

# loongarch rt-thread用户手册

龙芯乔崇

January 11, 2022

## Contents

1 编译命令	2
2 转Makefile方法	2
3 bsp/la1cdev/rtconfig.py里定义了gcc, ld参数	2
4 scons - menuconfig生成	2
5 .config里有CONFIG_前缀, rtconfig.h里没有	2
6 Sconstruct, SconScript里的depend里可以依赖这个	2
7 源码里可以ifdef依赖这个	2
8 链接脚本里可以指定开始地址	3
9 rtconfig.py ldflags里可以-Ttext可以覆盖开始地址	3
10 data段可以和text挨着放, 如分开, data用AT属性让数据放在代码后, 启动程序拷贝到内存里	3
11 目前代码设置的入口地址0x1c000000, 代码数据都放在0x1c000000开始内存	3
12 libcpu/loongarch/entry_gcc.S是elf文件的入口, text和data分开放要拷贝	3
13 libcpu/loongarch/common/exception_gcc.S, exception.c是异常处理函数	3
14 libcpu/loongarch/common/context_gcc.S, stack.c, stackframe.h 是进程相关函数	3
15 libcpu/loongarch/common/asm-offsets.h是由libcpu/loongarch/common/asm-offsets.c和bsp/la1cdev/Sconstruct自动生成	3
16 地址映射	3
16.1 如果没有cache可以只使用线性地址模式	3
16.2 如果有cache应该采用映射地址模式, 设置两个直接映射地址段和熟悉	3
16.3 启动采用线性地址模式	3
16.4 现在代码里启动后马上设置成映射地址模式, 模式转换时候需要注意当前执行pc地址要合法	3
17 中断	3
17.1 中断可以采用向量中断和非向量中断	3
17.2 现在大核软件都采用的向量中断	3
17.3 现在小核代码都采用的非向量中断	3

18 bsp/la1cdev/runqemu启动运行rtthread.bin	4
19 qemu模拟了小核串口和中断处理	4
20 qemu模拟内存0x1c000000开始64M	4
21 loongarch-linux-gdb rt-thread.elf	4
22 ./runqemu -S可以让qemu启动时候停住，通过loongarch-linux-gdb连接qemu调试代码执行过程	4
23 ./runqemu -d in_asm,cpu,exe -D /tmp/qemu.log可以把程序跑的指令和数据存到/tmp/qemu.log 编译	4

## 1 编译命令

```
cd bsp/la1cdev
scons --menuconfig
scons -c
scons --verbose
```

scons具体用法参考

```
man scons
```

和 rt-thread/tools/building.py,makefile.py

## 2 转Makefile方法

```
scons --target=makefile
```

Sconstruct, Sconscript和rtconfig.py

## 3 bsp/la1cdev/rtconfig.py里定义了gcc, ld参数

- 通过修改里面Glob()可以加入和删除某个文件
- 默认Glob把所有文件都加了，可以设置exclude变量去掉某些
- DefineGroup(depend = [ ' RT\_USING\_NEWLIB' , ' RT\_USING\_PTHREADS' ])可以设置depend让代码选了某些选项才编译

rtconfig.h和.config

## 4 scons - menuconfig生成

## 5 .config里有CONFIG\_前缀, rtconfig.h里没有

## 6 Sconstruct, SconScript里的depend里可以依赖这个

## 7 源码里可以ifdef依赖这个

链接脚本

## 8 链接脚本里可以指定开始地址

### 9 rtconfig.py ldflags里可以-Ttext可以覆盖开始地址

10 data段可以和text挨着放，如分开，data用AT属性让数据放在代码后，启动程序拷贝到内存里

11 目前代码设置的入口地址0x1c000000，代码数据都放在0x1c000000开始内存

源码

```
libcpu/loongarch/common
libcpu/loongarch/gsl32
bsp/la1cdev
```

12 libcpu/loongarch/entry\_gcc.S是elf文件的入口, text和data分开放要拷贝

13 libcpu/loongarch/common/exception\_gcc.S, exception.c是异常处理函数

14 libcpu/loongarch/common/context\_gcc.S, stack.c, stackframe.h是进程相关函数

15 libcpu/loongarch/common/asm-offsets.h是由libcpu/loongarch/common/asm-offsets.c和bsp/la1cdev/Sconstruct自动生成

loongarch相关

## 16 地址映射

16.1 如果没有cache可以只使用线性地址模式

16.2 如果有cache应该采用映射地址模式，设置两个直接映射地址段和熟悉

16.3 启动采用线性地址模式

16.4 现在代码里启动后马上设置成映射地址模式，模式转换时候需要注意当前执行pc地址要合法

## 17 中断

17.1 中断可以采用向量中断和非向量中断

17.2 现在大核软件都采用的向量中断

17.3 现在小核代码都采用的非向量中断

qemu相关

- 18 bsp/la1cdev/runqemu启动运行rtthread.bin
- 19 qemu模拟了小核串口和中断处理
- 20 qemu模拟内存0x1c000000开始64M
- 21 loongarch-linux-gdb rt-thread.elf

```
target remote 127.0.0.1:1234
display/i $pc
si
si
i r
b rt_kprintf
```

- 22 ./runqemu -S可以让qemu启动时候停住，通过loongarch-linux-gdb连接qemu调试代码执行过程
- 23 ./runqemu -d in\_asm,cpu,exe -D /tmp/qemu.log可以把程序跑的指令和数据存到/tmp/qemu.log