



Shell User Manuel

Shell command user manuel

SPC002012 V1.00 Date: 2018/05/17

Use manuel



Contents

1. Add and configure	1
1.1. Introduction	1
1.2. Configure	1
1.2.1. Configure shell	1
1.2.2. Configure a shell command	1
1.3. Add	2
2. System commands	4
2.1. Introduction	4
2.2. Use of commands	5
2.2.1. aborts - Display current operating system exception handling statistics	5
2.2.2. affinity - a set of CPUs that display or set thread or process scheduling	6
2.2.3. buss - Displays all bus information mounted in the system	8
2.2.4. clear - clear the current screen	8
2.2.5. color - Initialize color scheme based on LS_COLORS	9
2.2.6. cpuus - View cpu utilization	10
2.2.7. devs - Display all devices mounted in the system	11
2.2.8. drvlics - Display all installed driver table license information in the system	12
2.2.9. drvs - Display device driver table information	13
2.2.10. env - View the current environment variable table	14
2.2.11. echo - Echo command, this command will echo user input parameters	15
2.2.12. eventset - Display event set information	16
2.2.13. exec - Execute a program	16
2.2.14. exit - Exit the current shell terminal	17
2.2.15. free - Display the current memory information of the system	17
2.2.16. help - Display a list of all built-in commands of ttinyShell. If there is a parameter, description of the command named after the parameter can be displayed.	18
2.2.17. hostname - Display or set the current SylixOS image host name	19
2.2.18. ints - View system interrupt vector table information	20
2.2.19. kill - Send a signal to the specified thread or process. The SIGKILL signal is sent by default.	21
2.2.20. login - Switch user, login again	22
2.2.21. loglevel - Display or set the current kernel log (printk) print level	23
2.2.22. mems - View operating system kernel memory heap and system memory heap memory usage	24
2.2.23. part - Display the specified memory partition information	24
2.2.24. pcis - Display PCI bus information	26
2.2.25. ps - View information about all processes in the system	27

2.2.26. reboot - Restart the computer	27
2.2.27. renice - Set the priority of the specified process	28
2.2.28. restart - Restart thread	29
2.2.29. sem - Display signal information	30
2.2.30. shell - Create a shell using ttydevice as the standard file	31
2.2.31. shstack - Display or set the shell task stack size.....	31
2.2.32. shutdown - shut down or restart the system.....	32
2.2.33. sigqueue - Send a signal to a thread or process.....	33
2.2.34. sleep - Current thread sleep time specified	34
2.2.35. sprio - set the priority of the specified thread.....	34
2.2.36.ss - View all thread and interrupt system stack usage in the system	35
2.2.37. top - View cpu usage rate	36
2.2.38. tp - View blocked thread information in the system	37
2.2.39. ts - View system thread information	38
2.2.40. tty - Display the tty file corresponding to the current shell terminal	39
2.2.41. varedl- Delete a specified system environment variable.....	39
2.2.42. varload - Extract the loading environment variable table from the file with the specified parameters.....	40
2.2.43. vars - Display the current environment variable	41
2.2.44. varsave – save the current operating system environment variables table	42
2.2.45. virtuals - Display vmm virtual memory information	43
2.2.46. which - Check the file location specified by the parameter	44
2.2.47. who - View current login user identity.....	45
2.2.48. zones - Check operating system physical page partition usage	45
3. File command	47
3.1. Introduction.....	47
3.2. Use of commands.....	48
3.2.1. cat- Display contents of the file.....	48
3.2.2. cd- Switch the current directory.....	49
3.2.3. ch- Change the directory.....	49
3.2.4. chmod - Change the access permissions of a file or directory.	50
3.2.5. close – Close a file.....	50
3.2.6. cmp-Compare a file.....	51
3.2.7. cp- Copy a file	51
3.2.8. df- View the file system information in the specified directory	52
3.2.9. dosfslabel – View or set file system label.....	53
3.2.10. dsize - Calculate all file information contained in a specified directory.....	53
3.2.4. fatugid – Set fat file system user and group id	54
3.2.12. fdentrys – List all file information that operating system is operating...54	
3.2.13. fdisk- Display or make a disk partition table.....	55

3.2.14. files-List all file information that operating system is operating	56
3.2.15. gzip - Compress or decompress a file	57
3.2.16. ll - Display the detailed information in the specified directory	58
3.2.17. ln- Create a symbolic link file.....	59
3.2.18. logfileadd - Add kernel file descriptor to kernel log print function	59
3.2.19. Logfileclear -Clear the specified kernel file descriptor from the kernel log print file table.....	60
3.2.20. logfiles -Display the list of kernel log print files	60
3.2.21. ls - Display the file name in the specified directory, with the default current directory.....	61
3.2.22. mkdir -Create a directory.....	62
3.2.23. mkfifo - Create a named pipe that can only be created under the root file system device	62
3.2.24. mkfs -Format the specified disk	63
3.2.25. mmaps - Display the system mmap information	64
3.2.26. Mount - Mount a volume.....	64
3.2.27. msgq - Display message queue information.....	65
3.2.28. mv- Move to or rename a file.....	66
3.2.29. open- Open a file.....	66
3.2.30. pwd-View the current working directory.....	67
3.2.31. rm-Delete a file.....	68
3.2.32. rmdir- Delete a folder	68
3.2.33. showmount -View all mounted volumes in the system.....	69
3.2.34. Shfile - Execute the specified shell script	70
3.2.35. Sync - Write all system cached information to physical devices.....	70
3.2.36. tmpname - Get a temporary file name that can be created	71
3.2.37. Touch - Create a common file.....	71
3.2.38. umount - Unload a volume.....	72
3.2.39. untar - Unpack or untar a tar or tar.gz package.	73
3.2.40. vi- Start the vi editor	73
3.2.41. yaffscmd - Display, set, and erase a block.....	74
3.2.42. zlib- Add a .gz compressed file	75
4. User's commands.....	77
4.1. Introduction.....	77
4.2. Use of commands	77
4.2.1. gadd - Add a new user group.....	77
4.2.2. gdel - Delete a user group.....	77
4.2.3. group - Display the information of user group.....	78
4.2.4. pmod - Modify user password	79
4.2.5. uadd - Add user	79
4.2.6. udel - Delete user	80
4.2.7. umod - Set the user's mode.....	80
4.2.8. user - Display user information or generate a password for the user.....	81
5. Network	83

5.1. Introduction.....	83
5.2. Use of commands	83
5.2.1. aodvs – Display the aodv routing table.....	83
5.2.2. arp – Add, delete or view ARP tables	84
5.2.3. ftpdpath – View or set the ftp server initialization path	85
5.2.4. ftpds – Display ftp server information.....	85
5.2.5. hosttable – View, add or delete host address mapping	86
5.2.6. ifconfig – Display or configure network configuration information ..	87
5.2.7. ifdown – Disable a network interface.....	88
5.2.8. ifrouter – Set the default route interface	88
5.2.9. ifup – Enable a network interface	89
5.2.10. ipv6 – Set or show ipv6	89
5.2.11. nat – Enable, disable or set the NAT virtual network address service	90
5.2.12. natalias – Add or delete NAT alias	91
5.2.13. natmap – Add or delete NAT mapping.....	91
5.2.14. nats – View current NAT virtual address service status.....	92
5.2.15. nbname – Display or set the name of this machine’s NetBIOS	93
5.2.16. netstat – View network status.....	94
5.2.17. npfattach – Enable network packet filter on the specified network interface.....	95
5.2.18. npfdetach – Disable network packet filters on the specified network interface.....	96
5.2.19. npfruleadd – Add a network packet filter rule	96
5.2.20. npfruledel – Delete a network packet filter rule	97
5.2.21. npfs – View network packet filter status	98
5.2.22. ping – Ping command.....	98
5.2.23. ping6 – Ipv6 Ping command.....	99
5.2.24. route – Add, delete, modify, or view system routing tables	100
5.2.25. tftp – Use the tftp command to receive or send a file	102
5.2.26. tftpdpath – View or set the local path of the tftp server	102
5.2.27. vlan – Display, set and delete net interface’s VLANID	103
5.2.28. vpnclose – Delete a virtual network interface**	104
5.2.29. vpnopen – Create a virtual network interface**	104
6. Time.....	107
6.1. Introduction.....	107
6.2. Use of commands	107
6.2.1. date – Display or set the current system time.....	107
6.2.2 hwclock – Display or synchronize the operating system and hardware RTC clock.....	107
6.2.3. times – Display utc or local time.....	108
6.2.4. tzsync – Time zone synchronization with environment variable TZ	109
7. Dynamic loading	111
7.1 Introduction.....	111

7.2 Use of commands.....	111
7.2.1. debug – debug a process.....	111
7.2.2 dlconfig – configure the operating parameters of dynamic linker	112
7.2.3 leakchk – check the memory leak.....	113
7.2.4 leakchkstart – start the memory leak tracker	113
7.2.5 leakchkstop – close the memory leak tracker	114
7.2.6 lsmod – view the information of all kernel modules loaded by the system.....	114
7.2.7 modulegcov – generate the kernel module code file (*.gcda).....	115
7.2.8 moduleunreg – register a module.....	116
7.2.9 modules – view the information of all kernel modules and process dynamic link librarys loaded by the system	116
7.2.10 modulestat – view the information of a kernel module or dynamic link library file.....	117
7.2.11 moduleunreg – unload a module	118
8. Others	120
8.1 Introductoin.....	120
8.2 Use of commands.....	120
8.2.1 args – display all input parameters, with spaces as delimiters	120
8.2.2 crypt – encrypt data.....	121
8.2.3. perfrefresh – update performance statistics	121
8.2.4. perfs – display performance statistics	122
8.2.5. perfstart – start the performance analysis tool.....	122
8.2.6. perfstop – stop the performance analysis tool	123
8.2.7. xmodemr - receive a file using the xmodem protocol.....	123
8.2.8. xmodems - send a file using the xmodem protocol.....	124

1. Add and configure

1.1. Introduction

SpaceChain OS and SylixOS share the same shell and have the same usage. This section describes how to add or configure shell naming.

1.2. Configure

1.2.1. Configure shell

SpaceChain OS shell support can be tailored. If you do not need shell functionality, just set the `LW_CFG_SHELL_EN` configuration macro to 0.

To configure `LW_CFG_SHELL_EN` to 0, find `/libsylxos/SylixOS/config/shell/shell_cfg.h` under Base project, just set `LW_CFG_SHELL_EN` to 0, recompile the Base and Bsp projects, and restart the virtual machine.

1.2.2. Configure a shell command

SylixOS shell command support can be tailored. If you need to tailor a command, you can trim the command by setting the configuration macro of the specific command to 0.

Take the `uadd` command as an example to view the `uadd` command description:

`Uadd - add user`

Format:

`uadd name password enable[0 / 1] uid gid comment homedir`

Explanation:

This command is used to add a user.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

User id

It cannot be repeated. When adding a user, it must be added to an existing user group.

Example:

Configuration:

When `LW_CFG_SHELL_EN>0`, the operating system is allowed to provide `tshell` commands.

When (`LW_CFG_SHELL_USER_EN> 0`), the shell user management tool is enabled, and this command will be included.

Function interface:

The `uadd` command is implemented by a C language function. The function prototype is:
`static INT __tshellUserCmdUadd (INT iArgC, PCHAR ppcArgV[]);`

Configure `LW_CFG_SHELL_USER_EN` to 0 and trim the `uadd` command. Locate the `shell_cfg.h` file, set `LW_CFG_SHELL_USER_EN` to 0, recompile the base and bsp files, and then restart the virtual machine. (The path to the `shell_cfg.h` configuration file is `base/libsylxos/SylixOS/config/shell/shell_cfg.h`)

Use `help` to see if there is a `uadd` command.


```

[root@sylixos:/root]# help uadd
[root@sylixos:/root]# help udel
[root@sylixos:/root]# uadd
sh: command not found.
[root@sylixos:/root]# udel
sh: command not found.
[root@sylixos:/root]#

```

Check to see that the uadd command's help file and the command have been trimmed.

Note: The configuration shell command should configure the minimum entry macro to reduce the scope of the configuration. The uadd command was configured in the above example, but related commands such as udel, umod, gadd, gdel, and other related commands were also trimmed.

1.3. Add

The SylixOS system shell command allows users to add their own commands. Add as:

Write the corresponding interface function in ttintShellSysCmd.c and add the shell command in __tshellSysCmdInit. Three system interfaces will be used here. Here are:

```

LW_API ULONGAPI_TShellKeywordAdd (
    CPCHAR    pcKeyword, PCOMMAND_START_ROUTINE    pfuncCommand);
LW_API    ULONGAPI_TShellFormatAdd (CPCHAR    pcKeyword,    CPCHAR
pcFormat);
LW_API    ULONG    API_TShellHelpAdd (CPCHAR    pcKeyword,    CPCHAR
pcFormat);

```

Among which, API_TShellKeywordAdd is used to add a shell command and associate the shell command with the corresponding processing function.

The API_TShellFormatAdd function is used to add the format of the corresponding shell command.

TShellHelpAdd is used to add instructions for corresponding shell commands and describes the corresponding functions that the command needs to perform.

Example: Add a hello shell command.

1. Add __tshellCostomCmdHello function in ttinyShellSysCmd.c file.

```

static INT __tshellCostomCmdHello(INT    iArgC, PCHAR    ppcArgU[])
{
    if(iArgC > 2)
    {
        fprintf(stderr, "argument error.\n");
        return (-ERROR_TSHELL_EPARAM);
    }
    if (iArgC == 1)
    {
        printf("hello SylixOS\n");
    }
    if (iArgC == 2)
    {
        printf("hello %s\n", ppcArgU[1]);
    }
    return 0;
}

```

2. Add code in the __tshellSysCmdInit function.

```

VOID __tshellSysCmdInit (VOID)
{
    API_TShellKeywordAdd("hello", __tshellCostonCmdHello);
    API_TShellFormatAdd ("hello", " [message]");
    API_TShellHelpAdd ("hello", "show hello sylixOS\n");
}

```

3. Recompile the base project and bsp project. Then start the virtual machine to test whether the addition was successful.

```

[root@sylixos:/root]# help hello
show hello sylixOS
hello [message]
[root@sylixos:/root]# hello
hello SylixOS
[root@sylixos:/root]# hello world
hello world
[root@sylixos:/root]# hello SylixOS world
argument error.
parameter(s) error.
[root@sylixos:/root]#

```

The test shows that the addition was successful.

2. System commands

2.1. Introduction

System-related commands are:

- aborts - Display the current operating system exception handling statistics
- affinity - the set of CPUs that display or set thread or process schedules
- Buss - Display all bus information mounted in the system
- clear - Clear the current screen
- color - Initialize the color scheme according to LS_COLORS
- cpuus - View cpu utilization
- devs - Display all devices mounted in the system
- drvlics - Display all installed driver table license information in the system
- drvs - Display device driver table information
- echo - Echo command, this command will echo user input parameters
- env - View the current environment variable table
- eventset - Display the event set information
- exec - Execute a program
- Exit - Exit the current shell terminal
- free - Display the current memory information of the system
- help - Display the list of all built-in commands of tinyShell. If there is a parameter, the introduction to the command named after the parameter can be displayed.
- hostname - Display or set the current SylixOS image host name
- ints - View system interrupt vector table information
- kill - Send a signal to the specified thread or process. By default, the SIGKILL signal is sent.
- login - Switch user, log in again
- Loglevel - Display or set the current kernel log (printk) print level
- mems - View memory usage of the operating system kernel memory heap and system memory heap
- Part - Display the specified memory partition information
- pcis - Display PCI bus information
- ps - View information about all processes in the system
- reboot - Restart the computer
- renice - Set the priority of the specified process
- restart - Restart a thread
- Sem - Display signal information
- Shell - Create a shell using ttydevice as a standard file
- Shstack - Display or set the shell task stack size
- shutdown - Shut down or restart the system
- sigqueue - Send a signal to a thread or process
- sleep - Current thread sleep time
- sprio - Set the priority of the specified thread
- ss - View all system threads and interrupt system stack usage

- top – Check the CPU usage
- tp - View blocked thread information in the system
- ts - View system thread information
- Tty - Display the tty file corresponding to the current shell terminal
- vardel - Delete a specified system environment variable
- varload - Extract the load environment variable table from the specified parameter file
- vars - Show the current environment variable
- varsave - Save the current operating system environment variable table
- virtuals - Display vmm virtual memory information
- which - Check the file location specified by the parameter
- who - View current logged-in user's identity
- zones - Check operating system physical page partition usage

2.2. Use of commands

System-related shell commands are introduced and used in detail. They mainly use the format, description, remarks, usage examples, and configuration and response interfaces.

2.2.1. aborts - Display current operating system exception

handling statistics

Format:

aborts

Explanation:

This command displays the number of current memory access exceptions, memory allocation errors, physical page allocation errors, and page mapping errors for the system.

Return value:

The return value is 0.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# aborts
vmm abort statistics information show >>
vmm abort (memory access error) counter : 0
vmm page fail (alloc success) counter   : 0
vmm alloc physical page error counter   : 0
vmm page map error counter               : 0
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell command. When LW_CFG_VMM_EN>0, virtual memory is supported, and this command will be included.

Function interface:

The aborts command is implemented via a c language function. The function prototype is: static INT __tshellSysCmdAborts (INT iArgC, PCHAR ppcArgV[]);

2.2.2. affinity - a set of CPUs that display or set thread or process scheduling

Format:

```
affinity
affinity pid | thread id cpu id
affinity pid | thread id 'clear'
```

Explanation:

There are three usages of this command. When the affinity has no parameters, it shows the cpu set of thread and process scheduling. When the affinity parameter is the thread id or process id and cpu id, set the cpu set of the process or thread scheduling; when the affinity parameter is the thread id or process id and clear, clear the scheduling of the process or thread on the cpu set.

Return value:

Return "0" on sucesss and "-1" on failure.

Remarks:

The thread or process scheduling cpu information is stored in the file /proc/kernel/affinity.

