# HarmonyOS编译框架介绍

# 前言

**本节主要介绍：**

- ninja编译工具
- 如何编译模块
- 代码运行逻辑

# 目录

# Ninja编译工具简介

## 什么是Ninja

在Unix/Linux下通常使用Makefile来控制代码的编译，但是Makefile对于比较大的项目有时候会比较慢，代码在编译都变成了程序员放松的借口了。所以这个Google的程序员在开发Chrome的时候因为忍受不了Makefile的速度，自己重新开发出来一套新的控制编译的工具叫作Ninja，Ninja相对于Makefile这套工具更注重于编译速度。除了Chrome现在还有一些其他的比较大的项目也在开始使用Ninja。

# 如何编译模块

## 模块gn文件

```
import("//build/lite/config/component/lite_component.gni")

lite_component("app") {
    features = [
        #"A1_kernal_thread:thread_example",
        #"A2_kernel_timer:timer_example",
        #"A3_kernel_event:event_example",
        #"A4_kernel_mutex:mutex_example",
        #"A5_kernel_semaphore:semaphore_example",
        #"A6_kernel_message:message_example",

        #"B1_basic_led_blink:led_example",
        #"B2_basic_button:button_example",
        #"B3_basic_pwm_led:pwm_example",
        #"B4_basic_adc:adc_example",
        "B5_basic_i2c_nfc:i2c_example",
        #"B6_basic_uart:uart_example",

        #"C1_e53_sf1_mq2:e53_sf1_example",
        #"C2_e53_ia1_temp_humi_pls:e53_ia1_example",
        #"C3_e53_sc1_pls:e53_sc1_example",
        #"C4_e53_sc2_axis:e53_sc2_example",
        #"C5_e53_is1_infrared:e53_is1_example",

        #"D1_iot_wifi_scan:wifi_scan",
        #"D2_iot_wifi_connect:wifi_connect",
        #"D3_iot_udp_client:udp_client",
        #"D4_iot_tcp_server:tcp_server",
        #"D5_iot_mqtt:iot_mqtt",
        #"D6_iot_cloud_oc:oc_mqtt",
        #"D7_iot_cloud_onenet:onenet_mqtt",


    ]
}
```

## 业务gn文件

```
static_library("thread_example") {
    sources = [
        "Thread_example.c"
    ]

    include_dirs = [
        "//utils/native/lite/include",
        "//kernel/liteos_m/components/cmsis/2.0",
        "//base/iot_hardware/interfaces/kits/wifiiot_lite",
    ]
}
```

## Json文件

```
"subsystem": [
  {
    "name": "applications",
    "component": [
      { "name": "BearPi-HM_nano", "dir": "//applications/BearPi/BearPi-HM_nano/sample:app", "features":[] }
    ]
  },
```

# 代码如何执行的

## 初始化入口

```
printf("\n app_main test\n");
MODULE_INIT(bsp);
MODULE_INIT(device);
MODULE_INIT(core);
SYS_INIT(service);
SYS_INIT(feature);
MODULE_INIT(run);
printf("\n app_main INIT_TEST_CALL\n");
```

## 注册入口

```
static void Thread_example(void)
{
    osThreadAttr_t attr;

    attr.name = "thread1";
    attr.attr_bits = 0U;
    attr.cb_mem = NULL;
    attr.cb_size = 0U;
    attr.stack_mem = NULL;
    attr.stack_size = 1024*4;
    attr.priority = 25;

    if (osThreadNew((osThreadFunc_t)thread1, NULL, &attr) == NULL) {
        printf("[task_entry] Falied to create thread1!\n");
    }

    attr.name = "thread2";

    if (osThreadNew((osThreadFunc_t)thread2, NULL, &attr) == NULL) {
        printf("[task_entry] Falied to create thread2!\n");
    }
}

APP_FEATURE_INIT(Thread_example);
```

# 本节小结

- 1、了解如何编译模块
- 2、掌握代码是如何执行的

# 谢 谢 观 看

开 源 从 小 熊 派 开 始

OPEN-SOURCE STARTED WITH THE BEARPI