

VxWorks Kernel Shell 的启用

Kernel Shell 是 VxWorks 系统的一个 Component (组件)，是在 Target 端驻留的命令。可以通过系统默认的全局 IO 来访问，即多数架构的串口 0 或者 X86 架构的 PC Console；也可以通过 Telnet 或 rlogin 进行远程访问。

在 VxWorks 5.x 年代，还没有 RTP，这个组件就叫做 Target Shell。到了 VxWorks 6，为了表示它是在 Kernel 里运行，而不是 RTP 里，就改了名字。

Kernel Shell 很强大，有很多作用，可以用来加载、执行、监控、调试 Kernel 和 RTP 代码。而且随着 VxWorks 版本的提高，Kernel Shell 在功能也在加强。我们先来看看怎么用它，至于其它功能，下次再详细阐述。

首先，需要在 VIP 中包含组件 INCLUDE_SHELL

Components

Component Configuration			
Description	Name	Type	
shell history saving/loading mechanism	INCLUDE_SHELL_HISTORY_...		
shell hooks	INCLUDE_SHELL_HOOKS		
target-resident kernel shell (default)	INCLUDE SHELL		
Default kernel shell task spawn options	SHELL_TASK_OPTIONS	uint	
Options of tasks spawned by the shell	SHELL		
Priority of tasks spawned by the shell	SHELL_SPAWNED_TASK	uint	

有了这个组件，Kernel 就会启动一个 Shell 任务。任务的属性由这几个宏来定义

- ❖ SHELL_TASK_PRIORITY, 优先级，默认值 1
 - ❖ SHELL_STACK_SIZE, 栈，默认值 0x10000
 - ❖ SHELL_TASK_NAME_BASE, 任务名前缀，默认值 "tShell"，后跟从 0 开始递增的序号
 - ❖ SHELL_TASK_OPTIONS, 选项，默认值 (VX_UNBREAKABLE | VX_PRIVATE_ENV)
- | target-resident kernel shell (default) | INCLUDE SHELL | | |
|---|-------------------------------|--------|-------------------------------------|
| Default kernel shell task spawn options | SHELL_TASK_OPTIONS | uint | (VX_UNBREAKABLE VX_PRIVATE_ENV) |
| Options of tasks spawned by the shell | SHELL_SPAWNED_TASK_OPTIONS | uint | (VX_COPROCS_ALL_TASK VX_STDIO) |
| Priority of tasks spawned by the shell | SHELL_SPAWNED_TASK_PRIORITY | uint | 100 |
| Shell stack size | SHELL_STACK_SIZE | uint | 0x10000 |
| Stack size of tasks spawned by the shell | SHELL_SPAWNED_TASK_STACK_SIZE | uint | 20000 |
| compatible behavior of the kernel shell | SHELL_COMPATIBLE | BOOL | FALSE |
| default kernel shell configuration variables | SHELL_DEFAULT_CONFIG | string | "LINE_EDIT_MODE=,LINE_LENGTH=256,.. |
| initial shell session configuration variables | SHELL_FIRST_CONFIG | string | NULL |
| kernel shell launched at boot time | SHELL_START_AT_BOOT | BOOL | TRUE |
| kernel shell login access | SHELL_SECURE | BOOL | TRUE |
| kernel shell task basename string | SHELL_TASK_NAME_BASE | string | "tShell" |
| kernel shell task priority | SHELL_TASK_PRIORITY | uint | 1 |
| maximum number of simultaneous shell sessions | SHELL_MAX_SESSIONS | int | -1 |
| remote shell session configuration variables | SHELL_REMOTE_CONFIG | string | NULL |

```

-> ti tShell10