Example:

```

[root@sylixos_station:/root]# affinity
      NAME      TID    PID  CPU
-----
t_idle0        4010000    0    0
t_itimer       4010001    0    *
t_isrdefer     4010002    0    *
t_except       4010003    0    *
t_log          4010004    0    *
t_power        4010005    0    *
t_hotplug      4010006    0    *
t_reclaim      4010008    0    *
t_netjob       4010009    0    *
t_netproto     401000a    0    *
t_tftpd        401000b    0    *
t_ftpd         401000c    0    *
t_telnetd      401000d    0    *
t_tshell       401000f    0    *
[root@sylixos_station:/root]# affinity 4010009 0
affinity set thread 0x4010009 to cpu 0 ok.
[root@sylixos_station:/root]# affinity
      NAME      TID    PID  CPU
-----
t_idle0        4010000    0    0
t_itimer       4010001    0    *
t_isrdefer     4010002    0    *
t_except       4010003    0    *
t_log          4010004    0    *
t_power        4010005    0    *
t_hotplug      4010006    0    *
t_reclaim      4010008    0    *
t_netjob       4010009    0    0
t_netproto     401000a    0    *
t_tftpd        401000b    0    *
t_ftpd         401000c    0    *
t_telnetd      401000d    0    *
t_tshell       401000f    0    *
[root@sylixos_station:/root]# affinity 4010009 clear
affinity clear thread 0x4010009 ok.
[root@sylixos_station:/root]# affinity
      NAME      TID    PID  CPU
-----
t_idle0        4010000    0    0
t_itimer       4010001    0    *
t_isrdefer     4010002    0    *
t_except       4010003    0    *
t_log          4010004    0    *
t_power        4010005    0    *
t_hotplug      4010006    0    *
t_reclaim      4010008    0    *
t_netjob       4010009    0    *
t_netproto     401000a    0    *
t_tftpd        401000b    0    *
t_ftpd         401000c    0    *
t_telnetd      401000d    0    *
t_tshell       401000f    0    *
[root@sylixos_station:/root]# █

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell

command. When (LW_CFG_SMP_EN > 0) and (LW_CFG_POSIX_EN > 0), and the system supports the multiprocessor and enables the posix compatible library, this command will be included.

Function interface:

The affinity command is implemented through a c language function. The function prototype is: static INT __tshellSysCmdAffinity (INT iArgC, PCHAR ppcArgV[]);

2.2.3. buss - Displays all bus information mounted in the system

Format:

buss

Explanation:

This command is used to display all bus information mounted in the system.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# buss

  BUS NAME    DEV NUM
-----
/bus/i2c/0      0
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The buss command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdBuss (INT iArgC, PCHAR ppcArgV[]);

2.2.4. clear - clear the current screen

Format:

clear

Explanation:

This command is used to clear the current screen display.

Return value:

Return "0".

Remarks:

N/A.

Example:

```

machine : SAMSUNG S3C2440A (ARM920T 405/101MHz NonFPU)

      [[                                     (R)
[[[[      [[[[      [[[[      [[[[      [[[[
[[  [[      [[      [[      [[      [[      [[      [[      [[      [[      [[
[[      [[  [[      [[      [[      [[      [[      [[      [[      [[      [[
[[      [[  [[      [[      [[      [[      [[      [[      [[      [[      [[
      [[      [[      [[      [[      [[      [[      [[      [[      [[      [[
      [[      [[      [[      [[      [[      [[      [[      [[      [[      [[
      [[      [[      [[      [[      [[      [[      [[      [[      [[      [[
[[  [[      [[      [[      [[      [[      [[      [[      [[      [[      [[
[[[[      [[[[      [[[[[[      [[[[[[      [[      [[      [[      [[      [[
      [[
      [[      KERNEL: Long-Wing(C) 1.3.5 (5)
[[[[      COPYRIGHT ACOINFO Co. Ltd. 2006 - 2017

SylixOS license: Commercial & GPL.
SylixOS kernel version: 1.3.5 (5) LongYuan(b)

CPU      : SAMSUNG S3C2440A (ARM920T 405/101MHz NonFPU)
CACHE    : 32KBytes L1-Cache (D-16K/I-16K)
PACKET   : Mini2440 Packet
ROM SIZE: 0x00200000 Bytes (0x00000000 - 0x001fffff)
RAM SIZE: 0x04000000 Bytes (0x30000000 - 0x33ffffff)
BSP      : BSP version 5.1.2 for GEMINI
[root@sylixos_station:/root]# clear

```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The clear command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdClear (INT iArgC, PCHAR ppcArgV[]);

2.2.5. color - Initialize color scheme based on LS_COLORS

Format:

color

Explanation:

This command is used to update the color scheme.

Return value:

Return "0".

Remarks:

N/A.

Example:

```

[root@sylixos_station:/root]# color
[root@sylixos_station:/root]#

```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands,

and this command is included.

Function interface:

The color command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdColor (INT iArgC, PCHAR ppcArgV[]);

2.2.6. cpuus - View cpu utilization

Format:

cpuus

cpuus -n times -t wait_seconds

Explanation:

There are two usages of this command, where -n is the number of times the cpu utilization is detected and -t is the time spent in each check. The cpuus detects one time by default, with the detection time being 1s, and cpuus -n times -t wait_seconds can designate the times (times) of detection, and the time (wait_seconds) of each detection. The specified detection time must not exceed 10 seconds.

返回值:

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/etc]# cpuus
CPU usage checking, please wait...
CPU usage show (measurement accuracy 1.0%) >>

      NAME      TID  PRI   CPU   KERN
-----
t_tshell      401000f 150   0.0%   0.0%
t_telnetd     401000d 160   0.0%   0.0%
t_ftpd        401000c 160   0.0%   0.0%
t_tftpd       401000b 160   0.0%   0.0%
t_netproto    401000a 110   0.0%   0.0%
t_netjob      4010009 110   0.0%   0.0%
t_reclaim     4010008 253   0.0%   0.0%
t_hotplug     4010006 250   0.0%   0.0%
t_power       4010005 254   0.0%   0.0%
t_log         4010004  60   0.0%   0.0%
t_except      4010003  0    0.0%   0.0%
t_isrdefer    4010002  0    0.0%   0.0%
t_itimer      4010001  20   0.0%   0.0%
t_idle0       4010000 255  99.0%   0.0%

[root@sylixos_station:/etc]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The cpuus command is implemented through a c language functions. The function prototype is: static INT __tshellSysCmdCpuus (INT iArgC, PCHAR ppcArgV[]);

2.2.7. devs - Display all devices mounted in the system

Format:

devs
devs-a

Explanation:

This command is used to display the information about the mounted device in the system. Among which, devs without parameters displays the device number, opening status, and device name of the mounted device, devs with parameter -a displays all the information of the mounted device, including the device number, opening status, device name, and device type.

返回值:

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# devs
device show (minor device) >>
drv open name
32    0 /dev/input/xmse
32    0 /dev/input/xkbd
31    0 /dev/socket
30    0 /dev/netevent
28    1 /dev/input/touch0
29    0 /dev/fb0
26    0 /dev/yaffs2
17    0 /dev/ttyS2
17    0 /dev/ttyS1
17    1 /dev/ttyS0
14    0 /dev/urandom
14    0 /dev/random
13    0 /dev/shm
12    0 /dev/proc
```

```
[root@sylixos_station:/root]# devs -a
device show (minor device) >>
drv open name          type
32    0 /dev/input/xmse  character
32    0 /dev/input/xkbd  character
31    0 /dev/socket      socket
30    0 /dev/netevent    character
28    2 /dev/input/touch0 character
29    0 /dev/fb0         character
26    0 /yaffs2          directory
17    0 /dev/ttyS2       character
17    0 /dev/ttyS1       character
17    1 /dev/ttyS0       character
14    0 /dev/urandom      character
14    0 /dev/random       character
13    0 /dev/shm         directory
12    0 /proc            directory
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The devs command is implemented through a c language function. The function prototype is: static INT __tshellSysCmdDevs (INT iArgC, PCHAR ppcArgV[]);

2.2.8. drvlics - Display all installed driver table license

information in the system

Format:

drvlics

Explanation:

该命令用于显示系统中安装设备的设备号、设备的描述、作者和版本号。

This command is used to display the device number, device description, author, and version number of the installed device in the system.

Return value:

Return "0".

备注:

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# drvlics
driver License show (major device) >>
```

DRV	DESCRIPTION	AUTHOR	LICENSE
0	null device driver.	Han.hui	GPL->Ver 2.0
1	zero device driver.	Han.hui	GPL->Ver 2.0
2	rootfs driver.	Han.hui	GPL->Ver 2.0
3	eventfd driver.	Han.hui	GPL->Ver 2.0
4	timerfd driver.	Han.hui	GPL->Ver 2.0
5	hstimerfd driver.	Han.hui	GPL->Ver 2.0
6	signalfd driver.	Han.hui	GPL->Ver 2.0
7	gpiofd driver.	Han.hui	GPL->Ver 2.0
8	blk io driver.	Han.hui	GPL->Ver 2.0
9	epoll driver.	Han.hui	GPL->Ver 2.0
10	hotplug message driver.	Han.hui	GPL->Ver 2.0
11	hardware rtc.	Han.hui	GPL->Ver 2.0
12	procfs driver.	Han.hui	GPL->Ver 2.0

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The drvlics command is implemented through a c language functions. The function prototype is: static INT __tshellSysCmdDrvlics (INT iArgC, PCHAR ppcArgV[]);

2.2.9. drvs - Display device driver table information

Format:

drvs

Explanation:

This command will display the device driver's drive number (drv) and create, delete, open, close, read, write, control (ioctl), read device extension (Readex), write device extension (writeex), select, read/write pointer movement (lseek), create link file, read link file, and address of file mapping function.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# drvs
driver show (major device) >>
drv      create      delete      open      close      read      write      ioctl
1  301b9be0      0  301b9be0  301b9bd0  301b9c00      0  301b9cb4
2  3008b5f4  3008b4c4  3008b5f4  3008b024  3008ab90  3008ab78  3008aba8
3      0      0  3018db88  3018db20  3018d990  3018d828  3018d5d8
4      0      0  301c0c74  301c0bf8  301c0b14      0  301c0934
5      0      0  30190080  30190018  30190260      0  3018fe38
6      0      0  301c4428  301c3ebc  301c41b4      0  301c3f0c
7      0      0  3018e028  3018df9c  3018df40  3018deec  3018e20c
8      0      0  30081b94  30081b4c  30081c94  30081c28  3008137c
9      0      0  301b424c  301b4194      0      0  301b4358
10  301b52ec      0  301b52ec  301b5268  301b5160  301b4e70  301b4e88
11  30190b50      0  30190b50  30190b28      0      0  30190a28
12  30084414      0  30084414  300843a4  3008428c  30083b40  30083c2c
13  301afd00  301af664  301afd00  301af50c  301aec8c  301aec5c  301aeca4
14  301a5420      0  301a5420  301a54cc  301a5550  301a56e8  301a5700
15  301a4c18      0  301a4c18  301a4c3c  301a4c78  301a4d6c  301a4dc0
16  301a5000      0  301a5000  301a5024  301a5084  301a50b8  301a50ec

drv      readex      writeex      select      lseek      symlnk      readlnk      mmap
1      0      0      0      0      0      0      0
2      0      0      0      0      3008b30c  3008b2c8      0
3      0      0      0      0      0      0      0
4      0      0      0      0      0      0      0
5      0      0      0      0      0      0      0
6      0      0      0      0      0      0      0
7      0      0      0      0      0      0      0
8  30081914  30081670      0      0      0      0      0
9      0      0      0      0      0      0      0
10      0      0      0      0      0      0      0
11      0      0      0      0      0      0      0
12  3008419c  30083b28      0      0      0  30083b58      0
13  301aec74  301aec44      0  301aec38  301afcd8  301af304  301af8b4
14      0      0      0      0      0      0      0
15      0      0      0      0      0      0      0
16      0      0      0      0      0      0      0
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The drvs command is implemented via a c language function. The function prototype is: static INT __tshellSysCmdDrvs (INT iArgC, PCHAR ppcArgV[]);

2.2.10. env - View the current environment variable table

Format:

env

Explanation:

This command is used to display the global environment variable table of the system.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# env
variable show >>
```

VARIABLE	REF	VALUE
TERMCAP		/etc/termcap
TERM		vt100
LUA_CPATH		?..so;/usr/local/lib/lua/?..so;/usr/lib/lua/?..so;/lib/lua/?..so
LUA_PATH		?..lua;/usr/local/lib/lua/?..lua;/usr/lib/lua/?..lua;/lib/lua/?..lua
DEBUG_CPU		-1
PATH_LOCALE		/usr/share/locale
LC_ALL		
LANG		C
LD_LIBRARY_PATH		/usr/lib:/lib:/usr/local/lib
PATH		/usr/bin:/bin:/usr/pkg/sbin:/sbin:/usr/local/bin
NFS_CLIENT_PROTO		udp
NFS_CLIENT_AUTH		AUTH_UNIX
SYSLOGD_HOST		0.0.0.0:514
FIO_FLOAT		1
SO_MEM_PAGES		8192
TSLIB_CALIBFILE		/etc/pointercal
TSLIB_TSDEVICE		/dev/input/touch0
MOUSE		/dev/input/mouse0:/dev/input/touch0
KEYBOARD		/dev/input/keyboard0
TZ		CST-8:00:00
TMPDIR		/tmp/
LICENSE		SylixOS license: Commercial & GPL.
VERSION		1.3.5 (5)
SYSTEM		SylixOS kernel version: 1.3.5 (5) LongYuan(b)

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The env command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdVars (INT iArgC, PCHAR ppcArgV[]);

2.2.11. echo - Echo command, this command will echo user

input parameters

Format:

echo message

Explanation:

The parameters are echoed by this command.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# echo hello word
hello word
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

The echo command is implemented by a c language function. The function

prototype is: static INT __tshellSysCmdEcho (INT iArgC, PCHAR ppcArgV[]);

2.2.12. eventset - Display event set information

Format:

eventset eventset handle

Explanation:

This command is used to display the related information about event sets.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

Write a program using the SylixOS event set operation function to print the event set id number.

```
[root@sylixos:/apps/mq]# eventset 20010009
event set show >>

event set name   : event_set
event set event  : 0x00000000

[root@sylixos:/apps/mq]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The eventset command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdEventSet (INT iArgC, PCHAR ppcArgV[]);

2.2.13. exec - Execute a program

Format:

exec program filearguments

Explanation:

This command is used to execute a program, and exec is followed by the file name and parameter list.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# exec /apps/helloWord/helloWord
Hello SylixOS!
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MODULELOADER_EN> 0, module loading service needs to

be provided and this command will be included.

Function interface:

The exec command is implemented by a c language function. The function prototype is: static INT __tshellExec (INT iArgC, PCHAR *ppcArgV)

2.2.14. exit – Exit the current shell terminal

Format:

exit

Explanation:

This command is used to exit the current shell terminal.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# exit
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The exit command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdExit (INT iArgC, PCHAR ppcArgV[]);

2.2.15. free - Display the current memory information of the system

Format:

free

Explanation:

This command is used to display the current memory information. If virtual memory is supported, the vmm physical memory information is displayed.

Return value:

Return "0".

Remarks:

N/A.

Example:


```
[root@sylixos_station:/root]# free
heap show >>

-----
HEAP      TOTAL      USED      MAX USED  SEGMENT USED
-----
kersys    13279KB    3359KB    3376KB    1083  25%

vmm physical zone show >>

ZONE PHYSICAL  SIZE  PAGESIZE  PGD  FREEPAGE  DMA  USED
-----
0 31800000  600000  1000 30bc8000  1348 true  12%
1 31e00000  2200000  1000 30bc8000  8695 false  0%

ALL-Physical memory size: 64 MBytes (67108864 Bytes)
VMM-Physical memory size: 40 MBytes (41943040 Bytes)
VMM-Physical memory free: 39 MBytes (41136128 Bytes)
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The free command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdFree (INT iArgC, PCHAR ppcArgV[]);

2.2.16. help - Display a list of all built-in commands of ttinyShell. If there is a parameter, description of the command named after the parameter can be displayed.

Format:

```
help
help keyword
help-s keyword
```

Explanation:

There are only 3 usages of this command. When the help command is not followed by a parameter, it displays the list of all built-in commands of ttinyShell. The function of “help keyword” is the same as that of “help -s keyword”, both displaying the description of the command named after the keyword.

返回值:

Return value:

Return “1” on success and non-“0” on failure.

Remarks:

The help parameter keyword supports wildcard characters (*, ?). When wildcard characters are used, the list of matching built-in commands is listed.

Example:

```
[root@sylixos_station:/root]# help
ps
lsmod
modulegcov          [kernel module handle]
modules             [module name]
modulestat          [program file]
moduleunreg         [kernel module handle]
modulereg           [kernel module file *.ko]
dlconfig            {[share {en | dis}} | [refresh [*]]}
exec                [program file] [arguments...]
which               [program file]
mmaps
vpnclose            [netifname]
vpnopen             [configuration file]
npfdetach           [netifname]
npfattach           [netifname]
npfruledel          [netifname] [rule sequence num]
npfruleadd          [netifname] [rule] [args...]
npfs
nats
natmap              {[add] | [del]} [WAN port] [LAN port] [LAN IP] [protocol]
natalias            {[add] [alias] [LAN start] [LAN end]} | {[del] [alias]}
nat                 [stop] | {[LAN netif] [WAN netif]}
ftpdpath            [new path]

press ENTER to continue, 'Q' to quit.
```

```
[root@sylixos_station:/root]# help help
display help information.
help [-s keyword]
help [-s keyword]
[root@sylixos_station:/root]#
```

```
[root@sylixos_station:/root]# help l?
ln          [-s | -f] [actualpath] [sympath]
ll          [path name]
ls          [path name]
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The help command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdHelp (INT iArgC, PCHAR ppcArgV[]);

2.2.17. hostname - Display or set the current SylixOS image

host name

Format:

```
hostname
hostname name
```

Explanation:

There are usages of this command. When the hostname is not followed by a parameter, the current SylixOS image host name is displayed. If hostname is followed by a parameter, the Sylix image name is changed to this parameter.

返回值:

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# hostname
hostname is sylixos_station
[root@sylixos_station:/root]# hostname shell
[root@shell:/root]# hostname
hostname is shell
[root@shell:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The hostname command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdHostname (INT iArgC, PCHAR ppcArgV[]);

2.2.18. ints - View system interrupt vector table information

Format:

ints
ints cupidstart
ints cupidstartcupidend

Explanation:

This command is used to display all cpus' interrupt vector table information. Ints cupidstart displays interrupt vector table information with cpu id greater than or equal to cupidstart. Ints cupidstart cupidend shows interrupt vector table information with cpu id greater than or equal to cupidstart and less than or equal to cupidend.

Return value:

Return "0".

Remarks:

This virtual machine has only 1 CPU.

Example:

```
[root@sylixos_station:/root]# ints
interrupt vector show >>

  IRQ      NAME      ENTRY      CLEAR      PARAM      ENABLE RND PREEMPT      CPU 0
-----
   4 dm9000_isr    30002958        0 3062bb90  true                22
  14 tick_timer    30000a0c        0      0 true                48655
  15 uart2_isr     30006410        0 3062deb4 false                0
  17 dma0_isr      300012f8        0      0 true                0
  18 dma1_isr      300012f8        0      1 true                0
  19 dma2_isr      300012f8        0      2 true                0
  20 dma3_isr      300012f8        0      3 true                0
  23 uart1_isr     30006410        0 3062de30 false                0
  27 i2c_isr       3000167c        0 3062bcbc false                0
  28 uart0_isr     30006410        0 3062ddac true                 565
  31 touchscr      30005fb8        0      0 true                0

interrupt nesting show >>

CPU  MAX NESTING      IPI
-----
  0          1          0
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The ints command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdInts (INT iArgC, PCHAR ppcArgV[]);

2.2.19. kill - Send a signal to the specified thread or process.

The SIGKILL signal is sent by default.

Format:

```
kill tid/pid
```

```
kill -n signum tid/pid
```

There are two usages of this command, kill tid/pid, to send a SIGKILL(9) signal to a process or thread, and kill -n signum tid/pid, to send the signal represented by number signum to the process or thread.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

The process number is decimal and the thread number is hexadecimal.

Example:

```

[root@sylixos_station:/root]# ts
thread show >>

```

NAME	TID	PID	PRI	STAT	ERRNO	DELAY	PAGEFAILS	FPU	CPU
t_idle0	4010000	0	255	RDY	0	0	0	0	0
t_itimer	4010001	0	20	SLP	0	1	0	0	0
t_isrdefer	4010002	0	0	SEM	0	0	0	0	0
t_except	4010003	0	0	SEM	0	0	0	0	0
t_log	4010004	0	60	MSGQ	0	0	0	0	0
t_power	4010005	0	254	SLP	0	86	0	0	0
t_hotplug	4010006	0	250	SEM	506	86	0	0	0
t_reclaim	4010008	0	253	MSGQ	0	0	0	0	0
t_netjob	4010009	0	110	SEM	0	0	0	0	0
t_netproto	401000a	0	110	MSGQ	506	8	0	0	0
t_telnetd	401000d	0	160	MSGQ	2	0	0	0	0
t_xinput	401000e	0	199	SEM	506	79	0	0	0
t_touch	4010010	0	160	SEM	0	0	0	0	0
t_tshell	4010011	0	150	RDY	1503	0	0	0	0

```

thread: 14
[root@sylixos_station:/root]# kill 0x4010010
[root@sylixos_station:/root]# ts
thread show >>

```

NAME	TID	PID	PRI	STAT	ERRNO	DELAY	PAGEFAILS	FPU	CPU
t_idle0	4010000	0	255	RDY	0	0	0	0	0
t_itimer	4010001	0	20	SLP	0	1	0	0	0
t_isrdefer	4010002	0	0	SEM	0	0	0	0	0
t_except	4010003	0	0	SEM	0	0	0	0	0
t_log	4010004	0	60	MSGQ	0	0	0	0	0
t_power	4010005	0	254	SLP	0	61	0	0	0
t_hotplug	4010006	0	250	SEM	506	61	0	0	0
t_reclaim	4010008	0	253	MSGQ	0	0	0	0	0
t_netjob	4010009	0	110	SEM	0	0	0	0	0
t_netproto	401000a	0	110	MSGQ	506	3	0	0	0
t_telnetd	401000d	0	160	MSGQ	2	0	0	0	0
t_xinput	401000e	0	199	SEM	506	54	0	0	0
t_tshell	4010011	0	150	RDY	1503	0	0	0	0

```

thread: 13
[root@sylixos_station:/root]# █

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SIGNAL_EN > 0, the system is allowed to use the signal and this command will be included.

