_t thread, void **retval) – Функция pthread_join() ждёт завершения нити, указанной в thread. Если нить уже завершила работу, то pthread_join() завершается сразу. Нить, задаваемая в thread, должна позволять присоединение. Если retval не равно NULL, то pthread_join() копирует код выхода нити назначения (т. е., значение, которое нить назначения передала через pthread_exit(3)) в расположение по указателю retval. Если нить назначения была отменена, то в расположение по указателю retval помещается значение PTHREAD_CANCELED.
    - Функция int MsgReceivePulse(int chid, void *pulse, int bytes, struct _msg_info *info) – ждет поступления импульса в канал, имеющий идентификатор chid, и сохраняет полученные данные в буфере, на который указывает pulse.
    - Функция int MsgSendPulse(int coid, int priority, int code, int value) – посылает короткое неблокирующее сообщение процессу через канал, с которым установлено соединение coid.
2. Для второго задания была использована функция delay();

- Функция unsigned int delay (unsigned int duration) – приостанавливает вызывающий поток на duration миллисекунд.

В программе замер средней неточности функции delay() был осуществлен с помощью замера времени с помощью uint64_t start1 = ClockCycles() и uint64_t end1 = ClockCycles() .

Результаты работы программы





Листинг программы

```
#include <pthread.h>
#include <sys/neutrino.h>
#include <sys/iomsg.h>
#include <sys/iofunc.h>
#include <stdio.h>
#include <time.h>
#include <sys/dispatch.h>
#include <sys/syspage.h>
#include <inttypes.h>

#define MY_PULSE_CODE _PULSE_CODE_MINAVAIL + 2
uint64_t start, stop;
void* threadFunction(void* arg)
{
  name_attach_t *attach;
  struct _pulse pulse;
  int rcvid; //идентификатор ранее принятого сообщения
  int val;

  attach = name_attach(NULL, "rgr", 0);
  rcvid = MsgReceivePulse(attach->chid, &pulse, sizeof(pulse), NULL);
  stop = ClockCycles();
  printf("Start thread = %f\n", (double) (stop - start)
                                / SYSPAGE_ENTRY(qtime)->cycles_per_sec); // Обработчик сигнала

  return NULL;
}
```

```
int main()
{
  pthread_t thread;
  int64_t threadCreationTime, pulseActivationTime;

  start = ClockCycles();
  pthread_create(&thread, NULL, threadFunction, NULL);

  threadCreationTime = ClockCycles();

  printf("Create thread = %f sec\n", (double)(threadCreationTime - start) /
                                          SYSPAGE_ENTRY(qtime)->cycles_per_sec);

  int fd = name_open("rgr",0);

    start = ClockCycles();
  MsgSendPulse(fd, 10, MY_PULSE_CODE, pulseActivationTime);

  uint64_t start1 = ClockCycles();
  delay(10);
  uint64_t end1 = ClockCycles();
  printf("Averge delay inaccuracy: %f sec\n", (double)(end1-start1)/
                                          SYSPAGE_ENTRY(qtime)->cycles_per_sec);

  getchar();
  return 0;
}
```

Вывод

В рамках данного курса я узнала о операционных системах реального времени и научилась работать в одной из них, QNX, узнала о принципе системы, чем она отличается от других систем.

В процессе выполнения расчетно-графического задания я углубила свои теоретические знания об ОСРВ, в частности о QNX. Больше узнала об алгоритме планирования и распределения ресурсов системы. Узнала о том, как измерить время работы части программы в QNX.

А также в ходе расчетно-графической работы я доказала, что активизация нити при помощи импульса на 1/3 времени меньше, чем ее запуск.