СТАНДАРТНАЯ БИБЛИОТЕКА СИ

Тема 4. Раздел 1. Assert, ctype, time.

План лекции

- Assert.h
- Ctype.h
- Time.h

assert.h

Процедура диагностики

assert.h

void assert(int expression);

- Макрос assert() добавляет к программе процедуру диагностики. После выполнения, если выражение ложно (то есть, результат сравнения 0), assert() пишет информацию о вызове в поток stderr и вызывает функцию abort(). Информация, которая пишется в stderr включает в себя:
  - текст выражения, значение которого равно нулю 0
  - имя файла с исходным кодом (предопределённый макрос __FILE__)
  - строка у файла с исходным кодом (предопределённый макрос __LINE__)

```
int main ()
{
 FILE *fd;
 fd = fopen ("/home/user/file.txt", "r");
 assert (fd);
 fclose (fd);
 return 0;
 }
```

ctype.h

Классификация и преобразование отдельных символов.

ctype.h

int isdigit (int c);

Имя функции	Проверяет, является ли аргумент…
isalnum	…буквой или цифрой
isalpha	…буквой
iscntrl	…управляющим символом
isdigit	…цифрой
isgraph	…символом, имеющим графическое представление
islower	…буквой в нижнем регистре
isprint	…символом, который может быть напечатан
ispunct	…символом, имеющим графическое представление, но не являющимся при этом буквой или цифрой
isspace	…разделительным символом
isupper	…буквой в верхнем регистре
isxdigit	…цифрой шестнадцатеричной системы счисления

ctype.h

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main ()
{
 char str[]="1776ad";
 int year;
 if (isdigit(str[0]))
 {
 year = atoi (str);
 printf ("The year that followed %d was %d.\n",year,year+1);
 }
 return 0;
 }
```

ctype.h

Имя функции	Описание
tolower	Преобразует аргумент в его строчный аналог (нижний регистр), если это возможно; иначе возвращается неизмененный аргумент.
toupper	Преобразует аргумент в его прописной аналог (верхний регистр), если это возможно; иначе возвращается неизмененный аргумент

```
int main ()
{
 int i=0;
 char str[]="Test String.\n";
 char c;
 while (str[i])
 {
 c=str[i];
 putchar (tolower(c));
 i++;
 }
 return 0;
}
```

time.h

Типы и функции для работы с датой и временем.

time.h

- ***clock_t***
- ***time_t***
- ***CLOCKS_PER_SEC***
- ***struct tm:***

Имя	Описание
int tm_sec;	Секунды от начала минуты(0,59)
int tm_min;	Минуты от начала часа(0,59)
int tm_hour;	Часы от полуночи(0,23)
int tm_mday;	Число месяца(1,31)
int tm_mon;	Месяцы после января(0,11)
int tm_year;	Годы с 1900
int tm_wday;	Дни с воскресенья(0,6)
int tm_yday;	Дни с первого января(0,365)
int tm_isdst;	Признак летнего времени

time.h

clock_t clock(void);

- Возвращает время, измеряемое процессором в тактах от начала выполнения программы, или −1, если оно не известно. Пересчет этого времени в секунды выполняется по формуле clock () / CLOCKS_PER_SEC

time_t time(time_t *timer)

- Возвращает текущее календарное время или −1, если это время не известно. Если указатель tp не равен NULL, то возвращаемое значение записывается также и в *tp.

double difftime(time_t timer1, time_t timer0)

- Возвращает разность time2-time1, выраженную в секундах.

char *asctime(const struct tm *tp)

- Преобразует время из структуры *tp в строку вида "Sun Jan 3 15:14:13 1988\n\0"

time.h

char *ctime(const time_t *timer)

- Преообразует время time_t в C-строку Формат строки Www Mmm dd hh:mm:ss yyyy

struct tm *gmtime(const time_t *timer)

- Преобразует time_t в структуру struct tm

struct tm *localtime(const time_t *timer)

- Заполняет структуру struct tm локальным временем

time.h

time_t mktime(struct tm *timeptr)

- Преобразует местное время, заданное структурой *tp, в календарное и возвращает его в том же виде, что и функция time(). Компоненты структуры будут иметь значения в указанных выше диапазонах. Функция возвращает календарное время или −1, если оно не представимо.

```
int main( void )
{
 struct tm when;
 time_t now, result;
 int days = 20;
 time( &now );
 localtime( &when);
 printf( "Current time is %s\n", asctime(&when ));
 when.tm_mday = when.tm_mday + days;
 if( (result = mktime( &when )) != (time_t)-1 )
            printf( "In %d days the time will be %s\n", days, asctime(&when ) );
 else perror( "mktime failed" );
}
```

time.h

size_t strftime(char *strDest, size_t maxsize, const char *format, const struct tm *timeptr)

- Выводит в строку strDest максимальной длинны maxsize времыя записанное в timeptr в определенном формате format

```
int main ()
{
 time_t rawtime;
 struct tm * timeinfo;
 char buffer [80];
 time ( &rawtime );
 timeinfo = localtime ( &rawtime );
 strftime (buffer,80,"Now it's %I:%M%p.",timeinfo);
 puts (buffer);
 return 0;
}
```

%a	Abbreviated weekday name *	Thu
%A	Full weekday name *	Thursday
%b	Abbreviated month name *	Aug
%B	Full month name *	August
%c	Date and time representation *	Thu Aug 23 14:55:02 2001
%d	Day of the month (01-31)	23
%H	Hour in 24h format (00-23)	14
%I	Hour in 12h format (01-12)	02
%j	Day of the year (001-366)	235
%m	Month as a decimal number (01-12)	08
%M	Minute (00-59)	55
%p	AM or PM designation	PM
%S	Second (00-61)	02
%U	Week number with the first Sunday as the first day of week one (00-53)	33
%w	Weekday as a decimal number with Sunday as 0 (0-6)	4
%W	Week number with the first Monday as the first day of week one (00-53)	34
%x	Date representation *	08/23/01
%X	Time representation *	14:55:02
%y	Year, last two digits (00-99)	01
%Y	Year	2001
%Z	Timezone name or abbreviation	CDT
%%	A % sign	%

Практика

- Реализовать функцию паузы программы на произвольное количество секунд использую функцию clock ()
- Вычислить сколько времени занимает чтение БД «Записной книжки» из файла
- Выводить на экран время запуска программы «Записная книжка»
- Хранить для каждой персоны дату ее рождения и выводить на экран. Напоминать о ДР