Function interface:

The kill command is implemented by a c language function. The function prototype is:
static INT __tshellSysCmdKill (INT iArgC, PCHAR ppcArgV[]);

2.2.20. login - Switch user, login again

Format:

login

Explanation:

This command is used to switch users. Entering login will prompt you to enter your login user name and password. If you pass the authentication, the user will be switched. Otherwise, you will still be logged in as the original user.

Return value:

Return "0".

Remarks:

The user to log in must be enabled. Otherwise, the login will fail.

Example:

```
[root@sylixos_station:/root]# login
login: liang
password:
[liang@sylixos_station:/root]$
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The login command is implemented by a c language function whose prototype is:
static INT __tshellSysCmdLogin (INT iArgC, PCHAR ppcArgV[]);

2.2.21. loglevel - Display or set the current kernel log (printk)

print level

Format:

loglevel
loglevel level

Explanation:

There are two usages of this command to display the print level of the current kernel log's printk interface. If there is a parameter, printk's print level is set to the corresponding level.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# loglevel
printk() log message current level is: 4
[root@sylixos_station:/root]# loglevel 5
[root@sylixos_station:/root]# loglevel
printk() log message current level is: 5
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_LOG_LIB_EN > 0, the system is allowed to provide a log management library and this command will be included.

Function interface:

The loglevel command is implemented by a c language function. The function

prototype is: static INT __tshellSysCmdLogLevel (INT iArgC, PCHAR ppcArgV[]);

2.2.22. mems - View operating system kernel memory heap and system memory heap memory usage

Format:

mems
mems rid

Explanation:

This command is used to view the system kernel memory heap and system memory heap usage.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# mems
heap show >>

  HEAP      TOTAL      USED      MAX USED  SEGMENT USED
-----
kersys      13279KB    3359KB    3376KB    1085  25%

[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The mems command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdMems (INT iArgC, PCHAR ppcArgV[]);

2.2.23. part - Display the specified memory partition information

Format:

partpartitionhandle

Explanation:

This command is used to display the specified memory partition information.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

Run the following program

```
#include<SylixOS.h>
typedef struct my_element {
    INT iValue;
} MY_ELEMENT;

#define ELEMENT_MAX    (8)
UINT8    _G_pucMyElementPool[sizeof(MY_ELEMENT) * ELEMENT_MAX];
LW_HANDLE _G_hMyPartition;
int main (int argc, char *argv[])
{
    _G_hMyPartition = Lw_Partition_Create("my_partition",
        _G_pucMyElementPool,
        ELEMENT_MAX,
        sizeof(MY_ELEMENT),
        LW_OPTION_DEFAULT,
        LW_NULL);
    if (_G_hMyPartition == LW_HANDLE_INVALID) {
        fprintf(stderr, "create partition failed.\n");
        return (-1);
    }
    printf("%x\n", _G_hMyPartition);
    while(1);
    return (0);
}
```

```
[root@sylixos:/apps/partition]# ./partition
4001000b
```

to get the memory partition handle.

```
[root@sylixos:/apps/partition]# part 4001000b
partition show >>

partition name      : my_partition
partition block number :      8
partition free block :      8
partition per block size :      4
[root@sylixos:/apps/partition]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The part command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdPart (INT iArgC, PCHAR ppcArgV[]);

2.2.24. pcis - Display PCI bus information

Format:

pcis

Explanation:

This command shows the PCI bus information.

Return value:

Return "0".

Remarks:

The PCI bus information is stored in the /proc/pci file.

X86 environment testing.

Example:

```
[root@sylixos:/]# pcis
PCI info:
Bus 0 Slot 0 Function 0 VendorID 8086 DeviceID 1237:
  Class 6 [Bridge] Sub 0 [Host bridge] Prog-if 0 []
  IRQ Line-0 Pin-0
  Latency=0 Min Gnt=0 Max lat=0
  Base0 00000000 Base1 00000000 Base2 00000000
  Base3 00000000 Base4 00000000 Base5 00000000
  Rom 0 SubVendorID 1af4 SubSystemID 1100
Bus 0 Slot 1 Function 0 VendorID 8086 DeviceID 7000:
  Class 6 [Bridge] Sub 1 [ISA bridge] Prog-if 0 []
  IRQ Line-0 Pin-0
  Latency=0 Min Gnt=0 Max lat=0
  Base0 00000000 Base1 00000000 Base2 00000000
  Base3 00000000 Base4 00000000 Base5 00000000
  Rom 0 SubVendorID 1af4 SubSystemID 1100
Bus 0 Slot 1 Function 1 VendorID 8086 DeviceID 7010:
  Class 1 [Mass storage controller] Sub 1 [IDE interface] Prog-if 128 []
  IRQ Line-0 Pin-0
  Latency=0 Min Gnt=0 Max lat=0
  Base0 00000000 Base1 00000000 Base2 00000000
  Base3 00000000 Base4 0000c101 Base5 00000000
  Rom 0 SubVendorID 1af4 SubSystemID 1100
  Region 4: I/O ports at c100 [Size 16] [flags 00040101]
Bus 0 Slot 1 Function 3 VendorID 8086 DeviceID 7113:
  Class 6 [Bridge] Sub 128 [Bridge] Prog-if 0 []
  IRQ Line-9 Pin-1
  Latency=0 Min Gnt=0 Max lat=0
  Base0 00000000 Base1 00000000 Base2 00000000
  Base3 00000000 Base4 00000000 Base5 00000000
  Rom 0 SubVendorID 1af4 SubSystemID 1100
  Irq start 9 end 9 [flags 00000400]
Bus 0 Slot 2 Function 0 VendorID 1013 DeviceID b8:
  Class 3 [Display controller] Sub 0 [VGA compatible controller] Prog-if 0 [VGA controller]
  IRQ Line-0 Pin-0
  Latency=0 Min Gnt=0 Max lat=0
  Base0 fc000008 Base1 febd0000 Base2 00000000
  Base3 00000000 Base4 00000000 Base5 00000000
  Rom febc0000 SubVendorID 1af4 SubSystemID 1100
  Region 0: Memory at fc000000 [Size 32M] [flags 00042208] (32-bit pre)
  Region 1: Memory at febd0000 [Size 4K] [flags 00040200] (32-bit non-pre)
  Region 6: Expansion ROM at febc0000 [Size 64K] [flags 00046200] (32-bit pre)
Bus 0 Slot 3 Function 0 VendorID 10ec DeviceID 8029:
  Class 2 [Network controller] Sub 0 [Ethernet controller] Prog-if 0 []
  IRQ Line-11 Pin-1
  Latency=0 Min Gnt=0 Max lat=0
  Base0 0000c001 Base1 00000000 Base2 00000000
  Base3 00000000 Base4 00000000 Base5 00000000
  Rom feb80000 SubVendorID 1af4 SubSystemID 1100
  Region 0: I/O ports at c000 [Size 256] [flags 00040101]
  Region 6: Expansion ROM at feb80000 [Size 256K] [flags 00046200] (32-bit pre)
  Irq start b end b [flags 00000400]
[root@sylixos:/]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The pcis command is implemented through c language functions. The function prototype is: static INT __tshellSysCmdPcis (INT iArgC, PCHAR ppcArgV[]);

2.2.25. ps - View information about all processes in the system**Format:**

ps

Explanation:

This command is used to display information about all processes in the system.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ps
```

NAME	FATHER	PID	GRP	MEMORY	UID	GID	USER
kernel	<orphan>	0	0	0KB	0	0	root
test1	<orphan>	1	1	232KB	0	0	root

```
total vprocess: 2
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MODULELOADER_EN> 0, module loading service needs to be provided and this command will be included.

Function interface:

The ps command is implemented by a c language function. The function prototype is: static INT __tshellVProcShow (INT iArgC, PCHAR ppcArgV[]);

2.2.26. reboot - Restart the computer**Format:**

reboot
reboot-n/-f

Explanation:

There are two usages of this command. When reboot is not followed by a parameter, the computer will be cold restarted. When reboot is followed by -n/-f, the computer will be restarted immediately. When reboot is followed by other parameters,

parameter error will be prompted.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# reboot
[reboot]Reboot...
kernel rebooting...
kernel rebooting down.
[reboot]Force reboot...
[reboot]Force reboot...
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The reboot command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdReboot (INT iArgC, PCHAR ppcArgV[]);

2.2.27. renice - Set the priority of the specified process

Format:

```
renice priority pid
renice priority-p pid
renice priority-g pgrp
renice priority -u user
```

Explanation:

There are 4 usages of this command,

renice priority pid	Adjust the priority of the pid process;
renice priority-p pid	Adjust the priority of the pid process;
renice priority-g pgrp	Adjust the priority of the process group;
renice priority -u user	Adjust the priority of the process user.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

Use ts to view the priority of the thread

test1	4010019	1	247	JOIN	0	0	17	0
pthread	401001a	1	247	RDY	0	0	0	0
pthread	401001b	1	247	RDY	0	0	0	0

Modify the priority of the process

```
[root@sylixos_station:/root]# renice 2 1
[root@sylixos_station:/root]#
```

Use ts again to view the priority of the thread

```
test1      4010019      1 249 JOIN      0      0      17      0
pthread    401001a      1 249 RDY       0      0      0      0
pthread    401001b      1 249 RDY       0      0      0      0
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The renice command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdRenice (INT iArgC, PCHAR ppcArgV[]);

2.2.28. restart - Restart thread

Format:

restart tidargument

Explanation:

This command is used to restart the thread.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

The thread id is hexadecimal.

Only kernel threads can be restarted.

Example:

Create a thread in the main function of bsp

```
static void * task0(void *arg)
{
    printf("hello task\n");
    while(1);
    return NULL;
}
```

Restart the virtual machine, give a thread id, and restart the thread

```
t_tshell    4010012      0 150 SEM       71      0      0      0
t_main      4010013      0 200 JOIN      71      0      0      0
t_tshell    4010014      0 150 RDY      1503     0      0      0
pthread     4010015      0 250 RDY      1503     0      0      0

thread: 20
[unknown@sylixos:/]# restart 4010015 0
[unknown@sylixos:/]# hello task
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_THREAD_RESTART_EN>0, the task is allowed to restart and the command will be included.

Function interface:

The restart command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdRestart (INT iArgC, PCHAR ppcArgV[]);

2.2.29. sem - Display signal information

Format:

sem semaphore handle

Explanation:

This command shows the information of the specified signal.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

Use tp to view signal handle

```
[root@sylixos_station:/root]# tp
thread pending show >>

-----
NAME          TID    PID  STAT  DELAY  PEND EVENT  OWNER
-----
t_itimer      4010001  0  SLP      3      0:
t_isrdefer    4010002  0  SEM      0  10010002:job_sync
t_except      4010003  0  SEM      0  10010003:job_sync
t_log         4010004  0  MSGQ     0  1c010004:log_msg:R
t_power       4010005  0  SLP     97      0:
t_hotplug     4010006  0  SEM     97  10010011:job_sync
t_reclaim     4010008  0  MSGQ     0  1c01001b:res_reclaim:R
t_netjob      4010009  0  SEM      0  1001005a:job_sync
t_netproto    401000a  0  MSGQ     2  1c010060:lwip_msg:R
t_tftpd       401000b  0  MSGQ     0  1c01007e:lwip_msg:R
t_ftpd        401000c  0  MSGQ     0  1c010082:lwip_msg:R
t_telnetd     401000d  0  MSGQ     0  1c010085:lwip_msg:R

pending thread: 12
[root@sylixos_station:/root]#

[root@sylixos_station:/root]# sem 1001005a
Semaphore show >>

Semaphore Name      : job_sync
Semaphore Id        : 0x1001005a
Semaphore Type      : BINARY
Thread Queuing      : FIFO
Pended Threads      : 1
Semaphore Value      : EMPTY

[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When

LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The sem command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdSem (INT iArgC, PCHAR ppcArgV[]);

2.2.30. shell - Create a shell using ttydevice as the standard file

Format:

shell tty devicenologin

Explanation:

Create a shell with a tty device as the standard file.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos:/root]# shell /dev/ttyS0
[root@sylixos:/root]# login: █
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The shell command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdShell (INT iArgC, PCHAR ppcArgV[]);

2.2.31. shstack - Display or set the shell task stack size

Format:

shstack

shstack new stack size

Explanation:

There are two usages of this command, and the default shell task stack size is displayed when there are no parameters. The stack size of the new shell task is set if the command is followed by a parameter.

Return value:

Return "0".

Remarks:

Settings are only valid for shells that are started later.

Example:

```
[root@sylixos_station:/root]# shstack
default shell stack: 32768
[root@sylixos_station:/root]# shstack 40000
default shell stack: 40000
[root@sylixos_station:/root]# shstack
default shell stack: 40000
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The shstack command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdShStack (INT iArgC, PCHAR ppcArgV[]);

2.2.32. shutdown - shut down or restart the system

Format:

```
shutdown
shutdown -r
shutdown -h
shutdown -f
```

Explanation:

There are four usages of this command, among which shutdown and shutdown -h are functionally the same, both shutting down and then restarting the system; shutdown -r shuts down and then cold start the operating system; shutdown -f restarts the system immediately.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# shutdown
[shutdown]Shutdown...
kernel rebooting...
kernel rebooting down.
```

```
[root@sylixos_station:/root]# shutdown -r
[shutdown]Shutdown & reboot...
kernel rebooting...
kernel rebooting down.
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The shutdown command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdShutdown (INT iArgC, PCHAR ppcArgV[]);

2.2.33. sigqueue - Send a signal to a thread or process**Format:**

```
sigqueue -n signum tid/pid
sigqueue  tid/pid
```

Explanation:

There are two usages of this command. Sigqueue tid/pid sends a SIGKILL(9) signal to a process or thread; sigqueue -n signum tid/pid sends the signal represented by number signum to the process or thread.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

The process number is decimal and the thread number is hexadecimal.

Example:

View thread information in the specified process

```
[root@sylixos_station:/apps/test1]# ts 3
thread show >>

      NAME      TID    PID  PRI  STAT  ERRNO    DELAY    PAGEFAILS  FPU  CPU
-----
test1          401001a    3 200  JOIN      0         0         17         0
pthread        401001b    3 245  RDY       0         0          0         0
pthread        401001c    3 243  RDY       0         0          0         0

thread: 3
[root@sylixos_station:/apps/test1]#
```

Send the message and view the thread information again

```
[root@sylixos_station:/apps/test1]# sigqueue 3
[root@sylixos_station:/apps/test1]# ts 3
thread show >>

      NAME      TID    PID  PRI  STAT  ERRNO    DELAY    PAGEFAILS  FPU  CPU
-----
thread: 0
[root@sylixos_station:/apps/test1]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SIGNAL_EN >0, the system is allowed to use the signal and this command will be included.

Function interface:

The sigqueue command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdSigqueue (INT iArgC, PCHAR ppcArgV[]);

2.2.34. sleep - Current thread sleep time specified

Format:

sleep n
sleep ns
sleep nm
sleep nh
sleep nd

Explanation:

There are five usages of this command, sleep n and sleep ns both sleep n seconds; sleep nm sleeps n minutes; sleep nh sleeps n hours, sleep nd sleeps n days.

Return value:

Return “-1” on failure, and return “0” if the execution is a success and the specified sleep time is due, otherwise it returns the remaining seconds.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# sleep 2s  
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The sleep command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdSleep (INT iArgC, PCHAR ppcArgV[]);

2.2.35. sprio - set the priority of the specified thread

Format:

sprio prioritythread_id

Explanation:

This command is used to set the priority of the specified thread.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

Thread_id is 16 processes.

Example:

Use the ts command to view the priority of an existing process

test1	401001a	2	200	JOIN	0	0	17	0
pthread	401001b	2	245	RDY	0	0	0	0
pthread	401001c	2	243	RDY	0	0	0	0

Use sprio to change the priority of a thread

```
[root@sylixos_station:/root]# sprio 240 4010015
[root@sylixos_station:/root]#
```

Use ts command again to see if the priority of the corresponding process has been changed

test1	4010013	1	200	JOIN	0	0	17	0
pthread	4010014	1	245	RDY	0	0	0	0
pthread	4010015	1	240	RDY	0	0	0	0

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The sprio command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdSprio (INT iArgC, PCHAR ppcArgV[]);

2.2.36.ss - View all thread and interrupt system stack usage in the system

Format:

ss

Explanation:

This command is used to view the use of threads and interrupt stacks.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ss
thread stack usage show >>

      NAME      TID    PRI  STK USE   STK FREE  USED
-----
t_idle0        4010000 255      356    3740    8%
t_itimer        4010001 20      804    3292   19%
t_isrdefer      4010002 0       432    3664   10%
t_except        4010003 0       496    3600   12%
t_log           4010004 60      520    3576   12%
t_power         4010005 254     352    3744    8%
t_hotplug       4010006 250     432    7760    5%
t_reclaim       4010008 253     384   16000    2%
t_netjob        4010009 110     416    3680   10%
t_netproto      401000a 110    1040    3056   25%
t_tftpd         401000b 160    3136    5056   38%
t_ftpd          401000c 160    2080   10208   16%
t_telnetd       401000d 160    2072    4072   33%
t_tshell        401000f 150    4096   28672   12%

interrupt stack usage show >>

CPU STK USE   STK FREE USED
---
0    388    3708   9%
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The cpuus command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdSs (INT iArgC, PCHAR ppcArgV[]);

2.2.37. top - View cpu usage rate

Format:

```
top
top -n times -t wait_seconds
```

Explanation:

There are two usages of this command, where -n is the number of times the cpu utilization is detected and -t is the time spent in each check. The cpuus detects one time by default, the detection time is 1s, top -n times -t wait_seconds can specify the times of detection, and the time of each detection is wait_seconds seconds. The specified inspection time must not exceed 10 seconds.

Return value:

Return "0".
N/A.

