1. 嵌入式Linux系统中，有哪些时间概念

UTC：世界标准时间，引用的是GMT（格林威治时间）

日历时间：从1970年1月1日0点开始

1. 编写一个程序，显示当地时间、标准时间，时间显示要求每秒更新一次，把时间显示到屏幕上

#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<time.h>

#define TRUE 1

int main(void)

{

time\_t t = 0;

struct tm \*LocalTime = NULL, \*UTime = NULL;

while (TRUE)

{

system("clear");

time(&t);

//get local time

LocalTime = localtime(&t);

printf("LocalTime: %s", asctime(LocalTime));

//get GMT

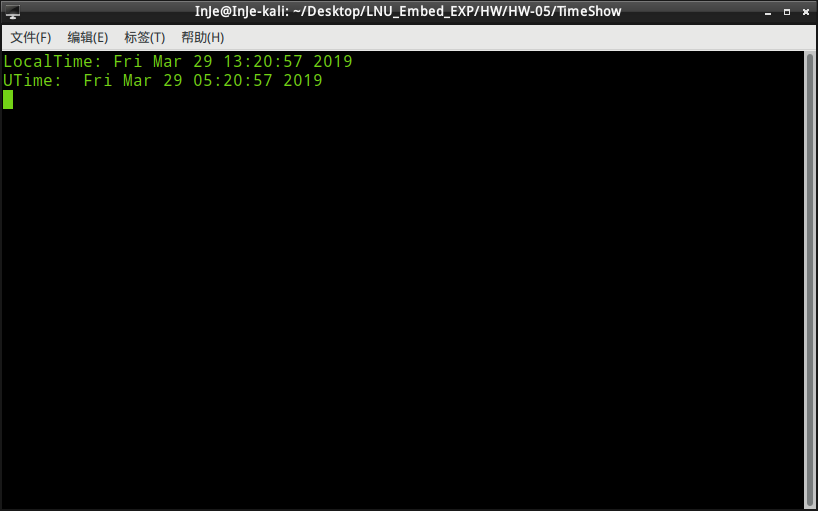
UTime = gmtime(&t);

printf("UTime:\t%s", asctime(UTime));

sleep(1);

}

return 0;

}

1. 时间差相关的函数有什么作用，举例说明

计算两个时间的秒钟数时间差

eg.

double difftime(time\_t time1, time\_t time2);//计算time1与time2之间的时间相差的秒数

1. 嵌入式Linux如何实现延迟功能

使用sleep函数或usleep函数，sleep函数实现精确到秒级的时间延迟，usleep实现精确到微妙级的时间延迟，使用前先引用UNIX标准头文件#include<unistd.h>

1. 编写一个程序，用你的计算机计算PI的值，如果程序运行1分钟，可以计算出的PI的精度是小数点后的多少位

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<time.h>

#define TRUE 1

#define MAX\_ACCU 500

int main(void)

{

int R[MAX\_ACCU] = { 2 };//R(1) = 2.00...

int Sum[MAX\_ACCU] = { 2 };//Sum(1) = 2.00...

long i, j, t, d;

int plus;

struct timeval tv1, tv2;

gettimeofday(&tv1, NULL);//get start time

//Calc

for (i = 1; i < MAX\_ACCU; i++)

{

//R(n)\*(n)

plus = 0;

for (j = MAX\_ACCU - 1; j >= 0; j--)

{

t = R[j] \* i + plus;

R[j] = t % 10;

plus = t / 10;

}

//R(n+1) = R(n)\*n / (2\*n+1)

d = 2 \* i + 1;

plus = 0;

for (j = 0; j < MAX\_ACCU; j++)

{

t = R[j] + plus \* 10;

R[j] = t / d;

plus = t % d;

}

//Sum(n+1) = Sum(n) + R(n+1)

plus = 0;

for (j = MAX\_ACCU - 1; j >= 0; j--)

{

t = Sum[j] + R[j] + plus;

Sum[j] = t % 10;

plus = t / 10;

}

}

gettimeofday(&tv2, NULL);//get stop time

//Output

printf("time: %d us\n", tv2.tv\_sec \* 1000000 + tv2.tv\_usec - tv1.tv\_sec \* 1000000 - tv1.tv\_usec);

printf("%d.", Sum[0]);

for (i = 1; i < MAX\_ACCU; i++)

{

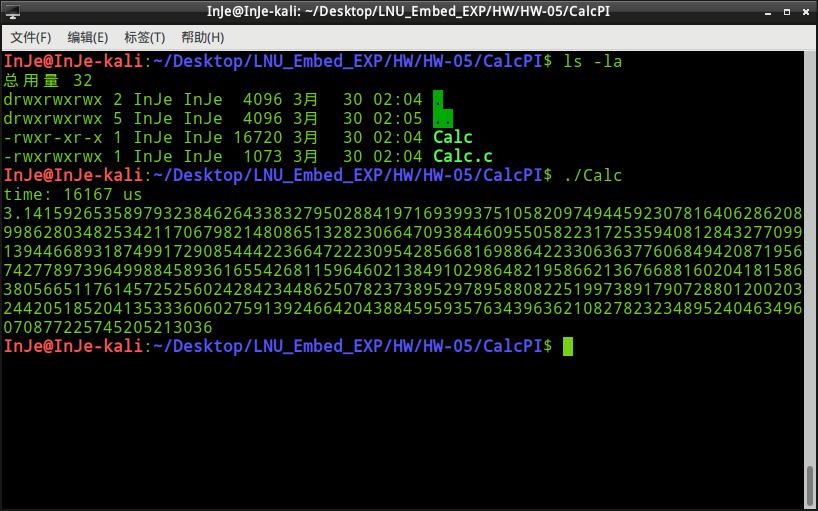
printf("%d", Sum[i]);

}

printf("\n");

return 0;

}



1. 设计程序测试usleep函数是否精确

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<time.h>

#define DelayTime 10000

int main(void)

{

struct timeval tv1, tv2;

int Delta;

gettimeofday(&tv1, NULL);

usleep(DelayTime);

gettimeofday(&tv2, NULL);

Delta = tv2.tv\_sec \* 1000000 + tv2.tv\_usec - tv1.tv\_sec \* 1000000 - tv1.tv\_usec - DelayTime;

printf("DelayTime = %d\nDelta = %ld\nDelta(Percent) = %.2f\n", DelayTime, Delta, (Delta \* 100.0) / DelayTime);

return 0;

}