```

NAME	ENTRY	TID	PRI	STATUS	PC	SP
tShell10	shellTask	16f65d0	1	PEND	50d7c2	16f6240

```

full task name : tShell10
task entry      : shellTask
process         : kernel
options         : 0x1009083
VX_SUPERVISOR_MODE VX_UNBREAKABLE VX_DEALLOC_TCB VX_FP
VX_PRIVATE_ENV   VX_DEALLOC_EXC_STACK

STACK          BASE          END          SP          SIZE          HIGH          MARGIN
-----
execution      16f65d0      16e65d0      16f6240      65536         8132         57404
exception      1670410      166e930

```

而进入 Kernel Shell 后，通过 `sp()`、`repeat()` 或 `period()` 再创建的任务会使用以下属性

- SHELL_SPAWNED_TASK_STACK_SIZE, 栈，默认值 20000；也可通过 `spTaskStackSize` 动态修改
- SHELL_SPAWNED_TASK_PRIORITY, 优先级，默认值 100；也可通过 `spTaskPriority` 动态修改
- SHELL_SPAWNED_TASK_OPTIONS, 选项，默认值 (`VX_COPROCS_ALL_TASK | VX_STDIO`)，其中不管 CPU 是什么结构，`VX_COPROCS_ALL_TASK` 都会包含 `VX_FP_TASK`；也可通过 `spTaskOptions` 动态修改
- `namelessPrefix`, 任务名前缀，默认值 "t"，后跟从 1 开始递增的序号

```

-> spTaskStackSize
spTaskStackSize = 0x577218: value = 20000 = 0x4e20 = ' '
-> spTaskPriority
spTaskPriority = 0x577218: value = 100 = 0x64 = 'd'
-> spTaskOptions
spTaskOptions = 0x577214: value = 16777232 = 0x1000010
\ printf "%s\n", namelessPrefix
t
value = 2 = 0x2

```

```

-> sp taskDelay,10000
Task spawned: id = 0x16e6330, name = t1
value = 24011568 = 0x16e6330 = '0'
-> ti t1

```

NAME	ENTRY	TID	PRI	STATUS	PC	SP
t1	taskDelay	16e6330	100	DELAY	514132	16e62d0

```

full task name : t1
task entry      : taskDelay
process         : kernel
options         : 0x1009011
VX_SUPERVISOR_MODE VX_DEALLOC_TCB VX_FP_TASK VX_ST
VX_DEALLOC_EXC_STACK

STACK          BASE          END          SP          SIZE          HIGH          MARGIN
-----
execution      16e6330      16e1510      16e62d0      20000         1180         10920
exception      166e400      166c920

```

如果要远程访问 Kernel Shell，可使用以下两个组件

- ❖ rlogin - INCLUDE_RLOGIN
- ❖ Telnet - INCLUDE_IPTELENETS

Components

Component Configuration			
Description	Name	Ty...	Value
> RPC Components	FOLDER_RPC		
▼ Remote Access Components (default)	FOLDER_REMOTE_ACCESS		
NetDrv for remote IO (default)	INCLUDE_NET_DRV		
▼ RLOGIN	INCLUDE_RLOGIN		
RLOGIN_MAXCLIENTS	RLOGIN_MAXCLIENTS	long	2
rlogin client numbers	RLOGIN_CLIENT_NUMBERS	int	1
> Remote Command (default)	INCLUDE_REMLIB		
network remote I/O access (default)	INCLUDE_NET_REM		
> SNMP Components	FOLDER_SNMP		

Components

Component Configuration			
Description	Name	Ty...	Value
> Show Routine Components (default)	FOLDER_NET_SHOW		
▼ TELNET Components (default)	FOLDER_TELNET		
▼ TELNET Server	INCLUDE_IPTELENETS		
IPCOM TELNET port	IPCOM_TELNET_PORT	ch...	"23"
Idle timeout for telnet client	IPCOM_TELNET_IDLE_TIMEOUT	ch...	"360"
Maximum number of simultaneous TELNET clients	IPCOM_TELNET_MAX_CLIENTS	ch...	"0"
TELNET client	INCLUDE_TELNET_CLIENT		
> TFTP Components	FOLDER_TFTP		