Example:

```
[root@sylixos_station:/root]# top
CPU usage checking, please wait...
CPU usage show (measurement accuracy 1.0%) >>

      NAME      TID  PRI   CPU   KERN
-----
t_tshell      401000f 150   1.0%   0.0%
t_telnetd     401000d 160   0.0%   0.0%
t_ftpd        401000c 160   0.0%   0.0%
t_tftpd       401000b 160   0.0%   0.0%
t_netproto    401000a 110   0.0%   0.0%
t_netjob      4010009 110   0.0%   0.0%
t_reclaim     4010008 253   0.0%   0.0%
t_hotplug     4010006 250   0.0%   0.0%
t_power       4010005 254   0.0%   0.0%
t_log         4010004  60   0.0%   0.0%
t_except      4010003  0    0.0%   0.0%
t_isrdefer    4010002  0    0.0%   0.0%
t_itimer      4010001 20    0.0%   0.0%
t_idle0       4010000 255  98.0%   0.0%

[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The top command is implemented via a c language function. The function prototype is:
static INT __tshellSysCmdCpuus (INT iArgC, PCHAR ppcArgV[]);

2.2.38. tp - View blocked thread information in the system

Format:

tp
tp pid

Explanation:

There are two usages of this command. When tp is not followed by a parameter, it will show blocked thread information in all processes; when tp is followed by a parameter, it will show the blocked thread information in the corresponding process.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# tp 1
thread pending show >>

      NAME      TID  PID  STAT   DELAY      PEND EVENT      OWNER
-----
test1      4010013    1  JOIN        0  4010014:pthread
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The tp command is implemented by a c language function. The function prototype is:
static INT __tshellSysCmdTp (INT iArgC, PCHAR ppcArgV[]);

2.2.39. ts - View system thread information**Format:**

ts
ts pid

Explanation:

There are 2 usages of this command. When ts is not followed by a parameter, it will show the thread information in all processes; when ts is followed by a parameter, it will show the thread information in the corresponding process.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ts
thread show >>
```

NAME	TID	PID	PRI	STAT	ERRNO	DELAY	PAGEFAILS	FPU	CPU
t_idle0	4010000	0	255	RDY	0	0	0	0	0
t_itimer	4010001	0	20	SLP	0	4	0	0	0
t_isrdefer	4010002	0	0	SEM	0	0	0	0	0
t_except	4010003	0	0	SEM	0	0	0	0	0
t_log	4010004	0	60	MSGQ	0	0	0	0	0
t_power	4010005	0	254	RDY	0	0	0	0	0
t_hotplug	4010006	0	250	RDY	506	0	0	0	0
t_reclaim	4010008	0	253	MSGQ	0	0	0	0	0
t_netjob	4010009	0	110	SEM	0	0	0	0	0
t_netproto	401000a	0	110	MSGQ	506	5	0	0	0
t_tftpd	401000b	0	160	MSGQ	2	0	0	0	0
t_ftpd	401000c	0	160	MSGQ	2	0	0	0	0
t_telnetd	401000d	0	160	MSGQ	2	0	0	0	0
t_tshell	401000f	0	150	RDY	1503	0	0	0	0
t_ptyserver	4010010	0	160	JOIN	0	0	0	0	0
t_ptyproc	4010011	0	150	RDY	1503	0	0	0	0
t_tshell	4010012	0	150	JOIN	71	0	0	0	0
test1	4010013	1	200	JOIN	0	0	17	0	0
pthread	4010014	1	245	RDY	0	0	0	0	0
pthread	4010015	1	243	RDY	0	0	0	0	0

```
thread: 20
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands,

and this command is included.

Function interface:

The ts command is implemented by a c language function. The function prototype is:
static INT __tshellSysCmdTs (INT iArgC, PCHAR ppcArgV[]);

2.2.40. tty - Display the tty file corresponding to the current shell terminal

Format:

tty

Explanation:

This command is used to display the tty file corresponding to the current shell terminal.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# tty
/dev/ttyS0
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The tty command is implemented by a c language function. The function prototype is:
static INT __tshellSysCmdTty (INT iArgC, PCHAR ppcArgV[]);

2.2.41. vardel- Delete a specified system environment variable

Format:

vardel variable

Explanation:

This command is used to delete the specified system environment variable.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

View the environment variables

```
[root@sylixos_station:/root]# vars
variable show >>

```

VARIABLE	REF	VALUE
MYPATH		/home/
TERMCAP		/etc/termcap
TERM		vt100
LUA_CPATH		?..so;/usr/local/lib/lua/?..so;/usr/lib/lua/?..so;/lib/lua/?..so

Delete environment variables and view again

```
[root@sylixos_station:/root]# vardel MYPATH
[root@sylixos_station:/root]# vars
variable show >>

```

VARIABLE	REF	VALUE
TERMCAP		/etc/termcap
TERM		vt100
LUA_CPATH		?..so;/usr/local/lib/lua/?..so;/usr/lib/lua/?..so;/lib/lua/?..so

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The vardel command is implemented via a c language function. The function prototype is: static INT __tshellSysCmdVardel (INT iArgC, PCHAR ppcArgV[]);

2.2.42. varload - Extract the loading environment variable table

from the file with the specified parameters

Format:

```
varload
varload profile
```

Explanation:

There are two usages of this command. Without a parameter, the environment variables are extracted from the /etc/profile file by default. With a parameter, the environment variables are extracted from the specified file.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

See the current environment variable

```
[root@sylixos_station:/root]# vars
variable show >>

```

VARIABLE	REF	VALUE
TERMCAP		/etc/termcap
TERM		vt100
LUA_CPATH		?..so;/usr/local/lib/lua/?..so;/usr/lib/lua/?..so;/lib/lua/?..so
LUA_PATH		?..lua;/usr/local/lib/lua/?..lua;/usr/lib/lua/?..lua;/lib/lua/?..lua
DEBUG_CPU		-1
PATH_LOCALE		/usr/share/locale
LC_ALL		
LANG		C
LD_LIBRARY_PATH		/usr/lib:/lib:/usr/local/lib
PATH		/usr/bin:/bin:/usr/pkg/sbin:/sbin:/usr/local/bin
NFS_CLIENT_PROTO		udp
NFS_CLIENT_AUTH		AUTH_UNIX
SYSDLOGD_HOST		0.0.0.0:514
FIO_FLOAT		1
SO_MEM_PAGES		8192
TSLIB_CALIBFILE		/etc/pointercal
TSLIB_TSDEVICE		/dev/input/touch0
MOUSE		/dev/input/mouse0:/dev/input/touch0
KEYBOARD		/dev/input/keyboard0
TZ		CST-8:00:00
TMPPDIR		/tmp/
LICENSE		SylixOS license: Commercial & GPL.
VERSION		1.3.5 (5)
SYSTEM		SylixOS kernel version: 1.3.5 (5) LongYuan(b)

```
[root@sylixos_station:/root]#
```

Create a path file, write MYPATH="/home/" in the file and save it.

```
[root@sylixos_station:/root]# vi path
MYPATH="/home/"
~
```

Load the command

```
[root@sylixos_station:/root]# varload path
environment variables load from path success.
```

Check the environment variables again

VARIABLE	REF	VALUE
MYPATH		/home/
TERMCAP		/etc/termcap

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The varload command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdVarload (INT iArgC, PCHAR ppcArgV[]);

2.2.43. vars - Display the current environment variable

Format:

vars

Explanation:

This command is used to view the current environment variable.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# vars
variable show >>

-----
VARIABLE      REF      VALUE
-----
TERMCAP        /etc/termcap
TERM           vt100
LUA_CPATH      ?.so;/usr/local/lib/lua/?.so;/usr/lib/lua/?.so;/lib/lua/?.so
LUA_PATH       ?.lua;/usr/local/lib/lua/?.lua;/usr/lib/lua/?.lua;/lib/lua/?.lua
DEBUG_CPU      -1
PATH_LOCALE    /usr/share/locale
LC_ALL         C
LANG           C
LD_LIBRARY_PATH /usr/lib:/lib:/usr/local/lib
PATH           /usr/bin:/bin:/usr/pkg/sbin:/sbin:/usr/local/bin
NFS_CLIENT_PROTO udp
NFS_CLIENT_AUTH AUTH_UNIX
SYSLOGD_HOST   0.0.0.0:514
FIO_FLOAT      1
SO_MEM_PAGES   8192
TSLIB_CALIBFILE /etc/pointercal
TSLIB_TSDEVICE /dev/input/touch0
MOUSE          /dev/input/mouse0:/dev/input/touch0
KEYBOARD       /dev/input/keyboard0
TZ             CST-8:00:00
TMPDIR         /tmp/
LICENSE        SylixOS license: Commercial & GPL.
VERSION        1.3.5 (5)
SYSTEM         SylixOS kernel version: 1.3.5 (5) LongYuan(b)
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The vars command is implemented via c language functions. The function prototype is: static INT __tshellSysCmdVars (INT iArgC, PCHAR ppcArgV[]);

2.2.44. varsave – save the current operating system environment variables table

Format:

varsave
varsave rofile

Explanation:

There are two usages of this command. When varsave is followed by a parameter, it saves the current environment variables in the /etc/profile file by default; if varsave is

followed by a parameter, it saves the environment variables in the corresponding file.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

If the profile file does not exist, varsave creates the file first, and if the profile file exists, varsave modifies the value of the environment variable response.

Example:

```
[root@sylixos_station:/root]# varsave path
envionment variables save to path success.
```

View the path file

```
[root@sylixos_station:/root]# cat path
#sylixos envionment variables profile.
TERMCAP="/etc/termcap"
TERM="vt100"
LUA_CPATH="? .so;/usr/local/lib/lua/? .so;/usr/lib/lua/? .so;/lib/lua/? .so"
LUA_PATH="? .lua;/usr/local/lib/lua/? .lua;/usr/lib/lua/? .lua;/lib/lua/? .lua"
DEBUG_CPU="-1"
PATH_LOCALE="/usr/share/locale"
LC_ALL=""
LANG="C"
LD_LIBRARY_PATH="/usr/lib:/lib:/usr/local/lib"
PATH="/usr/bin:/bin:/usr/pkg/sbin:/sbin:/usr/local/bin"
NFS_CLIENT_PROTO="udp"
NFS_CLIENT_AUTH="AUTH_UNIX"
SYSLOGD_HOST="0.0.0.0:514"
SO_MEM_PAGES="8192"
TSLIB_CALIBFILE="/etc/pointercal"
TSLIB_TSDEVICE="/dev/input/touch0"
MOUSE="/dev/input/mouse0:/dev/input/touch0"
KEYBOARD="/dev/input/keyboard0"
TZ="CST-8:00:00"
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The varsave command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdVarsave (INT iArgC, PCHAR ppcArgV[]);

2.2.45. virtuals - Display vmm virtual memory information

Format:

virtuals

Explanation:

This command is used to display virtual memory information.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos:/root]# virtuals
vmm virtual area show >>
vmm virtual program from: 0x60000000, size: 0x80000000
vmm virtual ioremap from: 0xe0000000, size: 0x10000000
vmm virtual area usage as follow:

VIRTUAL      SIZE      WRITE  CACHE
-----
60006000      1000    true   true
60007000      7000    true   true
6000e000      1000    true   true
6000f000      3000    true   true
e0000000      1000    true   false
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_VMM_EN>0, virtual memory is supported, and this command will be included.

Function interface:

The virtuals command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdVirtuals (INT iArgC, PCHAR ppcArgV[]);

2.2.46. which - Check the file location specified by the parameter

Format:

which programfile

Explanation:

该命令在环境变量指定的路径下查找指定的文件所在的位置，并显示文件所在的路径。

This command looks for the location of the specified file in the path specified by the environment variable and displays the path where the file is located.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/]# which myusr
/usr/bin/myusr
[root@sylixos_station:/]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MODULELOADER_EN>0, module loading service needs to be provided and this command will be included.

Function interface:

The which command is implemented by a c language function. The function prototype is: static INT __tshellWhich (INT iArgC, PCHAR *ppcArgV);

2.2.47. who - View current login user identity

Format:

who

Explanation:

This command is used to view the current login user information.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# who
user:root terminal:/dev/ttyS0 uid:0 gid:0 euid:0 egid:0
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command is included.

Function interface:

The who command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdWho (INT iArgC, PCHAR ppcArgV[]);

2.2.48. zones - Check operating system physical page partition

usage

Format:

zones

Explanation:

This command is used to display the vmm physical memory information.

Return value:

Return "0".

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# zones
vmm physical zone show >>

ZONE PHYSICAL    SIZE    PAGESIZE    PGD    FREEPAGE    DMA    USED
-----
  0 31800000    600000    1000 30bcc000    1536 true    0%
  1 31e00000    2200000    1000 30bcc000    8704 false   0%

ALL-Physical memory size: 64 MBytes (67108864 Bytes)
VMM-Physical memory size: 40 MBytes (41943040 Bytes)
VMM-Physical memory free: 40 MBytes (41943040 Bytes)
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_VMM_EN>0, virtual memory is supported, and this command will be included.

Function interface:

The zones command is implemented by a c language function. The function prototype is: static INT __tshellSysCmdZones (INT iArgC, PCHAR ppcArgV[]);

3. File command

3.1. Introduction

The file-related commands are:

- cat – Display contents of file.
- Cd – Switch the current directory.
- ch – Change the directory.
- chmod - Change the access permissions of a file or directory.
- close –Close a file.
- cmp - Compare a file.
- Cp - Copy a file.
- df - View the file system information in the specified directory.
- dosfslabel – View or set file system label.
- dsize - Calculate all file information contained in a specified directory.
- fatugid – Set fat file system user and group id.
- fdentrys –List all file information that operating system is opeating.
- fdisk - Display or make a disk partition table.

Files - List all file information that operating system is opeating.

gzip - Compress or decompress a file.

Ll - Display the detailed information in the specified directory, with the default current directory.

Ln - Create a symbolic link file.

- logfileadd - Add kernel file descriptor to kernel log print function.
- Logfileclear -Clear the specified kernel file descriptor from the kernel log print file table.
- Display the list of kernel log print files.
- ls - Display the file name in the specified directory, with the default current directory.
- mkdir -Create directory.

mkfifo - Create a named pipe that can only be created under the root file system device.

- mkfs –Format the specified disk.
- mmaps - Display the system mmap information.
- mount - Mount a volume.
- msgq – Display message queue information.
- mv –Move to or rename a file.
- open - Open a file.
- pwd –View the current working directory.
- rm - Delete a file.
- Rmdir - Delete a folder.
- showmount –View all mounted volumes in the system.

Shfile - Execute the specified shell script.

- sync - Write all system cached information to physical devices.
- tmpname - Get a temporary file name that can be created.
- touch - Create a common file.
- umount - Unload a volume.
- untar -Unpack or untar a tar or tar.gz package.
- vi - Start the vi editor.
- yaffscmd - Display, set, and erase a block.
- Zlib - Add a .gz compressed file.

3.2. Use of commands

The detailed description and usage of file-related shell commands include the format, description, remarks, usage examples, and tailorable configuration and response interfaces.

3.2.1. cat- Display contents of the file

Format:

cat filename

Explanation:

This command is used to display contents of the file, where filename can contain the path to the file.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# cat /proc/kernel/affinity
      NAME      TID    PID   CPU
-----
t_idle0        4010000    0    0
t_itimer       4010001    0    *
t_isrdefer     4010002    0    *
t_except       4010003    0    *
t_log          4010004    0    *
t_power        4010005    0    *
t_hotplug      4010006    0    *
t_reclaim      4010008    0    *
t_netjob       4010009    0    *
t_netproto     401000a    0    *
t_tftpd        401000b    0    *
t_ftpd         401000c    0    *
t_telnetd      401000d    0    *
t_tshell       401000f    0    *

[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell

commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The cat command is implemented through a c language function. The function prototype is:

```
Static INT __tshellFsCmdCat (INT iArgC, PCHAR ppcArgV[]);
```

3.2.2. cd- Switch the current directory

Format:

cd path

Explanation:

This command is used to switch to the specified directory.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# cd /
[root@sylixos_station:/]# cd /root/
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The cd command is implemented by a c language function. The function prototype is:

```
Static INT __tshellFsCmdCd (INT iArgC, PCHAR ppcArgV[]);
```

3.2.3. ch- Change the directory

Format:

ch dir

Explanation:

This command is used to switch to the specified directory.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/]# ch /lib/modules/
[root@sylixos_station:/lib/modules]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell

commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The ch command is implemented by a c language function. The function prototype is:
Static INT __tshellFsCmdCh (INT iArgC, PCHAR ppcArgV[]);

3.2.4. chmod - Change the access permissions of a file or directory.

Format:

chmod newmode filename

Explanation:

This command is used to change the access permissions of a file or directory.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ll
-rw-rw-rw- root      root      Thu Nov 17 13:38:35 2016      16 B, a
-rw-r--r-- root      root      Thu Nov 17 13:57:37 2016      653 B, path
total items: 2
[root@sylixos_station:/root]# chmod 777 a
[root@sylixos_station:/root]# chmod 666 path
[root@sylixos_station:/root]# ll
-rwxrwxrwx root      root      Thu Nov 17 13:38:35 2016      16 B, a
-rw-rw-rw- root      root      Thu Nov 17 13:57:37 2016      653 B, path
total items: 2
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The chmod command is implemented via c language functions. The function prototype is:

Static INT __tshellFsCmdChmod (INT iArgC, PCHAR ppcArgV[]);

3.2.5. close – Close a file

Format:

close fd

Explanation:

This command is used to close an open file.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# open a 2
open file return: 7 dev 3062d1d4 inode 303 size 0
[root@sylixos_station:/root]# close 7
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell command.

Function interface:

The close command is implemented by a c language function. The function prototype is:

Static INT __tshellSysCmdClose (INT iArgC, PCHAR ppcArgV[]);

3.2.6. cmp-Compare a file

Format:

cmp file one file two

Explanation:

This command is used to compare whether the two files are the same.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ls
c      b      a
[root@sylixos_station:/root]# cmp a b
file same!
[root@sylixos_station:/root]# cmp a c
file not same!
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The cmp command is implemented by a c language function. The function prototype is:

Static INT __tshellFsCmdCmp (INT iArgC, PCHAR ppcArgV[]);

3.2.7. cp- Copy a file

Format:

cp src file name dst file name

cp src file name dst file name

Explanation:

This command is used to copy the contents of the src file into the dst file.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

If dst file does not exist, it will be created first in the copy. If dst file exists, it will ask whether to overwrite dst file. If you choose not to overwrite, the copy will fail.

Example:

```
[root@sylixos_station:/root]# cp a c
copy complete. size:653(Bytes) time:0(s) speed:653(Bps)
[root@sylixos_station:/root]# cp c a
destination file is exist, overwrite? (Y/N)
N
[root@sylixos_station:/root]# cp c a
destination file is exist, overwrite? (Y/N)
Y
copy complete. size:653(Bytes) time:0(s) speed:653(Bps)
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The cp command is implemented by a c language function. The function prototype is: Static INT __tshellFsCmdCp (INT iArgC, PCHAR ppcArgV[]);

3.2.8. df- View the file system information in the specified directory

Format:

df volume name

Explanation:

This command checks the file system information in the specified directory.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# df /
  VOLUME      TOTAL      FREE   USED RO      FS TYPE
-----
/          3.09KB      0.00KB 100% n  ROOT FileSystem
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The df command is implemented through c language functions. The function prototype is:

Static INT __tshellFsCmdDf (INT iArgC, PCHAR ppcArgV[]);

3.2.9. dosfslabel – View or set file system label

Format:

dosfslabel vol
dosfslabel vol vol newlabel

Explanation:

There are only 2 usages of this command, with the parameter vol used to view the system of the corresponding label.

Return value:

Return 0 if the command is executed successfully and return -1 if the command fails.

Remarks:

N/A.

Example:

```
[root@sylixos:/media]# dosfslabel hdd0 hdd  
[root@sylixos:/media]# dosfslabel hdd0  
HDD  
[root@sylixos:/media]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0) and LW_CFG_FATFS_EN> 0, the number of supported volumes> 0 and the small FAT file system is allowed, this command will be included.

Function interface:

The dosfslabel command is implemented by a c language function. The function prototype is:

Static INT __tshellFsCmdDosfslabel (INT iArgC, PCHAR ppcArgV[])

3.2.10. dsize - Calculate all file information contained in a specified directory

Format:

dsize pathname

Explanation:

This command is used to calculate the information contained in the specified directory, including the number of files and the total size of the file.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# dsize /
scanning...
total file 104 size 779250 bytes.
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The dsize command is implemented by a c language function. The function prototype is:

Static INT __tshellFsCmdDsize (INT iArgC, PCHAR ppcArgV[]);

3.2.4.fatugid – Set fat file system user and group id

Format:

Fatugiduidgid

Explanation:

This command is used to set the fat file system user and group id

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# fatugid 0 0
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0) and LW_CFG_FATFS_EN> 0, the number of supported volumes> 0 and the small FAT file system is allowed, this command will be included.

