Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Реферат

По дисциплине «Основы алгоритмизации и программирования»

На тему «Функции работы с датой и временем»

 Выполнила:

Студентка1 курса 6 группы

Альшевская Алина Михайловна

Преподаватель: доц Белодед Н.И

2023, Минск

Функции работы с датой и временем

Существует четыре типа времени: clock\_t, time\_t, size\_t и tm . Типы - clock\_t, size\_t и time\_t могут представлять системное время и дату как некоторое целое число.

Тип структуры tm содержит дату и время в виде структуры C, имеющей следующие элементы:

struct tm {

int tm\_sec; // секунды

int tm\_min; // минуты

int tm\_hour; // часы

int tm\_mday; // день месяца

int tm\_mon; // месяц

int tm\_year; // год

int tm\_wday; // день недели

int tm\_yday; // день года

};

Ниже приведены важные функции, которые мы используем при работе с датой и временем на C или C ++.

|  |  |  |
| --- | --- | --- |
| **time\_t time (time\_t \* time);** | Это возвращает текущее время календаря системы в количестве секунд, прошедших с 1 января 1970 года. | #include <ctime>  #include <iostream>  int main() {  setlocale(LC\_ALL, "russian");  time\_t now = time(0); // получаем текущее время  std::cout << "Текущее время в секундах с начала эпохи Unix: " << now << std::endl;  return 0;  } |
| **char \* ctime (const time\_t \* time);** | Это возвращает указатель на строку формы день месяц год: минуты: секунды год \ n \ 0 | #include <ctime>  #include <iostream>  int main() {  setlocale(LC\_ALL, "russian");  time\_t now = time(0); // получаем текущее время  char\* dt = ctime(&now); // преобразуем время в строку  std::cout << "Текущая дата и время: " << dt << std::endl;  return 0;  } |
| **struct tm \* localtime (const time\_t \* time);** | Это возвращает указатель на структуру **tm,**  представляющую локальное время | #include <ctime>  #include <iostream>  int main() {  setlocale(LC\_ALL, "russian");  time\_t now = time(0); // получаем текущее время  struct tm\* ltm = localtime(&now); // преобразуем время в структуру tm  // выводим текущую дату и время  std::cout << "Год: " << 1900 + ltm->tm\_year << std::endl;  std::cout << "Месяц: " << 1 + ltm->tm\_mon << std::endl;  std::cout << "День: " << ltm->tm\_mday << std::endl;  std::cout << "Время: " << 1 + ltm->tm\_hour << ":";  std::cout << 1 + ltm->tm\_min << ":";  std::cout << 1 + ltm->tm\_sec << std::endl;  return 0;  } |
| **double difftime (time\_t time2, time\_t time1)** | Эта функция вычисляет разницу в секундах между временем 1 и временем 2. | #include <ctime>  #include <iostream>  int main() {  setlocale(LC\_ALL, "russian");  time\_t start\_time = time(0);// получаем текущее время  for (double i = 0; i < 999999999; i++) {} // выполняем какую-то задачу  time\_t end\_time = time(0);// получаем время после выполнения задачи  double diff = difftime(end\_time, start\_time); // вычисляем разницу между конечным и начальным временем  std::cout << "Время выполнения задачи: " << diff << " секунд" << std::endl;  return 0;  } |
| **size\_t strftime ();** | Эта функция может использоваться для форматирования даты и времени в определенном формате. | #include <ctime>  #include <iostream>  int main() {  setlocale(LC\_ALL, "russian");  time\_t now = time(0); // получаем текущее время  struct tm\* ltm = localtime(&now); // преобразуем время в структуру tm  char str[80];  strftime(str, sizeof(str), "Сегодня %A, %d %B %Y года. Текущее время: %H:%M:%S.", ltm);  std::cout << str << std::endl;  return 0;  } |
| **clock\_t clock (void);** | Это возвращает значение, которое приблизительно соответствует времени, в течение которого вызываемая программа была запущена | #include <ctime>  #include <iostream>  int main() {  setlocale(LC\_ALL, "russian");  clock\_t start\_time = clock();// начинаем отсчет времени  for (double i = 0; i < 99999999; i++) {}// выполняем какую-то задачу  clock\_t end\_time = clock();// получаем время после выполнения задачи  double execution\_time = static\_cast<double>(end\_time - start\_time) / CLOCKS\_PER\_SEC;// вычисляем время выполнения задачи  std::cout << "Время выполнения задачи: " << execution\_time << " секунд" << std::endl;  return 0;  } |

Функции работы с датой и временем в С++ позволяют управлять и манипулировать датами и временем, что является важной частью многих приложений. Они представляют гибкий и эффективный способ работы с датой и временем.