另外，Kernel Shell 还可以选择是否使用登录密码，密码长度是 8-40

- ❖ SHELL_SECURE -> TRUE
- ❖ INCLUDE_SECURITY

Component Configuration			
Description	Name	Type	Value
initial shell session configuration variable	SHELL_FIRST_CONFIG	string	NULL
kernel shell launched at boot time	SHELL_START_AT_BOOT	BOOL	TRUE
kernel shell login access	SHELL_SECURE	BOOL	TRUE
kernel shell task basename string	SHELL_TASK_NAME_BASE	string	"tShell"
kernel shell task priority	SHELL_TASK_PRIORITY	int	1
maximum number of simultaneous shell sessions	SHELL_MAX_SESSIONS	int	1

并且这个密码会通过 Salt 值加密。在 VxWorks DevelopmentShell 中使用工具 vxencrypt 来转换密码

```

VxWorks Development Shell
D:\>vxencrypt
please enter password:
salt is 2iAAAip1AAA=
hashed password is gQ+aiqYYmpQQuQVhR9ji9BT2cqDU

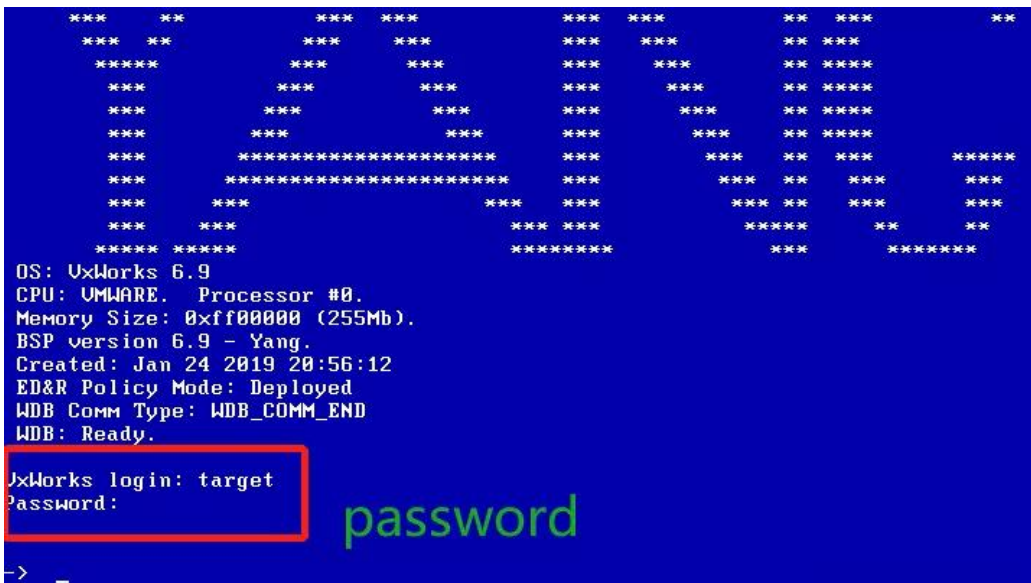
```

例如输入“password”作为密码，得到随机的 Salt 值和哈希加密后的密码，将其填入对应的组件属性中

Components

Description	Name	Type	Value
System Level Security Components	FOLDER_SYSTEM_SECURI...		
system-level password protection	INCLUDE_SECURITY		
enables SYSFLG_NO_SECURITY boot flag su...	SYSFLG_NO_SECURITY_E...	BOOL	FALSE
rlogin/telnet encrypted password	LOGIN_PASSWORD	char *	"gQ+aiqYYmpQQuQ...
rlogin/telnet encrypted password salt	LOGIN_PASSWORD_SALT	char *	"2iAAAlb1AAA="
rlogin/telnet user name	LOGIN_USER_NAME	char *	
Startup Sequence and Initialization Components	FOLDER_SSI		

编译后再启动 VxWorks 时，就需要输入 user name 和 password 了



使用 Telnet 也是如此