Function interface:

The fatugid command is implemented by a c language function. The function prototype is:

Static INT __tshellFatUGID (INT iArgC, PCHAR ppcArgV[]);

3.2.12. fdentrys – List all file information that operating system is operating

Format:

Fdentrys

Explanation:

This command is used to display all operating system file information.

Return value:

Return 0 if the command is executed successfully.

Remarks:

All file information includes files opened by the process.

Example:

```
[root@sylixos_station:/root]# fdentrys
all file entry show >>
ref abn name                      real                      type    drv
1    /dev/socket                  /dev/socket             socket   31
1    /dev/socket                  /dev/socket             socket   31
1    /dev/socket                  /dev/socket             socket   31
1    /dev/ttyS0                   /dev/ttyS0              orig     17
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The fdentrys command is implemented through c language functions. The function prototype is:

Static INT __tshellSysCmdFdentrys (INT iArgC, PCHAR ppcArgV[]);

3.2.13. fdisk- Display or make a disk partition table

Format:

fdisk block I/O device

fdisk -fblock I/O device

Explanation:

There are only two usages of this command. When not followed by -f, the current partition table is displayed, and the -f option is used to create a disk partition table.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

X86 environment testing.

To make a disk partition table, first determine that the disk is not mounted. Use showmount to see if it is mounted. If mounted, use umount to perform unmounting.

After the partition is completed, it is necessary to restart and initialize each partition.

Example:

```
[root@sylixos:/root]# fdisk /dev/blk/hdd-0
block device: /dev/blk/hdd-0 partition >>

PART  ACT  SIZE (KB)  OFFSET (KB)  TYPE
-----
0    *      261120      1024  SylixOS True Power Safe Partition

total partition 1
[root@sylixos:/root]#
```

```

[root@sylixos:/root]# fdisk -f /dev/blk/hdd-0
block device /dev/blk/hdd-0 total size: 256 (MB)
please input how many partition(s) you want to make (1 ~ 4) : 2
please input how many bytes align (4K 8K ...) : 4096
please input the partition 0 size percentage(%) 0 means all left space : 20
is this partition active(y/n) : y
please input the file system type
1: FAT 2: TPSFS 3: LINUX 4: RESERVED
1
please input the partition 1 size percentage(%) 0 means all left space : 0
is this partition active(y/n) : n
please input the file system type
1: FAT 2: TPSFS 3: LINUX 4: RESERVED
2
making partition...
block device: /dev/blk/hdd-0 partition >>

PART ACT  SIZE(KB)  OFFSET(KB)  TYPE
-----
0 *      52224     1024 Win95 FAT32 Partition
1       208896    53248 SylixOS True Power Safe Partition

total partition 2
[root@sylixos:/root]# █

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0) and LW_CFG_OEMDISK_EN> 0, the number of supported volumes >0 and the automount and partition tools are required, this command will be included.

Function interface:

The fdisk command is implemented through c language functions. The function prototype is:

Static INT __tshellFsCmdFdisk (INT iArgC, PCHAR ppcArgV[]);

3.2.14. files-List all file information that operating system is operating

Format:

Files

Explanation:

This command is used to display the file information that the operating system is operating.

Return value:

Return 0 if the command is executed successfully.

Remarks:

This command will display the kernel file without displaying the file opened by the thread.

Example:

```
[root@sylixos_station:/dev/blk]# files
kernel filedes show (process filedes in /proc/${pid}/filedes) >>
fd abn name                type   drv
 3   /dev/ttyS0             orig   17 GLB STD_IN GLB STD_OUT GLB STD_ERR
 4   /dev/socket            socket 31
 5   /dev/socket            socket 31
 6   /dev/socket            socket 31
[root@sylixos_station:/dev/blk]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The ch command is implemented by a c language function. The function prototype is:
Static INT __tshellSysCmdFdentrys (INT iArgC, PCHAR ppcArgV[]);

3.2.15. gzip - Compress or decompress a file

Format:

```
gzip [-c] [-d] [-f] [-h] [-r] [-1 to -9] [files...]
```

Explanation:

This command is used to compress or decompress a file, where the parameters are:

- c Write output to standard output and retain the original file;
- d Extract the compressed file;
- f Huffman encoding combined with string matching;
Use only Huffman compression algorithm to compress files;
- r Use RLE compression algorithm to compress files;
- 1 to -9 Specify the level of compression.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A

Example:

```
[root@sylixos_station:/root]# ll
-rw-r--r-- root    root    Thu Nov 24 12:18:05 2016    1251 B, b
-rw-r--r-- root    root    Thu Nov 24 12:17:57 2016    1251 B, a
total items: 2
[root@sylixos_station:/root]# gzip -r -3 a
[root@sylixos_station:/root]# gzip -f -3 b
[root@sylixos_station:/root]# ll
-rw-rw-rw- root    root    Thu Nov 24 12:18:38 2016    449 B, b.gz
-rw-rw-rw- root    root    Thu Nov 24 12:18:29 2016    697 B, a.gz
total items: 2
[root@sylixos_station:/root]# gzip -d a
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The gzip command is implemented by a c language function. The function prototype is:
Int minigzip_main(int argc, char *argv[]);

3.2.16. ll - Display the detailed information in the specified directory

Format:

ll [path name]

Explanation:

This command is used to display the detailed information of the file in the specified directory. When no parameter is specified, the file information in the current directory is displayed at default. When the parameter path is followed, the file information in the corresponding directory is displayed.

Return valve:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/]# ll
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    tmp -> /yaffs2/n1/tmp
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    var -> /yaffs2/n1/var
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    root -> /yaffs2/n1/root
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    home -> /yaffs2/n1/home
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    apps -> /yaffs2/n1/apps
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    sbin -> /yaffs2/n1/sbin
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    bin -> /yaffs2/n1/bin
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    usr -> /yaffs2/n1/usr
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    lib -> /yaffs2/n1/lib
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    qt -> /yaffs2/n1/qt
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    ftk -> /yaffs2/n1/ftk
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    etc -> /yaffs2/n0/etc
lrwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    boot -> /yaffs2/n0/boot
drwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    usb/
drw-rw-rw- root    root    Fri Nov 18 10:18:12 2016    yaffs2/
drw-r--r-- root    root    Fri Nov 18 10:18:12 2016    proc/
drwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    media/
drwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    mnt/
drwxr-xr-- root    root    Fri Nov 18 10:18:12 2016    dev/
total items: 19
[root@sylixos_station:/]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The ll command is implemented by a c language function. The function prototype is:

Static INT __tshellFsCmdLI (INT iArgC, PCHAR ppcArgV[]);

Remarks:

N/A.

3.2.17. ln – Create a symbolic link file

Format:

ln [-s | -f] actualpath sympathy

Explanation:

This command is used to create a symbolic link file, with the -s parameter at default. Actualpath is the actual path, and symoath is the link file name.

Return valve:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/]# ln -s /yaffs2/n0/lost+found/ lost
[root@sylixos_station:/]# ll
lrw-rw-rw- root      root      Thu Nov 24 13:38:54 2016      lost -> /yaffs2/n0/lost+found/
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The ln command is implemented by a c language function. The function prototype is:
Static INT __tshellFsCmdLn (INT iArgC, PCHAR ppcArgV[]);

3.2.18. logfileadd - Add kernel file descriptor to kernel log print

function

Format:

logfileadd file descriptor

Explanation:

This command is used to add the specified kernel file descriptor to the kernel log function.

Return valve:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# logfileadd 7
[root@sylixos_station:/root]# logfiles
log fd(s) include :
      1      7
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_LOG_LIB_EN> 0, the system is allowed to provide a log management library and this command will be included.

Function interface:

The logfileadd command is implemented through c language functions. The function prototype is:

Static INT __tshellSysCmdLogFileAdd (INT iArgC, PCHAR ppcArgV[]);

3.2.19. Logfileclear -Clear the specified kernel file descriptor from the kernel log print file table

Format:

logfileclear file descriptor

Explanation:

This command is used to delete the specified kernel file descriptor in the kernel log print file table.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# logfileclear 7
[root@sylixos_station:/root]# logfiles
log fd(s) include :
    1
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_LOG_LIB_EN> 0, the system is allowed to provide a log management library and this command will be included.

Function interface:

The logfileclear command is implemented through c language functions. The function prototype is:

Static INT __tshellSysCmdLogFileClear (INT iArgC, PCHAR ppcArgV[]);

.

3.2.20. logfiles -Display the list of kernel log print files

Format:

logfiles

Explanation:

This command is used to display a list of kernel log print files.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# logfiles
log fd(s) include :
1
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_LOG_LIB_EN> 0, the system is allowed to provide a log management library and this command will be included.

Function interface:

The logfiles command is implemented through c language functions. The function prototype is:

Static INT __tshellSysCmdLogFiles (INT iArgC, PCHAR ppcArgV[]);

3.2.21. ls - Display the file name in the specified directory, with the default current directory

```
ls
ls path name
```

Explanation:

This command is used to display the file name in the specified directory. Print the file name in the current directory with no parameters at default. With a path, it prints a list of file names in the corresponding directory.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ls
hello      test
[root@sylixos_station:/root]# mkdir test2
[root@sylixos_station:/root]# ls
test2      hello      test
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The ls command is implemented through c language functions. The function prototype is:

Static INT __tshellFsCmdLls(INT iArgC, PCHAR ppcArgV[]);

3.2.22. mkdir -Create a directory

Format:**Explanation:**

mkdir directory

This command is used to create a folder in the corresponding directory, with the default current path.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

The directory parameter can take a path and the folder will be created in the specified path.

Example:

```
[root@sylixos_station:/root]# mkdir dir
[root@sylixos_station:/root]# ls
dir
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The mkdir command is implemented by a c language function. The function prototype is:

Static INT __tshellFsCmdRmdir (INT iArgC, PCHAR ppcArgV[]);

3.2.23. mkfifo - Create a named pipe that can only be created under the root file system device

Format:

mkfifo fifo name

Explanation:

This command creates a named pipe in the root directory.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/]# mkfifo fif01
[root@sylixos_station:/]# ls
fif01      tmp      var      root     home
apps      sbin     bin      usr      lib
qt         ftk      etc      boot     usb
yaffs2     proc     media    mnt      dev
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The mkfifo command is implemented by a c language function. The function prototype is:

Static INT __tshellFsCmdMkfifo (INT iArgC, PCHAR ppcArgV[]);

3.2.24. mkfs –Format the specified disk

Format:

mkfs media name

Explanation:

This command is used to format the specified disk.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

X86 environment testing.

Example:

```
[root@sylixos:/root]# mkfs /media/hdd0/
now format media, please wait...
disk format ok.
[root@sylixos:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The mkfs command is implemented through c language functions. The function prototype is:

Static INT __tshellFsCmdMkfs (INT iArgC, PCHAR ppcArgV[]);

3.2.25. mmmaps - Display the system mmap information

Format:

Mmaps

Explanation:

This command is used to display the system mmap information. The information includes: ADDR address, mapping length (SIZE), file offset (OFFSET), page attributes, writable (WRITE), mapping identification, sharing (SHARE) , File Descriptor (FD).

Return value:

Return 0 if the command is executed successfully;

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# mmmaps
ADDR      SIZE      OFFSET      WRITE SHARE  PID  FD
-----
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_POSIX_EN> 0 and LW_CFG_VMM_EN> 0, the posix-compatible library is enabled and the virtual memory is supported. This command will be included.

Function interface:

The mmmaps command is implemented through c language functions. The function prototype is:

Static INT __tshellMmaps (INT iArgC, PCHAR *ppcArgV);

3.2.26 Mount - Mount a volume

Format:

mount -t fstype-o optionblk devmount path

Explanation:

This command is used to mount a volume.

-t Specify the file system type of the device

Nfs Network file system

ramfs Memory-based file system

romfs Simple, compact, read-only file system

Tpsfs SylixOS Integrated File Management System for Mass Storage

Devices

-o Specify read and write mode

ro Read-only mode mount

Rw Mounting in read-write mode

blk dev Equipment name

mount path Mount path

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

x86 environmental testing

Example:

```
[root@sylixos:/root]# mount /dev/blk/hdd-0 /mnt/hdd1
[root@sylixos:/root]# showmount
AUTO-Mount point show >>
      VOLUME                      BLK NAME
-----
/media/hdd0                      /dev/blk/hdd-0:0

Mount point show >>
      VOLUME                      BLK NAME
-----
/mnt/hdd1                        /dev/blk/hdd-0
/tmp                             0
[root@sylixos:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0) and LW_CFG_MOUNT_EN> 0, the number of volumes supported by the system at the same time> 0 and the system needs to use the mount tool, this command will be included.

Function interface:

The mount command is implemented by a c language function. The function prototype is:

```
static INT __tshellFsCmdMount (INT iArgC, PCHAR ppcArgV[]);
```

3.2.27. msgq – Display message queue information

Format:

msgq messagequeuehandle

Explanation:

This command is used to view the messages of the specified message queue.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

tp View message queue handles

```
[root@sylixos_station:/root]# msgq 1c01007e
MsgQueue show >>

MsgQueue Name      : lwip_msg
MsgQueue Id        : 0x1c01007e
MsgQueue Max Msgs   : 512
MsgQueue N Msgs     : 0
MsgQueue Max Msg Len: 4
Thread Queuing      : FIFO
Pended Threads      : 1

[root@sylixos_station:/root]#
```


Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

函数接口:

Function interface:

The msgq command is implemented by a c language function. The function prototype is:

```
tatic INT __tshellSysCmdMsgq (INT iArgC, PCHAR ppcArgV[]);
```

3.2.28. mv- Move to or rename a file**Format:**

```
mv SRC file nameDST file name
mv SRC file nameDST file name
```

Explanation:

This command only has one usage. If both are renamed src file to dst file in the same directory, srcfile is not moved to dst directory under the same directory and named dst.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

Mv will monitor the existence of the target file. If it exists, give a prompt to indicate whether it is covered. If it is covered, it will move, otherwise it will not move.

Example:

```
[root@sylixos_station:/root]# cp a /home/b
copy complete. size:1013(Bytes) time:0(s) speed:1013(Bps)
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The mv command is implemented by a c language function. The function prototype is:
Static INT __tshellFsCmdMv (INT iArgC, PCHAR ppcArgV[]);

3.2.29. open- Open a file**Format:**

```
open filename flag
open filename flag mode
```

Explanation:

This command is used to open or create a file without permission.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

flag option is:

O_RDONLY : 00000000
O_WRONLY : 00000001
O_RDWR : 00000002
O_APPEND : 00000008
O_SHLOCK : 00000010
O_EXLOCK : 00000020
O_ASYNC : 00000040
O_CREAT : 00000200
O_TRUNC : 00000400
O_EXCL : 00000800
O_SYNC : 00002000
O_NONBLOCK : 00004000
O_NOCTTY : 00008000
O_CLOEXEC : 00080000.

Example:

```
[root@sylixos_station:/root]# open a 8  
open file return: 7 dev 3062d1d4 inode 10f size 3f5  
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The open command is implemented by a c language function. The function prototype is:

```
static INT __tshellSysCmdOpen (INT iArgC, PCHAR ppcArgV[]);
```

3.2.30. pwd-View the current working directory

Format:

pwd

Explanation:

This command is used to display the current working directory.

Return value:

Return 0 if the command is executed successfully;

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# pwd  
/root  
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

```
static INT __tshellFsCmdPwd (INT iArgC, PCHAR ppcArgV[]);
```

The cpus command is implemented through c language functions. The function prototype is:

```
static INT __tshellFsCmdPwd (INT iArgC, PCHAR ppcArgV[]);
```

3.2.31. rm-Delete a file

Format:

rm file name

Explanation:

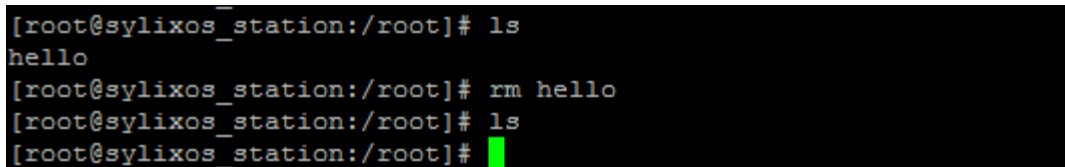
This command is used to delete a file.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ls
hello
[root@sylixos_station:/root]# rm hello
[root@sylixos_station:/root]# ls
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The rm command is implemented by a c language function. The function prototype is:

```
static INT __tshellFsCmdRm (INT iArgC, PCHAR ppcArgV[]);
```

3.2.32. rmdir- Delete a folder

Format:

rmdir directory

Explanation:

There is only one use of this command to delete a given folder.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

Only empty folder can be deleted. The deletion fails when there are files or directories under the folder.

Example:

```
[root@sylixos_station:/root]# ls
hello          a
[root@sylixos_station:/root]# rmdir a
[root@sylixos_station:/root]# ls
hello
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The rmdir command is implemented by a c language function. The function prototype is:

```
static INT __tshellFsCmdMkdir (INT iArgC, PCHAR ppcArgV[]);
```

3.2.33. showmount –View all mounted volumes in the system**Format:**

Showmount

Explanation:

This command is used to view all mounted volumes in the system.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

X86 environment testing.

Example:

```
[root@sylixos:/root]# showmount
AUTO-Mount point show >>
      VOLUME          BLK NAME
-----
/media/hdd0          /dev/blk/hdd-0:0

Mount point show >>
      VOLUME          BLK NAME
-----
/tmp                  0
[root@sylixos:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0) and LW_CFG_MOUNT_EN> 0, the number of volumes supported by the system at the same time> 0 and the system needs to use the mount tool, this command will be included.

Function interface:

The showmount command is implemented through c language functions. The function prototype is:

```
static INT __tshellFsCmdShowmount (INT iArgC, PCHAR ppcArgV[]);
```

3.2.34 Shfile - Execute the specified shell script

Format:

shfile shell file

Explanation:

This command is used to execute the specified shell script.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

Create a shell script file.

```
[root@sylixos_station:/root]# vi test.sh
#!/bin/bash
echo "hello world"
```

Use shfile running script.

```
[root@sylixos_station:/root]# shfile test.sh
hello world
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The shfile command is implemented by a c language function. The function prototype is:

```
Static INT __tshellFsCmdShfile (INT iArgC, PCHAR ppcArgV[]);
```

3.2.35. Sync - Write all system cached information to physical devices

Format:

Sync

Explanation:

This command writes all system-cached files, devices, and disk information to the corresponding physical device.

Return value:

Return 0 if the command is executed successfully.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# sync  
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The sync command is implemented via c language functions. The function prototype is:

```
static INT __tshellSysCmdSync (INT iArgC, PCHAR ppcArgV[]);
```

3.2.36. tmpname - Get a temporary file name that can be created

Format:

tmpname

Explanation:

This command is used to get a temporary file name that can be created.

Return value:

Return "0" if the command is executed successfully.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# tmpname  
can mktmp as name: /tmp/tmp.0.HjNXRI  
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The tmpname command is implemented by a c language function. The function prototype is:

```
static INT __tshellFsCmdTmpname (INT iArgC, PCHAR ppcArgV[]);
```

3.2.37. Touch - Create a common file

Format:

```
touch file name  
touch -amc file name
```

Explanation:

There are 2 kinds of usage of this command, directly create a common file with the file name; parameters can be combined with one or more of the AMC, where a means to change the file access time, m means to change the modification time, c means if the

file does not exist, the file is not created. If you do not follow the parameter c and follow a or m, if the file does not exist, the file will be created first.

Return value:

Return 0 if the command is executed successfully and return non-zero if the command fails.

Remarks:

The functions of parameters a and m are temporarily unimplemented.

