

在 Finsh 中运行自定义函数

RealTouch 评估板 RT-Thread 入门文档

版本号：1.0.0
日期：2012/8/17

修订记录

| 日期 | 作者 | 修订历史 |
|-----------|--------|------|
| 2012/8/17 | bloom5 | 创建文档 |

实验目的

快速熟悉了解命令行形式的调试和信息查看组件 finsh

硬件说明

本实验使用 RT-Thread 官方的 Realtouch 开发板作为实验平台。涉及到的硬件主要为

- ❑ 串口 3，作为 rt_kprintf 输出，需要连接 JTAG 扩展板
- 具体请参见《Realtouch 开发板使用手册》

实验原理及程序结构

对于用户来讲，finsh 组件有三个主要功能

- ❑ 获取系统运行时信息，如各种 RT-Thread 内核对象的动态信息。
- ❑ 能够对任意寄存器和内存地址进行读写操作
- ❑ 能够直接在 shell 中调用系统函数，访问系统变量

实验设计

本实验的主要设计目的是帮助读者了解 finsh 组件的基本用法及添加自定义函数。请读者注意，本实验本身不具有实际的工程参考价值，只是帮助读者快速了解 finsh 组件的用法。

源程序说明

系统依赖

在 rtconfig.h 中需要开启

- ❑ #define RT_USING_HEAP

此项可选，开启此项可以创建动态线程和动态信号量，如果使用静态线程和静态信号量，则此项不是必要的

- ❑ #define RT_USING_CONSOLE

此项必须，本实验使用 rt_kprintf 向串口打印按键信息，因此需要开启此项

- ❑ #define RT_USING_FINSH

此项必须，只有打开此项，finsh 组件才会被使用

❑ #define RT_USING_SYMTAB

此项可选，打开此项可以使用宏输出的方式向 finsh shell 中添加命令。

主程序说明

本次程序主要是关于相关宏的打开，以及如何建立可以运行的自设函数。在这里我们有三种方法，第一种是直接添加内置命令，另一种则是通过 FINSH_FUNCTION_EXPORT 宏，将函数增为 finsh shell 的命令，第三种是利用 finsh 的应用程序接口 finsh_syscall_append()。

首先我们来看一下第一种方式，

在 rt_config.h 中使能 finsh 组件

```
/* SECTION: finsh, a C-Express shell */
#define RT_USING_FINSH
/* Using symbol table */
#define FINSH_USING_SYMTAB
#define FINSH_USING_DESCRIPTION
```

事实上，FINSH_USING_SYMTAB 宏在第一种方法中可以不使能（编译时会报两个警告），但为了方便第二种方法的一块实现，我们将这个宏也打开了。

接着要做的是在 /finsh/cmd.c 中添加自定义函数，在此我们添加了一个简单的函数，它的功能仅仅是打印一条语句。

添加内置命令

```
long hello(void)
{
    rt_kprintf("Hello RT-Thread!\n");

    return 0;
}
FINSH_FUNCTION_EXPORT(hello, say hello world);
```

接下来让我们看一下第二种方法，通过在 application.c 文件中添加下述语句，在这里我们构建了一个带参数的函数 fun(int a)，当然不带参数也是可以的，不带参数的函数也是可以的。另外在 finsh shell 中调用带参数的命令别忘了加上合适的参数。

```
void fun_with_arg(int a)
{
    rt_kprintf("fun's arg is: %d\n", a);
}
FINSH_FUNCTION_EXPORT(fun_with_arg, funciton with a argument);
```

最后来看看第三种方法，在 application.c 中加入下面函数

```
void fun(void)
{
    rt_kprintf("fun is performed by finsh.\r\n");
}
```

并在初始化线程中添加 api 函数语句

```
void rt_init_thread_entry(void* parameter)
{
    rt_components_init();
    finsh_syscall_append("fun", (syscall_func)fun);
}
```

值得注意的是，第二和第三种方法，需要在 application.c 文件中包含头文件 finsh.h。

编译调试及观察输出信息

编译请参见《RT-Thread 配置开发环境指南》完成编译烧录，参考《Realtouch 开发板使用手册》完成硬件连接，连接扩展板上的串口和 jlink。

烧录程序以后可以看到如下信息：

方法一：添加内置命令

```
\ | /
- RT -   Thread Operating System
/ | \    1.1.0 build Aug 17 2012
2006 - 2012 Copyright by rt-thread team
finsh>>list()
--Function List:
hello
version
list
list_thread
list_sem
list_mutex
list_event
list_mb
list_mq
list_memp
list_timer
--Variable List:
    0, 0x00000000
finsh>>
```

运行创建的 hello 命令

```
\ | /
- RT -      Thread Operating System
/ | \      1.1.0 build Aug 17 2012
2006 - 2012 Copyright by rt-thread team

finsh>>list()
--Function List:
hello
version
list
list_thread
list_sem
list_mutex
list_event
list_mb
list_mq
list_memp
list_timer
--Variable List:
      0, 0x00000000
finsh>>hello()
Hello RT-Thread!
      0, 0x00000000
finsh>>
```

下面看看第二种、第三种方法下的打印结果:

```
\ | /
- RT -      Thread Operating System
/ | \      1.1.0 build Aug 17 2012
2006 - 2012 Copyright by rt-thread team

finsh>>list()
--Function List:
fun_with_arg      -- funciton with a argument
list_mem          -- list memory usage information
hello             -- say hello world
version           -- show RT-Thread version information
list_thread       -- list thread
list_sem          -- list semaphore in system
list_event        -- list event in system
list_mutex        -- list mutex in system
list_mailbox      -- list mail box in system
list_msgqueue     -- list message queue in system
```

```

list_mempool    -- list memory pool in system
list_timer     -- list timer in system
list_device    -- list device in system
list           -- list all symbol in system
[l] fun
--Variable List:
dummy          -- dummy variable for finsh
               0, 0x00000000
finsh>>

```

可以看到，list() 列出的命令中已然包含了我们刚才添加的两个函数，试着运行一下

```

\ | /
- RT -      Thread Operating System
/ | \      1.1.0 build Aug 17 2012
2006 - 2012 Copyright by rt-thread team
finsh>>list()
--Function List:
fun_with_arg    -- funciton with a argument
list_mem        -- list memory usage information
hello          -- say hello world
version         -- show RT-Thread version information
list_thread     -- list thread
list_sem        -- list semaphore in system
list_event      -- list event in system
list_mutex      -- list mutex in system
list_mailbox    -- list mail box in system
list_msgqueue   -- list message queue in system
list_mempool    -- list memory pool in system
list_timer      -- list timer in system
list_device     -- list device in system
list           -- list all symbol in system
[l] fun
--Variable List:
dummy          -- dummy variable for finsh
               0, 0x00000000
finsh>>fun()
fun is performed by finsh.
               0, 0x00000000
finsh>>fun_with_arg(7)
fun's arg is: 7
               0, 0x00000000

```

结果分析

本例程主要想要就是为大家展示了 finsh 组件如何添加自定义函数使之运行。