Example:

```
[root@sylixos_station:/tmp/tmp.0.mhA9dG]# ls
a
[root@sylixos_station:/tmp/tmp.0.mhA9dG]# touch temp
[root@sylixos_station:/tmp/tmp.0.mhA9dG]# ls
temp      a
[root@sylixos_station:/tmp/tmp.0.mhA9dG]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0), the number of volumes supported by the system at the same time> 0, the command will be included.

Function interface:

The touch command is implemented by a c language function. The function prototype is:

Static INT __tshellFsCmdTouch (INT iArgC, PCHAR ppcArgV[]);

3.2.38. umount - Unload a volume

Format:

umount mount path

Explanation:

This command is used to uninstall a volume that already exists on the system.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

X86 environment testing.

Example:

```
[root@sylixos:/root]# umount /mnt/hdd1
[root@sylixos:/root]# showmount
AUTO-Mount point show >>
      VOLUME                      BLK NAME
-----
/media/hdd0                      /dev/blk/hdd-0:0

Mount point show >>
      VOLUME                      BLK NAME
-----
/tmp                              0
[root@sylixos:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MAX_VOLUMES> 0) and LW_CFG_MOUNT_EN> 0, the number of volumes supported by the system at the same time> 0 and the system needs to use the mount tool, this command will be included.

Function interface:

The umount command is implemented by a c language function. The function prototype is:

```
static INT __tshellFsCmdUmount (INT iArgC, PCHAR ppcArgV[]);
```

3.2.39. untar - Unpack or untar a tar or tar.gz package.

Format:

```
untar .tar or .tar.gz file destination directory
```

Explanation:

This command is used to extract a .tar.gz or .tar file. You can specify the decompression path.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/home]# untar b.tar.gz
unpackage test.c size: 356 ...
unpackage a.out size: 4754 ...
unpackage total 2 files 0 directory.
[root@sylixos_station:/home]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SHELL_TAR_EN>0, this command will be included after the tar tool is enabled.

Function interface:

The untar command is implemented through c language functions. The function prototype is:

```
Static INT __tshellFsCmdUntar (INT iArgC, PCHAR ppcArgV[]);
```

3.2.40. vi- Start the vi editor

Format:

```
vi [filename]
```

Explanation:

This command is used to open the vi editor. You can specify the file name in advance or enter the file name when editing the document.

Return value:

Return 0 if the command is executed successfully.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/home]# vi a
~
~
~
~
~
~
~
~
~
~
~
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, it allows the operating system to provide a tshell command. This command is included.

Function interface:

The vi command is implemented by a c language function. The function prototype is:
Int vi_main(int argc, char **argv);

3.2.41. yaffscmd - Display, set, and erase a block

Format:

```
yaffscmd volname bad
yaffscmd volname info
yaffscmd volname markbad
yaffscmd volname erase
```

Explanation:

There are 4 usages of this command, including
Yaffscmd volname bad View bad blocks marked in the volume.
Yaffscmd volname info View all blocks in the volume.
Yaffscmd volname markbad Mark bad blocks in the volume.
Yaffscmd volname erase Erase a block in a volume.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```

[root@sylixos_station:/home]# yaffscmd n0 info
Device : "/n0"
startBlock..... 1
endBlock..... 128
totalBytesPerChunk. 2048
chunkGroupBits..... 0
chunkGroupSize..... 1
nErasedBlocks..... 126
nReservedBlocks.... 10
nCheckptResBlocks.. nil
blocksInCheckpoint. 0
nObjects..... 10
nTnodes..... 4
nFreeChunks..... 8181

[root@sylixos_station:/home]# yaffscmd n0 markbad 54
yaffs: marking block 84 bad
mark the block 0x54 is a bad ok.
[root@sylixos_station:/home]# yaffscmd n0 bad
block 0x1 is bad block.
block 0x2 is bad block.
block 0x24 is bad block.
block 0x54 is bad block.
[root@sylixos_station:/home]# █

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When (LW_CFG_MAX_VOLUMES > 0) and (LW_CFG_YAFFS_EN > 0) the system supports multiple volumes at the same time and allows the YAFFS file system, this command will be included.

Function interface:

The yaffscmd command is implemented via c language functions. The function prototype is:

```
static INT __tshellYaffsCmd (INT iArgC, PCHAR ppcArgV[]);
```

3.2.42. zlib- Add a .gz compressed file

Format:

```
zlib test.zlib
```

Explanation:

This command is used to add a .gz compressed file, zlib test interface.

Return value:

Return 0 if the command is executed successfully and return non-zero if the command fails.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# zlib test.gz
zlib version 1.2.8 = 0x1280, compile flags = 0x55
uncompress(): hello, hello!
gzread(): hello, hello!
gzgets() after gzseek: hello!
inflate(): hello, hello!
large_inflate(): OK
after inflateSync(): hello, hello!
inflate with dictionary: hello, hello!
[root@sylixos_station:/root]# ls
test.gz
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The zlib command is implemented by a c language function. The function prototype is:

```
int zlib_main(int argc, char *argv[]);
```

4. User's commands

4.1. Introduction

- gadd - Add a new user group
- gdel – Delete a user group
- group – Display the information of user group
- pmod – Modify user password
- uadd – Add user
- udel – Delete user
- umod – Set the user's mode
- user – Display user information or generate a password for the user

4.2. Use of commands

4.2.1. gadd – Add a new user groupd

Format:

gadd group_name gid

Explanation:

This command is used to add a user group and assign a gid to the group.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# gadd test 4
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell command. When (LW_CFG_SHELL_USER_EN > 0), the shell user management tool is enabled, and this command will be included.

Function interface:

The “gadd” command is implemented by a c language function. The function prototype is:

```
static INT __tshellUserCmdGadd (INT iArgC, PCHAR ppcArgV[]);
```

4.2.2. gdel – Delete a user group

Format:

gdel group_nam

Explanation:

This command is used to delete a user group.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

You can only delete empty user groups with users. If the user group is not empty, the deletion fails.

Example:

```
[root@sylixos_station:/root]# gdel test
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When (LW_CFG_SHELL_USER_EN > 0), the shell user management tool is enabled, and this command will be included.

Function interface:

The “gdel” command is implemented by a c language function. The function prototype is:

```
static INT __tshellUserCmdGdel (INT iArgC, PCHAR ppcArgV[]);
```

4.2.3. group - Display the information of user group

Format:

group

Explanation:

This command is used to display information of all user groups.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# group

  GROUP      GID      USERS
-----
root         0      root,
server       100    server,
user         200    user,
apps         300    user,
anonymous   400    anonymous,
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When (LW_CFG_SHELL_USER_EN > 0), the shell user management tool is enabled, and this command will be included.

Function interface:

The “group” command is implemented by a c language function. The function prototype is:

```
static INT __tshellUserCmdGroup(INT iArgC, PCHAR ppcArgV[]);
```

4.2.4. pmod – Modify user password

Format:

pmod name old_password new_password

Explanation:

This command is used to modify user password.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

Log in as the root user to change the password.

Example:

```
[root@sylixos_station:/root]# pmod liang liang 1234
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When (LW_CFG_SHELL_USER_EN > 0), the shell user management tool is enabled, and this command will be included.

Function interface:

The “pmod” command is implemented by a c language function. The function prototype is:

```
static INT __tshellUserCmdPmod(INT iArgC, PCHAR ppcArgV[]);
```

4.2.5. uadd – Add user

Format:

uadd name password enable[0 / 1] uid gid comment homedir

Explanation:

This command is used to add a user.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

The user id cannot be duplicated. When adding a user, it must be added to an existing user group.

Example:

```
[root@sylixos_station:/root]# uadd liang liang 1 2 0 liang /home/liang
[root@sylixos_station:/root]# user
```

USER	ENABLE	UID	GID
root	yes	0	0
hanhui	yes	1	0
anonymous	no	400	400
liang	yes	2	0

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When (LW_CFG_SHELL_USER_EN > 0), the shell user management tool is enabled, and this command will be included.

Function interface:

The “uadd” command is implemented by a c language function. The function prototype is:

```
static INT __tshellUserCmdUadd (INT iArgC, PCHAR ppcArgV[]);
```

4.2.6. udel – Delete user

Format:

```
udel username
```

Explanation:

This command is used to delete a user.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# udel test
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When (LW_CFG_SHELL_USER_EN > 0), the shell user management tool is enabled, and this command will be included.

Function interface:

The “udel” command is implemented by a c language function. The function prototype is:

```
static INT __tshellUserCmdUdel(INT iArgC, PCHAR ppcArgV[]);
```

4.2.7. umod - Set the user’s mode

Format:

```
umod name enable[0 / 1] comment homedir
umod name enable[0 / 1] comment homedir
```

Explanation:

This command is used to set the user’s enablement, user description and user directory.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# umod liang 0 testuser /home/liang/
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When (LW_CFG_SHELL_USER_EN > 0), the shell user management tool is

enabled, and this command will be included.

Function interface:

The “umod” command is implemented by a c language function. The function prototype is:

```
static INT __tshellUserCmdUmod(INT iArgC, PCHAR ppcArgV[]);
```

4.2.8. user – Display user information or generate a password for the user

Format:

```
user
user genpass
```

Explanation:

There are two kinds of usage for this command. When there are no parameters, the user name, enablement, user id and group id of all users are displayed. When the parameter “genpass” is followed, a password is generated for the user.

Return value:

Return “0” on success and non-“0” on failure.

备注:

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# user
```

USER	ENABLE	UID	GID
root	yes	0	0
hanhui	yes	1	0
anonymous	no	400	400
liang	yes	2	0

```
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SHELL_USER_EN > 0, the shell user management tool is enabled, and the command will be included.

Function interface:

The “user” command is implemented by a c language function. The function prototype is:

```
static INT __tshellUserCmdUser(INT iArgC, PCHAR ppcArgV[]);
```


5. Network

5.1. Introduction

The network-related commands are:

- aodvs – Display aodvs routing table
- arp – Add, delete, or view ARP tables
- ftpdpath – View or set the ftp server initialization path
- ftpds – Display ftp server information
- hosttable – View, add, or delete host address mapping
- ifonfig – Display or configure network configuration information
- ifdown – Disable a network interface
- ifronter – Set the default route interface
- ifup – Enable a network interface
- ipv6 – Set or show ipv6
- nat – Enable, disable or set the NAT virtual network address service
- natalias
- natmap
- nats – View current NAT virtual address service status
- ndname – Display or set the name of this machine's NetBIOS
- netstat – View network status
- npfattach – Enable network packet filters on the specified network interface
- npfdetach – Disable network packet filters on the specified network interface
- npfruleadd – Add a network packet filter rule
- npfruledel – Delete a network packet filter rule
- npfs – View network packet filter status
- ping – Ping command
- ping6 – Ipv6 Ping command
- route – Add, delete, modify, or view system routing tables
- tftp – Use the tftp command to receive or send a file
- tftpdpath – View or set the local path of the tftp server
- vlan – Display, set, and delete net interfaces
- vpnclose – Delete a virtual network interface
- vpnopen – Create a virtual network interface

5.2. Use of commands

5.2.1. aodvs – Display the aodv routing table

Format:

aodvs

Explanation:

This command is used to display the aodvs routing table.

Return value:

Return "0" on success and "-1" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# aodvs
aodv routing tables
Destination      Gateway          Mask            Flag            Hops            Interface
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The “aodvs” command is implemented through c language functions. The function prototype is:

```
static INT __tshellAodvs (INT iArgC, PCHAR *ppcArgV);
```

5.2.2. arp – Add, delete or view ARP tables

Format:

```
arp-a
arp -s inet_address physical_address
arp -d inet_address
```

Explanation:

There are three kinds of usage for this command, where “arp-a” is used to view the ARP table; “arp -s inet_addressphysical_address” is used to add an address to the ARP table; and “arp -d inet_address” is used to delete an address that exists in the ARP table.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

The arp table information is stored in the /proc/net/arp file.

Example:

```
[root@sylixos_station:/root]# arp -a
FACE INET ADDRESS      PHYSICAL ADDRESS  TYPE
[root@sylixos_station:/root]# arp -s 192.168.7.60 00:11:22:33:44:55
[root@sylixos_station:/root]# arp -a
FACE INET ADDRESS      PHYSICAL ADDRESS  TYPE
en1  192.168.7.60       00:11:22:33:44:55 static
[root@sylixos_station:/root]# arp -d 192.168.7.60
[root@sylixos_station:/root]# arp -a
FACE INET ADDRESS      PHYSICAL ADDRESS  TYPE
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The “arp” command is implemented through c language functions. The function prototype is:

```
static INT __tshellArp (INT iArgC, PCHAR *ppcArgV);
```

5.2.3 ftpdpath – View or set the ftp server initialization path

Format:

```
ftpdpath  
ftpdpath new path
```

Explanation:

There are two kinds of usage for this command. When there are no parameters, the initial path of the ftp server is displayed. When the parameter “genpass” is followed, the path of ftp is set.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ftpdpath  
ftpd path: /proc/  
[root@sylixos_station:/root]# ftpdpath /  
[root@sylixos_station:/root]# ftpdpath  
ftpd path: /  
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When (LW_CFG_NET_EN> 0) and (LW_CFG_NET_FTPD_EN> 0), it is allowed to provide network functions and the ftp server is enabled, and this command will be included.

Function interface:

The “ftpdpath” command is implemented by a c language function. The function prototype is:

```
static INT __tshellNetFtpdPath (INT iArgC, PCHAR ppcArgV[]);
```

5.2.4. ftpds – Display ftp server information

Format:

```
ftpds
```

Explanation:

This command is used to display the ftp server information.

Return value:

Return “0” if the command is executed.

Remarks:

N/A.

Example:

```

[root@sylixos_station:/root]# ftpds
ftpd show >>
ftpd path: /

REMOTE                                TIME                                ALIVE (s)
-----
total ftp session: 0
[root@sylixos_station:/root]# █

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When (LW_CFG_NET_EN> 0) and (LW_CFG_NET_FTPD_EN> 0), it is allowed to provide network functions and the ftp server is enabled, and this command will be included.

Function interface:

The “ftpdpath” command is implemented by a c language function. The function prototype is:

```
static INT __tshellNetFtpdShow (INT iArgC, PCHAR ppcArgV[]);
```

5.2.5. hosttable – View, add or delete host address mapping

Format:

```

hosttable
hosttable -s host addr
hosttable -d host

```

Explanation:

There are three kinds of usage for this command, where there is no parameter, the command is used to display the address mapping relationship of the host; the parameter “-s” means to add the address mapping relationship, followed by the host name and address; and the parameter “-d” is used to delete the address mapping relationship, followed by the host name.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```

[root@sylixos_station:/root]# hosttable -s localhost 127.0.0.1
[root@sylixos_station:/root]# hosttable

addr                                HOST
-----
127.0.0.1                          localhost
[root@sylixos_station:/root]# hosttable -d localhost
[root@sylixos_station:/root]# hosttable

addr                                HOST
-----
[root@sylixos_station:/root]# █

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The “hosttable” command is implemented by a c language function. The function prototype is:

```
static INT __tshellHostTable (INT iArgC, PCHAR ppcArgV[]);
```

5.2.6. ifconfig – Display or configure network configuration

information

Format:

```
ifconfig
ifconfig netifname
ifconfig netifnameinetaddress
ifconfig netifnamenetmaskaddress
ifconfig netifnamegateway address
ifconfig dns 0 address
```

Explanation:

There are five kinds of usage for this command, where there is no parameter, the command is used to display all the network configuration information; followed by the name of the network port, it displays configuration information of the specified network port; the “inet” is used to configure the IP address, and the “netmask” is used to configure the subnet mask; “gateway” is used to configure the default gateway; and “dns 0” is used to configure the preferred dns server, and “dns 1” is used to configure the standby dns server.

Return value:

Return “0” if the command is executed.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ifconfig en1 inet 192.168.7.32
[root@sylixos_station:/root]# ifconfig en1 netmask 255.255.255.0
[root@sylixos_station:/root]# ifconfig en1 gateway 192.168.7.1
[root@sylixos_station:/root]# ifconfig dns 192.168.1.254
[root@sylixos_station:/root]# ifconfig en1
en1      enable: true linkup: true MTU: 1500 multicast: false
        metric: 1 type: Ethernet-Cap HWaddr: 08:08:3E:26:0A:5A
        DHCP: Disable(Off) speed: 100(Mbps)
        inet addr: 192.168.7.32 netmask: 255.255.255.0
        gateway: 192.168.7.1 broadcast: 192.168.7.255
        inet6 addr: FE80::A08:3EFF:FE26:A5A Scope:link <valid>
        RX ucast packets:628 nucast packets:0 dropped:0
        TX ucast packets:9 nucast packets:0 dropped:0
        RX bytes:61752 TX bytes:594

[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The “ifconfig” command is implemented through c language functions. The function prototype is:

```
static INT __tshellIfconfig (INT iArgC, PCHAR *ppcArgV);
```

5.2.7. ifdown – Disable a network interface

Format:

```
ifdown netifname
```

Explanation:

This command is used to disable the specified network port.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ifdown en1
net interface "en1" set down.
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The “ifdown” command is implemented by a c language function. The function prototype is:

```
static INT __tshellIfDown (INT iArgC, PCHAR *ppcArgV);
```

5.2.8. ifrouter – Set the default route interface

Format:

```
ifconfig netifname
```

Explanation:

This command is used to set the default route interface.

Return value:

Return “0” if the command is executed.

Remarks:

This command has been replaced by route.

Example:

N/A.

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The “ifrouter” command is implemented via c language functions. The function prototype is:

```
static INT __tshellIfRouter (INT iArgC, PCHAR *ppcArgV);
```

5.2.9. ifup – Enable a network interface

Format:

```
ifup netifname  
ifupnetifname -dhcp  
ifupnetifname-nodhcp
```

Explanation:

There are 3 kinds of usage for this command, including the command followed by the network interface only, used to enable a network; followed by the parameter “-dhcp”, indicating that the network address is obtained using “dhcp” customer service (automatically obtain the IP address); and followed by the parameter “-nodhcp”, indicating that “dhcp” is not used to obtain IP address.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

1. “Ifup netifname” enables the network interface, depending on whether dhcp is used or not last time.
2. The use of dhcp needs to be configured with LWIP_DHCP > 0, and whether dhcp is allowed.

Example:

```
[root@sylixos_station:/root]# ifup en1  
DHCP client starting...  
DHCP client start.  
net interface "en1" set up.  
[root@sylixos_station:/root]# ifup en1 -nodhcp  
net interface "en1" set up.  
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The “ifup” command is implemented through c language functions. The function prototype is:

```
static INT __tshellIfUp (INT iArgC, PCHAR *ppcArgV);
```

5.2.10. ipv6 – Set or show ipv6

Format:


```
ipv6 address ifname address%prefixlen
ipv6 noaddress ifname address%prefixlen
```

Explanation:

There are two kinds of usage for this command, followed by the parameter “address”, the command is used to set the address of ipv6 for “ifname”; followed by the parameter “noaddress”, it is used to remove this ipv6 address from the interface.

Return value:

Return “0” if the command is executed.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ipv6 address en1 80::d473:8fd4:249c:ba%14
[root@sylixos_station:/root]# ifconfig en1
en1      enable: true linkup: true MTU: 1500 multicast: false
        metric: 1 type: Ethernet-Cap HWaddr: 08:08:3E:26:0A:5A
        DHCP: Disable(On) speed: 100(Mbps)
        inet addr: 0.0.0.0 netmask: 0.0.0.0
        gateway: 0.0.0.0 broadcast: 255.255.255.255
        inet6 addr: FE80::A08:3EFF:FE26:A5A Scope:link <valid>
        inet6 addr: 21::1:30E1:F9BC:3013:4E68 Scope:unknown <valid>
        inet6 addr: 80::D473:8FD4:249C:BA Scope:unknown <valid>
        RX ucast packets:703 nucast packets:0 dropped:0
        TX ucast packets:92 nucast packets:0 dropped:0
        RX bytes:69102 TX bytes:14600

[root@sylixos_station:/root]#

[root@sylixos_station:/root]# ipv6 noaddress en1 80::d473:8fd4:249c:ba%14
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The “ipv6” command is implemented by the c language function. The function prototype is:

```
static INT __tshellIpv6(INT iArgC, PCHAR *ppcArgV);
```

5.2.11. nat – Enable, disable or set the NAT virtual network

address service

Format:

```
nat stop
nat LAN netif WAN netif
```

Explanation:

There are two kinds of usage for this command, followed by the parameter “stop”, the command is used to turn off the NAT virtual network address service; and followed by the parameter “LAN netif WAN netif”, it starts the NAT virtual network address service.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# nat stop
NAT network stoped.
[root@sylixos_station:/root]# nat lo0 en1
NAT network started, [LAN: lo0] [WAN: en1]
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NAT_EN> 0, it is allowed to provide network functions and use the nat service, and this command will be included.

Function interface:

The “nat” command is implemented by a c language function. The function prototype is:

```
static INT __tshellNat (INT iArgC, PCHAR ppcArgV[]);
```

5.2.12. natalias – Add or delete NAT alias

Format:

```
natalias addaliasLAN startLAN end
natalias delalias
```

Explanation:

There are two kinds of usage for this command. Used together with the parameter “add”, the command is used to add an alias to the segment from “LAN start” to “LAN end”; and used together with the parameter “del”, it is used to delete a NAT alias.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# natalias add 192.168.7.2 192.168.7.2 192.168.7.9
[root@sylixos_station:/root]# natalias del 192.168.7.2
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NAT_EN> 0, it is allowed to provide network functions and use the nat service, and this command will be included.

Function interface:

The “natalias” command is implemented through c language functions. The function prototype is:

```
static INT __tshellNatAlias (INT iArgC, PCHAR ppcArgV[]);
```

5.2.13. natmap – Add or delete NAT mapping

Format:

```

natmap addWAN portLAN portLAN IPprotocol
natmap delWAN portLAN portLAN IPprotocol

```

Explanation:

There are two kinds of usage for this command. Used together with the parameter “add”, the command is used to add a NAT mapping; and used together with the parameter “del”, it is used to delete a NAT mapping. There are two protocols: tcp and udp.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```

[root@sylixos_station:/root]# natmap add 80 80 192.168.7.65 tcp
[root@sylixos_station:/root]# nats
NAT networking alias setting >>

  ALIAS      LOCAL START  LOCAL END
-----
NAT networking direct map setting >>

  ASS PORT  LOCAL PORT  LOCAL IP  PROTO
-----
      80      80 192.168.7.65  TCP

NAT networking summary >>
  LAN: lo0 WAN: en1
  Total Ass-node: 2048
  Used Ass-node: 0
[root@sylixos_station:/root]# natmap add 80 80 192.168.7.65 tcp
[root@sylixos_station:/root]#

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NAT_EN> 0, it is allowed to provide network functions and use the nat service, and this command will be included.

Function interface:

The “natmap” command is implemented by a c language function. The function prototype is:

```
static INT __tshellNatMap (INT iArgC, PCHAR ppcArgV[]);
```

5.2.14. nats – View current NAT virtual address service status

Format:

```
nats
```

Explanation:

This command is used to check the status of current NAT virtual address service.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

The state of nats is in the /proc/net/nat/info file.

Example:

```
[root@sylixos_station:/root]# nats
NAT networking alias setting >>

  ALIAS          LOCAL START    LOCAL END
-----
NAT networking direct map setting >>

  ASS PORT  LOCAL PORT    LOCAL IP    PROTO
-----
NAT networking summary >>
  LAN: lo0 WAN: en1
  Total Ass-node: 2048
  Used Ass-node: 0
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NAT_EN> 0, it is allowed to provide network functions and use the nat service, and this command will be included.

Function interface:

The “nats” command is implemented through c language functions. The function prototype is:

```
static INT __tshellNatShow (INT iArgC, PCHAR ppcArgV[]);
```

5.2.15. nbname – Display or set the name of this machine’s

NetBIOS

Format:

```
nbname
nbname hostname
```

Explanation:

There are two kinds of usage for this command, it is used to display the name of NetBIOS of this machine without any parameter; and it is used to set the name of NetBIOIS with the parameter.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# nbname
local host NetBIOS name is: SYLIXOS
[root@sylixos_station:/root]# nbname user
[root@sylixos_station:/root]# nbname
local host NetBIOS name is: USER
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NETBIOS_EN> 0, it is allowed to provide network functions and enable the simple “netbios” name service, and this command will be included.

Function interface:

The “nbname” command is implemented by a c language function. The function prototype is:

```
static int __inetBiosNameSet (int iArgC, char *pcArgV[]);
```

5.2.16. netstat – View network status

Format:

```
netstat
netstat -wtux -A -i
netstat-hrigrsapl
```

Explanation:

There are three kinds of usage for this command. The command without parameter is used to display all socket information; and with parameter, corresponding information is displayed.

-h, --help	Display help information
-r, --route	Display routing table information
-i, --interface	Display network interface information
-g, --groups	Display the multicast table status
-s, --statistics	Display statistic information
-a, --all	Display all socket information
-p, --packet	Display packet socket information
-l, --listening	Display the received server socket information
-w, --raw	Display raw socket information
-t, --tcp	Display tcp socket information
-u, --udp	Display udp socket information
-x, --unix	Display unix socket information

The “-A <net type>” lists related addresses in the connection of this network type, there are inet, inet6 and unix.

Numbers 1, 2 and 3 Network types can be represented by numbers, 1 for unix, 2 for inet, and 3 for inet6.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

```

[root@sylixos_station:/root]# netstat
--UNIX--:
TYPE          FLAG STATUS  SHUTD      NREAD MAX_BUFFER PATH
--PACKET--:
TYPE          FLAG PROTOCOL INDEX MMAP MMAP_SIZE TOTAL      DROP
--TCP LISTEN--:
LOCAL          REMOTE          STATUS  RETRANS RCV_WND SND_WND
*:21           *:              listen   0        0        0
*:23           *:              listen   0        0        0
--UDP--:
LOCAL          REMOTE          UDPLITE
*:69           *:0             no
*:137          *:0             no
*:161          *:0             no

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The “netstat” command is implemented through c language functions. The function prototype is:

```
static INT __tshellNetstat (INT iArgC, PCHAR *ppcArgV);
```

5.2.17. npfattach – Enable network packet filter on the specified network interface

Format:

```
npfattach etifname
```

Explanation:

This command is used to enable the network packet filter on the specified network interface.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```

[root@sylixos_station:/root]# npfattach en1
attached.
[root@sylixos_station:/root]# npfs
NETIF ATTACH SEQNUM RULE ALLOW MAC          IPs          IPe          PORTs  PORTe
en1  YES      0 MAC  NO    12:12:25:12:45:65 N/A          N/A          N/A    N/A
en1  YES      1 IP   NO    N/A          192.168.7.65 192.168.7.96 N/A    N/A
drop:0 allow:1

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell

commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NPF_EN> 0, it is allowed to provide network functions and enable NPF service, and this command will be included.

Function interface:

The “npfattach” command is implemented via c language functions. The function prototype is:

```
static INT __tshellNetNpfAttach (INT iArgC, PCHAR *ppcArgV);
```

5.2.18. npfdetach – Disable network packet filters on the specified network interface

Format:

```
npfdetach netifname
```

Explanation:

There is one kind of usage for this command, which disables the network packet filter on the specified network interface.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# npfdetach en1
detached.
[root@sylixos_station:/root]# npfs
NETIF ATTACH SEQNUM RULE ALLOW MAC IPs IPe PORTs PORTe
en1 NO 0 MAC NO 12:12:25:12:45:65 N/A N/A N/A N/A
en1 NO 1 IP NO N/A 192.168.7.65 192.168.7.96 N/A N/A
drop:0 allow:1
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NPF_EN> 0, it is allowed to provide network functions and enable NPF service, and this command will be included.

Function interface:

The “npfdetach” command is implemented by a c language function. The function prototype is:

```
static INT __tshellNetNpfDetach (INT iArgC, PCHAR *ppcArgV);
```

5.2.19. npfruleadd – Add a network packet filter rule

Format:

```
npfruleadd netifname mac ????:????:????:????:???
npfruleadd netifname ip ????.????.????.??? ????.????.????.???
npfruleadd netifname udp ????.????.????.??? ????.????.????.??? iports iporte
npfruleadd netifname tcp ????.????.????.??? ????.????.????.??? iports iporte
```

Explanation:

This command is used to add a network packet filter rule. The rule type can have mac, ip, udp, and tcp. When the mac rule is added, the parameter is a forbidden mac address array; when adding an ip rule, the parameter is the ip starting and ending address that is forbidden to pass; when adding the udp or tcp rule, the parameters are the forbidden ip starting and ending address, and locally prohibited starting and ending port numbers.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# npfruleadd en1 mac 12:12:25:12:45:65
rule add ok
[root@sylixos_station:/root]# npfruleadd en1 ip 192.168.7.65 192.168.7.96
rule add ok
[root@sylixos_station:/root]# npfs
NETIF ATTACH SEQNUM RULE ALLOW MAC          IPs          IPe          PORTs  PORTe
en1    NO          0 MAC   NO    12:12:25:12:45:65 N/A          N/A          N/A    N/A
en1    NO          1 IP    NO    N/A          192.168.7.65 192.168.7.96 N/A    N/A
drop:0 allow:1
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NPF_EN> 0, it is allowed to provide network functions and enable NPF service, and this command will be included.

Function interface:

The “nbname” command is implemented by a c language function. The function prototype is:

```
static INT __tshellNetNpfRuleAdd (INT iArgC, PCHAR *ppcArgV);
```

5.2.20. npfruledel – Delete a network packet filter rule

Format:

```
npfruledel netifnamerule sequence num
```

Explanation:

This command is used to delete a network packet filter rule.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# npfruledel en1 0
delete.
[root@sylixos_station:/root]# npfs
NETIF ATTACH SEQNUM RULE ALLOW MAC          IPs          IPe          PORTs  PORTe
en1    NO          0 IP    NO    N/A          192.168.7.65 192.168.7.96 N/A    N/A
drop:0 allow:1
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell

commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NPF_EN> 0, it is allowed to provide network functions and enable NPF service, and this command will be included.

Function interface:

The “npfruledel” command is implemented by a c language function. The function prototype is:

```
static INT __tshellNetNpfRuleDel (INT iArgC, PCHAR *ppcArgV);
```

5.2.21. npfs – View network packet filter status

Format:

nbfs

Explanation:

This command is used to check the status of the network packet filter.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

The state of the network packet filter is stored under /proc/net/netfilter.

Example:

```
[root@sylixos_station:/root]# npfs
NETIF ATTACH SEQNUM RULE ALLOW MAC          IPs          IPe          PORTs  PORTe
p57   NO          0 IP   NO    N/A          192.168.7.2   192.168.7.29  N/A    N/A
ma208 NO          0 MAC NO    32:56:21:23:23:32 N/A          N/A          N/A    N/A
drop:0  allow:0
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NPF_EN> 0, it is allowed to provide network functions and enable NPF service, and this command will be included.

Function interface:

The “nbfs” command is implemented through c language functions. The function prototype is:

```
static INT __tshellNetNpfShow (INT iArgC, PCHAR *ppcArgV);
```

5.2.22. ping – Ping command

Format:

ping ip/hostname [-l datalen] [-n times] [-i ttl] [-w timeout]

Explanation:

Data is sent to the specified IP to test the network’s continuity. The parameter “-l” is used to indicate the size of the packet. The parameter “-n” is used to indicate the size of the packet to be sent. The parameter “-i” is used to set the value of “ttl”, up to 255. “-w” is used to specify the timeout period. The default packet size is 32 bytes, the times of transmissions are 4, the TTL is 255, and the timeout period is 3000.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ping 192.168.7.30
Pinging 192.168.7.30

Reply from 192.168.7.30: bytes=32 time=0ms TTL=255
Reply from 192.168.7.30: bytes=32 time=0ms TTL=255
Reply from 192.168.7.30: bytes=32 time=0ms TTL=255
Reply from 192.168.7.30: bytes=32 time=0ms TTL=255

Ping statistics for 192.168.7.30:
    Packets: Send = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_PING_EN> 0, it is allowed to provide network functions and require ping tools, and this command will be included.

Function interface:

The “ping” command is implemented by a c language function. The function prototype is:

```
static INT __tshellPing (INT iArgC, PCHAR *ppcArgV);
```

5.2.23. ping6 – Ipv6 Ping command

Format:

```
ping6 ip(v6)/hostname [-l datalen] [-n times] [-w timeout] [-I interface]
```

Explanation:

Data is sent to the specified ip (ipv6) to test the network's continuity. The parameter “-l” is used to indicate the size of the packet. The parameter “-n” is used to indicate the size of the packet to be sent. The parameter “-I” is used to set the network interface name. “-w” is used to specify the timeout period. The default packet size is 32 bytes, the times of transmissions are 4, and the timeout period is 3000.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# ping6 FE80::A08:3EFF:FE26:A5A
Pinging FE80::A08:3EFF:FE26:A5A

Reply from FE80::A08:3EFF:FE26:A5A: bytes=32 time=0ms hoplim=255
Reply from FE80::A08:3EFF:FE26:A5A: bytes=32 time=0ms hoplim=255
Reply from FE80::A08:3EFF:FE26:A5A: bytes=32 time=0ms hoplim=255
Reply from FE80::A08:3EFF:FE26:A5A: bytes=32 time=0ms hoplim=255

Ping statistics for FE80::A08:3EFF:FE26:A5A:
    Packets: Send = 4, Received = 4, Lost = 0(0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_PING_EN> 0, it is allowed to provide the network function and require “ping” tools, and this command will be included.

Function interface:

The “ping6” command is implemented by a c language function. The function prototype is:

```
static INT __tshellPing6 (INT iArgC, PCHAR *ppcArgV);
```

5.2.24. route – Add, delete, modify, or view system routing tables

Format:

- 1) route
- 2) route add|change -host | -netipaddrgateway if/dev name
- 3) route add|change default if|dev name
- 4) route del ipaddr

Explanation:

There are only four kinds of usage for this command. Format 1, that has no parameter, is used to display the routing table. Format 2 is used to modify or add a routing table information. Format 3 is used to modify or add default routing information. And Format 4 is used to delete a routing information.

Options:

add	Used to add a routing information
change	Change existing routing information
del	Delete existing routing information
-host	Target address is a host.
-net	Target address is a network.
if	It specifies the interface index for the target interface that can be accessed.
dev	It specifies the target interface index as the input name.

default	Operation of default route
Parameters:	
ipaddr	IP address
gateway	Mask
name	The name of the interface index

Check the routing table output information:

Destination: Network destination address, which lists all network segments connected to the router.

Gateway: The gateway, once the router determines which destination network it wants to forward the packet to, the router checks the list of gateways. The gateway table tells the router which IP address the packet should be forwarded to in order to reach the destination network.

Mask: Network mask, which provides the subnet mask of the network segment itself, rather than the subnet mask of the network card connected to this segment. This basically allows the router to determine the destination network's address class.

Flag: The route flag, which indicates the status of the current network node.

U Up, indicating that this route is currently started.

H Host, indicating that the gateway is a host.

G Gateway, indicating that this gateway is a router.

R Reinstate Route, the route that is reinitialized using dynamic routing

D Dynamically, this route is written dynamically

M Modified, this route is dynamically modified by the routing daemon or director

! means this route is currently closed

Interface: The interface index tells the router which network card is connected to the appropriate destination network.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# route add -host 192.168.7.98 255.255.255.0 dev en1
route 192.168.7.98 add successful.
[root@sylixos_station:/root]# route
kernel routing tables
Destination      Gateway          Mask             Flag             Interface
192.168.7.98      *                255.255.255.0    UH               en1

build-in routing tables
Destination      Gateway          Mask             Flag             Interface
192.168.7.0      *                255.255.255.0    U               en1
192.168.7.30      *                255.255.255.0    UH              en1
127.0.0.0         *                255.0.0.0        U               lo0
127.0.0.1         *                255.0.0.0        UH              lo0
default          192.168.7.1      255.255.255.0    UG               en1
[root@sylixos_station:/root]#
```

```
[root@sylixos_station:/root]# route del 192.168.7.98
route 192.168.7.98 delete successful.
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0, it is allowed to provide network functions, and this command will be included.

Function interface:

The route command is implemented through c language functions. The function prototype is:

```
static INT __tshellRoute (INT iArgC, PCHAR ppcArgV[]);
```

5.2.25. tftp – Use the tftp command to receive or send a file

Format:

```
tftp -i Host get | put SourceDestination
```

Explanation:

This command is used to send or receive a file.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# tftp -i localhost put /apps/helloWord/helloWord hello
sending file...
file transfer completed.
[root@sylixos_station:/root]# ls /tmp/
hello
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_TFTP_EN> 0, it is allowed to provide network functions and enable the tftp service, and this command will be included.

Function interface:

The “tftp” command is implemented via c language functions. The function prototype is:

```
static INT __tshellTftp (INT iArgC, PCHAR *ppcArgV);
```

5.2.26. tftpdpath – View or set the local path of the tftp server

Format:

```
tftpdpath
```

```
tftpdpath newpath
```

Explanation:

There are 2 kinds of usage for this command, used together with the parameter, the command is used to view the local route of the current tftp server; followed by “newpath”,

it is used to set up the local route of tftp server.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# tftpdpath
tftpd path: /tmp
[root@sylixos_station:/root]# tftpdpath /
[root@sylixos_station:/root]# tftpdpath
tftpd path: /
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_NETBIOS_EN> 0, it is allowed to provide network functions and enable the simple “netbios” name service, and this command will be included.

Function interface:

The “tftpdpath” command is implemented through c language functions. The function prototype is:

```
static INT __ __tshellNetTftpdPath (INT iArgC, PCHAR *ppcArgV);
```

5.2.27. vlan – Display, set and delete net interface’s VLANID

Format:

```
vlan
vlan set  netifanme
vlan clear  netifaname
```

Explanation:

There are three kinds of usage for this command. The “net” interface is displayed when there is no parameter; the option “set” is to set “VLAN ID”; and “clear” is to clear the “VLAN ID” of the responding interface.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

```

[root@sylixos_station:/etc]# vlan set en1 1
[root@sylixos_station:/etc]# vlan

INDEX  VLAN ID
-----
      1      1
[root@sylixos_station:/etc]# vlan clear en1
[root@sylixos_station:/etc]# vlan

INDEX  VLAN ID
-----
[root@sylixos_station:/etc]#

```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_VLAN_EN> 0, it is allowed to provide network functions and enable VLAN tools, and this command will be included.

Function interface:

The “vlan” command is implemented through c language functions. The function prototype is:

```
static INT __tshellVlan (INT iArgC, PCHAR *ppcArgV);
```

5.2.28. vpnclose – Delete a virtual network interface**

Format:

```
vpnclose netifname
```

Explanation:

This command is used to delete a specified virtual network interface.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

N/A.

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_VPN_EN> 0, it is allowed to provide network functions and enable the VNP service, and this command will be included.

Function interface:

The “vpnclose” command is implemented by a c language function. The function prototype is:

```
INT __tshellVpnClose (INT iArgC, PCHAR *ppcArgV)
```

5.2.29. vpnopen – Create a virtual network interface**

Format:

vpnopen configuration file

Explanation:

This command is used to create a virtual network interface.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

The configuration file should include the following information:

- ca certificate file name (.pem or .crt),
- Private certificate file name (.pem or .crt),
- Private key file (.pem or .crt),
- Private key file extract password,
- Server IP,
- VPN virtual network card address,
- VNP virtual network card mask,
- VPN virtual network card gateway,
- VNP server port,
- SSL communication timeout period,
- SSL authentication options,
- 6 bytes of virtual network card MAC address.

Example:

N/A.

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_NET_EN> 0 and LW_CFG_NET_VPN_EN> 0, it is allowed to provide network functions and enable the VNP service, and this command will be included.

Function interface:

The “vpnopen” command is implemented by a c language function. The function prototype is:

```
static INT __tshellVpnOpen (INT iArgC, PCHAR *ppcArgV);
```


6. Time

6.1. Introduction

- date – Display or set the current system time
- hwclock – Display or synchronize the operating system and hardware RTC clock
- times – Display utc or local time
- tzsync–Time zone synchronization with environment variable TZ

6.2. Use of commands

6.2.1. date – Display or set the current system time

Format:

date
date-stime | date

Explanation:

There are two kinds of usage for this command, the command without parameter is used to display the current system time; and parameter “-s” is used to set time or date, but only one of them can be set at a time.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# date
Mon Nov 28 09:11:36 2016
[root@sylixos_station:/root]# date -s 20161129
Tue Nov 29 09:11:41 2016
[root@sylixos_station:/root]# date -s 10:12:00
Tue Nov 29 10:12:00 2016
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide a tshell command, and this command is included.

Function interface:

The “date” command is implemented by a c language function. The function prototype is:

```
static INT __tshellSysCmdDate (INT iArgC, PCHAR *ppcArgV);
```

6.2.2 hwclock – Display or synchronize the operating system and hardware RTC clock

Format:

```
hwclock --show
hwclock --hctosys
hwclock --systohc
```

Explanation:

There are three kinds of usage for this command, the command with the parameter “-show” shows the hardware RTC clock; that with the parameter “-hctosys” synchronizes the RTC clock to the operating system clock; and that with the parameter “-systohc” synchronizes the operating system clock to the RTC clock.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# hwclock --show
Tue Nov 22 15:54:59 2016
[root@sylixos_station:/root]# hwclock --hctosys
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_RTC_EN > 0, it supports the command from version 0.9.7, and this command will be included.

Function interface:

The “hwclock” command is implemented by a c language function. The function prototype is:

```
static INT __tshellSysCmdHwclock (INT iArgC, PCHAR *ppcArgV);
```

6.2.3. times – Display utc or local time

Format:

```
times
times-utc
```

Explanation:

There are 2 kinds of usage for this command, the command without parameter is used to display local time; and that with the parameter “-utc” shows Greenwich time.

Return value:

Return “0” if the command is executed.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# times
Tue Nov 22 15:56:31 2016
[root@sylixos_station:/root]# times -utc
UTC : Tue Nov 22 07:56:37 2016
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide a tshell command, and this command is included.

Function interface:

The “times” command is implemented through c language functions. The function prototype is:

```
static INT __tshellSysCmdTimes (INT iArgC, PCHAR *ppcArgV);
```

6.2.4. tzsync – Time zone synchronization with environment

variable TZ

Format:

tzsync

Explanation:

This command synchronizes the time with the time zone of the environment variable TZ.

Return value:

Return “0” if the command is executed.

Remarks:

If the time zone of the environment variable TZ changes, the corresponding time also changes.

Example:

Change the environment variable TZ

```
[root@sylixos_station:/etc]# vi temp
TZ=CST-8:00:00
```

Load environment variables, and view the time before and after synchronization

```
[root@sylixos_station:/etc]# varload temp
environment variables load from temp success.
[root@sylixos_station:/etc]# times
Tue Nov 22 17:09:02 2016
[root@sylixos_station:/etc]# tzsync
[root@sylixos_station:/etc]# times
Tue Nov 22 16:09:11 2016
[root@sylixos_station:/etc]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide a tshell command, and this command is included.

Function interface:

The “tzsync” command is implemented by a c language function. The function prototype is:

```
static INT __tshellSysCmdTzsync (INT iArgC, PCHAR *ppcArgV);
```


7. Dynamic loading

7.1 Introduction

- debug – debug a process
- dlconfig – configure the operating parameters of dynamic linker
- leakchk – check the memory leak
- leakchkstart – start the memory leak tracker
- leakchkstop – close the memory leak tracker
- lsmod – view the information of all kernel modules loaded by the system
- modulegcov – generate the kernel module code file (*.gcda)
- modulereg – register a module
- modules – view the information of all kernel modules and process dynamic link libraries loaded by the system
- modulestat – view the information of one kernel module or dynamic link library file
- moduleunreg – unload a module

7.2 Use of commands

7.2.1. debug – debug a process

Format:

```
debug connect options program arguments
debug --attach connect options program arguments
```

Explanation:

This command is used to debug a process. The following parameters are connection mode, project name and parameter list successively, among which the connection mode may be network, serial port, terminal or attach.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

Debugging via network

```
[root@sylixos_station:/apps/test1]# debug localhost:1234 ./test1
[GDB]Waiting for connect...
```

Debugging via serial port

```
[root@sylixos_station:/apps/test1]# debug /dev/ttyS1 ./test1
[GDB]Serial device: /dev/ttyS1 115200,n,8,1
```

Debugging via terminal

```
[root@sylixos_station:/apps/test1]# debug terminal ./test1
[GDB]Serial device: terminal 115200,n,8,1
```

Debugging via attach mode

```
[root@sylixos_station:/apps/test1]# debug --attach localhost:1234 ./test1
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command will be included.

Function interface:

The debug command is implemented by a c language function. The function prototype is:

```
static INT gdbMain (INT argc, CHAR **argv).
```

7.2.2 dlconfig – configure the operating parameters of dynamic linker

Format:

```
dlconfig share en | dis
dlconfig refresh *
dlconfig refresh
```

Explanation:

This command has 2 usages, with the option share, it indicates enabling or disabling the dynamic linker; with option refresh, it indicates clearing the shared data information, and with * means only clearing the information of the system, otherwise clearing all.

Return value:

Return 0 after the execution.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/apps/test1]# dlconfig refresh *
[root@sylixos_station:/apps/test1]# dlconfig refresh
[root@sylixos_station:/apps/test1]# dlconfig share dis
[root@sylixos_station:/apps/test1]# dlconfig share en
[root@sylixos_station:/apps/test1]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MODULELOADER_EN> 0, the module loading service needs to be provided and this command will be included.

Function interface:

The dlconfig command is implemented by a c language function. The function prototype is:

```
static INT __tshellDIConfig (INT iArgC, PCHAR *ppcArgV).
```

7.2.3 leakchk – check the memory leak

Format:

leakchk

Explanation:

This command is used to print the memory leak tracking information.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/lib/modules]# leakchk
  HEAP      THREAD      TIME      ADDR      SIZE      PURPOSE
-----
kersys      t_tshell      Mon Nov 28 09:58:50 2016 30e13f88      32 __tshellHistorySave
total unfree segment: 1 size: 32
[root@sylixos_station:/lib/modules]#
```

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SHELL_HEAP_TRACE_EN>0, the shell heap trace tool is enabled and this command will be included.

Function interface:

The leakchk command is implemented by a c language function. The function prototype is:

```
static INT __tshellHeapCmdLeakChk (INT iArgC, PCHAR ppcArgV[]).
```

7.2.4 leakchkstart – start the memory leak tracker

Format:

leakchkstart max save node number pid

Explanation:

This command is used to start the memory leak tracker. The following parameters are the maximum number of tracked nodes and the process id.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

The maximum number of tracked nodes is 1024 at least.

The process id is 0 in default.

Example:

```
[root@sylixos_station:/lib/modules]# leakchkstart 1024 0
leakcheck start checking...
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SHELL_HEAP_TRACE_EN>0, the shell heap trace tool is enabled and this command will be included.

Function interface:

The leakchkstart command is implemented by a c language function. The function

prototype is:

```
static INT __tshellHeapCmdLeakChkStart (INT iArgC, PCHAR *ppcArgV).
```

7.2.5 leakchkstop – close the memory leak tracker

Format:

```
leakchkstop
```

Explanation:

This command is used to close the memory leak tracker.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# leakchkstop
```

HEAP	THREAD	TIME	ADDR	SIZE	PURPOSE
kersys	t_tshell	Mon Nov 28 10:03:30 2016	30e22e50	32	__tshellHistorySave
kersys	t_tshell	Mon Nov 28 10:02:58 2016	30bc8600	48	API_TShellOptInd
kersys	t_tshell	Mon Nov 28 10:02:58 2016	30bc85c8	32	__tshellHistorySave
kersys	t_tshell	Mon Nov 28 10:02:07 2016	30bc85a0	16	__tshellReadlineInit
kersys	t_tshell	Mon Nov 28 10:02:07 2016	30bc8578	16	API_ThreadCleanupPush
kersys	t_tshell	Mon Nov 28 10:02:07 2016	30bc8350	528	__tshellShowPrompt
kersys	t_tshell	Mon Nov 28 10:02:07 2016	30e22c18	544	_rngCreate
kersys	t_tshell	Mon Nov 28 10:02:07 2016	30e229e0	544	_rngCreate
kersys	t_tshell	Mon Nov 28 10:02:07 2016	30e227b8	528	lib_malloc
kersys	t_tshell	Mon Nov 28 10:02:07 2016	30e22590	528	_IosEnvCreate
kersys	t_tshell	Mon Nov 28 10:02:07 2016	30e22568	16	API_ThreadCleanupPush
kersys	t_except	Mon Nov 28 10:02:07 2016	30e22230	800	__selTaskCreateHook

```
total unfree segment: 12 size: 3632
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SHELL_HEAP_TRACE_EN>0, the shell heap trace tool is enabled and this command will be included.

Function interface:

The leakchkstop command is implemented by a c language function. The function prototype is:

```
static INT __tshellHeapCmdLeakChkStop (INT iArgC, PCHAR *ppcArgV).
```

7.2.6 Ismod – view the information of all kernel modules

loaded by the system

Format:

```
Ismod
```

Explanation:

This command has 1 usage, i.e. to display the information of all kernel modules loaded by the system.

Return value:

Return 0 after the execution.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# lsmod
-----
NAME                HANDLE  TYPE  GLB  BASE  SIZE  SYMCNT
-----
VPROCESS: kernel    pid:   0 TOTAL MEMORY: 0

total modules: 0
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MODULELOADER_EN> 0, the module loading service needs to be provided and this command will be included.

Function interface:

The lsmod command is implemented by a c language function. The function prototype is:

```
static INT __tshellLsmod (INT iArgC, PCHAR *ppcArgV).
```

7.2.7 modulegcov – generate the kernel module code file

(*gcda)

Format:

```
modulegcov kernel module handle
```

Explanation:

This command is used to generate the kernel module code file.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# modules
-----
NAME                HANDLE  TYPE  GLB  BASE  SIZE  SYMCNT
-----
VPROCESS: kernel    pid:   0 TOTAL MEMORY: 8192
+ file.ko           30e14be8 KERNEL YES 60005000      940      1

total modules: 1
[root@sylixos_station:/root]# modulegcov 30e14be8
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MODULELOADER_EN> 0 and LW_CFG_MODULELOADER_GCOV_EN>0, the module loading service needs to be provided and the kernel module code coverage analysis interface is allowed. This command will be included.

Function interface:

The modulegcov command is implemented by a c language function. The function prototype is:

```
static INT __tshellModuleGcov (INT iArgC, PCHAR *ppcArgV).
```

7.2.8 moduleunreg – register a module

Format:

```
modulereg kernel module file *.ko
```

Explanation:

This command has 1 usage, i.e. to register a module file.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# modulereg /lib/modules/file.ko
module /lib/modules/file.ko register ok, handle: 0x30e15370
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MODULELOADER_EN> 0, the module loading service needs to be provided and this command will be included.

Function interface:

The modulereg command is implemented by a c language function. The function prototype is:

```
static INT __tshellModuleReg (INT iArgC, PCHAR *ppcArgV).
```

7.2.9 modules – view the information of all kernel modules and process dynamic link libraries loaded by the system

Format:

```
modules
modules module name
```

Explanation:

This command has 1 usage, no parameter to display the information of all kernel modules and process dynamic link libraries; with parameters to view the information of specified kernel module and process dynamic link library.

Return value:

Return 0 after the execution.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# modules
```

NAME	HANDLE	TYPE	GLB	BASE	SIZE	SYMCNT

VPROCESS: kernel	pid: 0	TOTAL MEMORY: 16384				
+ interruptK.ko	30e13f98	KERNEL	YES	60006000	28c	2
+ file.ko	30e14be8	KERNEL	YES	60005000	940	1

total modules: 2						

```
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands, and this command will be included.

Function interface:

The modules command is implemented by a c language function. The function prototype is:

```
static INT __tshellModuleShow (INT iArgC, PCHAR *ppcArgV).
```

7.2.10 modulestat – view the information of a kernel module or dynamic link library file

Format:

```
modulestat program file
```

Explanation:

This command is used to view the information of a kernel module or dynamic link library file.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# modulestat /lib/modules/file.ko
```

```
File Type: ELF
```

```
Machine: ARM family
```

```
Type: ET_REL
```

```
Entry: 0
```

```
Section Headers:
```

TYPE	ADDRESS	OFFSET	SIZE	FLAGS
NULL	00000000	0	0	
PROGBITS	00000000	34	7c4	[ALLOC] [EXEC]
REL	00000000	dc4	178	
PROGBITS	00000000	7f8	78	[ALLOC]
PROGBITS	00000000	870	a	[ALLOC] [WRITE]
NOBITS	00000000	87c	94	[ALLOC] [WRITE]
PROGBITS	00000000	87c	79	
NONE	00000000	8f5	2f	
STRTAB	00000000	924	51	
SYMTAB	00000000	b30	1a0	
STRTAB	00000000	cd0	f4	

```
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MODULELOADER_EN> 0, the module loading service needs to be provided and this command will be included.

Function interface:

The modulestat command is implemented by a C language function. The function prototype is:

```
static INT __tshellModulestat (INT iArgC, PCHAR *ppcArgV).
```

7.2.11 moduleunreg – unload a module

Format:

```
moduleunreg kernel module handle
```

Explanation:

This command has 1 usage, i.e. to unload a module.

Return value:

Return “0” on success and non-“0” on failure.

Example:

N/A.

Example:

```
[root@sylixos_station:/root]# modules
-----
NAME                HANDLE  TYPE  GLB  BASE      SIZE  SYMCNT
-----
VPROCESS: kernel    pid:   0 TOTAL MEMORY: 16384
+ interruptK.ko      30e13f98 KERNEL YES 60006000    28c    2
+ file.ko            30e14be8 KERNEL YES 60005000    940    1

total modules: 2
[root@sylixos_station:/root]# moduleunreg 30e13f98
hello_module exit!
module /lib/modules/interruptK.ko unregister ok.
[root@sylixos_station:/root]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_MODULELOADER_EN> 0, the module loading service needs to be provided and this command will be included.

Function interface:

The moduleunreg command is implemented by a c language function. The function prototype is:

```
static INT __tshellModuleUnreg (INT iArgC, PCHAR *ppcArgV).
```


8. Others

8.1 Introductoin

- args – display all input parameters, with spaces as delimiters
- crypt – encrypt data
- perfrefresh – update performance statistics
- perfs – display performance statistics
- perfstart – start the performance analysis tool
- perfstop – stop the performance analysis tool
- xmodemr - receive a file using the xmodem protocol
- xmodems - send a file using the xmodem protocol

8.2 Use of commands

8.2.1 args – display all input parameters, with spaces as delimiters

Format:

args [any argument...]

Explanation:

This command is used to display the input parameters in a certain format.

Return value:

Return 0 after the execution.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# args
arg 1 is args
[root@sylixos_station:/root]# args SylixOS system is gool
arg 1 is args
arg 2 is SylixOS
arg 3 is system
arg 4 is is
arg 5 is gool
[root@sylixos_station:/root]#
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The args command is implemented by a c language function. The function prototype is:

```
static INT __tshellSysCmdArgs (INT iArgC, PCHAR ppcArgV[]).
```

8.2.2 crypt – encrypt data

Format:

crypt key salt

Explanation:

This command is used to display the input parameters in the form of key-value pairs. Where key is the plaintext to be encrypted, and salt specifies the key to use for encryption.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/etc]# crypt root root
roK20XGbWEsSM
[root@sylixos_station:/etc]# crypt root 2root
2rUSJaaFNwmv6
[root@sylixos_station:/etc]# crypt root 1root
1r9MzDzsjs4uM
[root@sylixos_station:/etc]#
```

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SHELL_PASS_CRYPT_EN>0, the user password related support is provided and this command will be included.

The crypt command is implemented by a c language function. The function prototype is:

```
static INT __tshellSysCmdCrypt (INT iArgC, PCHAR ppcArgV[]).
```

8.2.3. perfrefresh – update performance statistics

Format:

perfrefresh

Explanation:

This command is used to update the information statistics.

Return value:

The return value is 0.

Remarks:

N/A.

Example:

N/A.

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SYSPERF_EN> 0 and LW_CFG_SHELL_PERF_TRACE_EN >0, the system performance analysis is allowed and the system performance analysis tool is enabled, and this command will be included.

The perfrefresh command is implemented by a c language function. The function prototype is:


```
static INT __tshellPerfCmdPerfRefresh (INT iArgC, PCHAR ppcArgV[]).
```

8.2.4. perfs – display performance statistics

Format:

perfs

Explanation:

This command is used to display the performance statistics. The parameter is pipe buffer size (128~4096).

Return 0 after the execution.

Remarks:

N/A.

Example:

N/A.

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SYSPERF_EN> 0 and LW_CFG_SHELL_PERF_TRACE_EN >0, the system performance analysis is allowed and the system performance analysis tool is enabled, and this command will be included.

The perfs command is implemented by a c language function. The function prototype is:

```
static INT __tshellPerfCmdPerfShow (INT iArgC, PCHAR ppcArgV[]).
```

8.2.5. perfstart – start the performance analysis tool

Format:

perfstart pipe buffer len performance save node refresh period

Explanation:

This command is used to display the performance statistics. The parameters are pipe buffer size (128~4096), the performance saved node (10~30), and the update period (greater than or equal to 1s).

Return value:

Return “0” on success and “-1” on failure.

Remarks:

pipe buffer len	: default 1024
performance save node	: default 20
refresh period	: default 10000

Example:

N/A.

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SYSPERF_EN> 0 and LW_CFG_SHELL_PERF_TRACE_EN >0, the system performance analysis is allowed and the system performance analysis tool is enabled, and this command will be included.

The perfstart command is implemented by a c language function. The function prototype is:

```
static INT __tshellPerfCmdPerfStart (INT iArgC, PCHAR ppcArgV[]).
```

8.2.6. perfstop – stop the performance analysis tool

Format:

```
perfstop
```

Explanation:

This command is used to display the input parameters in the form of key-value pairs.

Return value:

Return “0” on success and “-1” on failure.

Remarks:

N/A.

Example:

N/A.

Configuration:

When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands. When LW_CFG_SYSPERF_EN> 0 and LW_CFG_SHELL_PERF_TRACE_EN >0, the system performance analysis is allowed and the system performance analysis tool is enabled, and this command will be included.

The args command is implemented by a c language function. The function prototype is:

```
static INT __tshellSysCmdArgs (INT iArgC, PCHAR ppcArgV[]).
```

8.2.7. xmodemr - receive a file using the xmodem protocol

Format:

```
xmodemr file path
```

Explanation:

This command is used to receive a file remotely using the xmodem protocol.

Return value:

Return “0” on success and non-“0” on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/apps/hello]# xmodems hello
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

The xmodemr command is implemented by a c language function. The function prototype is:

```
static INT __tshellFsCmdXmodemr (INT iArgC, PCHAR ppcArgV[]).
```

8.2.8. xmodems - send a file using the xmodem protocol

Format:

xmodems file path

Explanation:

This command is used to send a file to remote device using the xmodem protocol.

Return value:

Return "0" on success and non-"0" on failure.

Remarks:

N/A.

Example:

```
[root@sylixos_station:/root]# xmodems /apps/helloWord/helloWord
```

Configuration:

This command belongs to the system-supplied tshell command. When LW_CFG_SHELL_EN>0, the operating system is allowed to provide tshell commands.

Function interface:

The xmodems command is implemented by a c language function. The function prototype is:

```
static INT __tshellFsCmdXmodems (INT iArgC, PCHAR ppcArgV[]).
```