VxWorks

Copyright © 2007 – 2012 Ralf Hoppe

1 „C" Interpreter[1]

1.1 Specials

cmd Switch to command interpreter
emacs Turn on Emacs mode
Ctrl+Break Interrupt shell command
...Ctrl+D Matching symbols
...TAB Symbol expansion
cmd **Ctrl+W** HTML-help for *cmd* (host shell)
@*cmd* Execute cmd in host shell, z. B. @cd "/tffs0/"

1.2 General

help Print synopsis of selected command
printErrno # Explain error number #
?shConfig Show/Set shell config
bootParamsShow Show boot parameters
bootChange Modify boot parameters
reboot Reboot target system

1.3 Debug

browse Browse an address
d [*adr***][,***width***][,***num***]** Show memory
m [*adr***][,***width***][,***num***]** Modify memory
l [*adr***]** Disassemble memory
mRegs Modify registers
b [*adr***]** Set/Show breakpoint(s)
bd [*adr***]** Delete breakpoint
bdall Delete all breakpoints
c [*name/id***]** Continue (task *name/id*)
s [*name/id***]** Step into (task *name/id*)
so [*name/id***]** Step over (task *name/id*)
cret [*name/id***]** Step out (task *name/id*)

1.4 Modules & Symbols

ld [*mod***]** Load object module
unld [*mod***]** Unload object module
moduleShow Show status of modules
lkup *sym* List address of symbol *sym*
lkAddr [*adr***]** List symbols near *adr*

1.5 Tasks

i Print task info from TCB
sp [*adr***][,***args***]** Spawn a task[2]
sps Spawn a task, set then suspended
ts *name/id* Suspend a task
td *name/id* Delete a task
tr *name/id* Resume a suspended task
w Task's pending information
tw *name/id* Object info task is pending on
tt Show stack trace
checkStack Print task stack usage
spy, spyStop, spyHelp Processor usage

1.6 Resources

vmContextShow Show memory config (MMU)
memShow Show system memory partition
memPartShow *partid* Show partition blocks of *partId*
pmShow Show „persistent" memory info
msgQShow *msgQId* Show message queue *msgId* info
semShow *semId* Show info about semaphore *semId*
wdShow *wdId* Show info about watchod *wdId*
classShow *class* Show info about object *class*[3]
envShow Show environment info
edrHelp Shows all available ED&R commands
edrShow Show ED&R error log

1.7 Real Time Processes (RTP)

syscallShow [*grp, level***]** Shows all system calls
rtpShow [*name/id, level***]** Show RTP(s) info
shlShow/rtpShlShow Show shared libraries info
rtpMemShow *name/id* **[,** *level***]** Show RTP memory info
objHandleTblShow *rtpId* Show all objects in handle table
objShow *objId* Show information on an object
iosRtpFdShow Show the RTP file descriptor table
rtpLkup *sym, id* List address of RTP *id* symbol *sym*[4]
rtpLkAddr *adr, id* List RTP *id* symbols near *adr*
rtpi *rtpId* Show all tasks in RTP *rtpId*
rtpSp **"<***path***>/myVxApp.vxe [***args***]"** Execute RTP

1.8 I/O-System[5]

devs List all system-known devices
iosDrvShow Show all system drivers
iosFdShow Show all file descriptor names
logFdSet *fd* Changes file descriptor of *logMsg* to *fd*[6]
tffsShowAll Show TFFS socket information
dosFsShow *dev* Show DOS filesystem info's for *dev*
logShow Show log info
loggerInfoShow Show log info

1.9 Network

netHelp Show network helper functions
nfsHelp Show NFS helper functions
hostShow Show the remote host table
unstatShow Show AF_LOCAL/COMP sockets
netPoolShow *poolId* Show pool statistics
netStackSysPoolShow Show network system pool[7]
netStackDataPoolShow Show network data pool
muxShow [*devName***][,***unit***]** Show interfaces info
ifShow [*ifName***]** Show attached network interfaces
mbufShow Show mbuf statistics
mRouteShow Show all IP routes
arpShow Show ARP table
icmpstatShow Show ICMP (Ping) statistics

---

[1]Especially for target (kernel) shell. The host shell additionally offers GDB and TCL commands.
[2]Default settings can be controlled via variables spTaskPriority, spTaskStackSize, spTaskOptions.
[3]lkup "ClassId" shows available classes
[4]If sym is 0 prints a short summary of symbol table, if it is "" lists all RTP symbols.
[5]For commands as cd, pwd, rm, (x)copy and friends see usrFsLib. For all available commands see the Host Shell Users Guide.
[6]Typically used as "logFdSet 3", which assigns the file descriptor of /tyCo/0.
[7]netStackSysPoolShow and netStackDataPoolShow typically expand to netPoolShow(_pNetSysPool) and netPoolShow(_pNetDpool).

inetstatShow Show active connections

ipstatShow Show IP statistics

tcpStatShow Show TCP statistics

udpStatShow Show UDP statistics

routestatShow Show routing statistics

1.10 BSP

2 Command Interpreter

C Switch to „C" interpreter

ifconfig Interface configuration

netstat Shows network connections

syslog System event log[8]

ping Sends ICMP Echo Requests

arp Manipulate the ARP cache

sysvar Reads/Writes system IPNET variables

ike IPsec IKE status

ipsecctrl IPsec Control

ipf Firewall

ipd Interpeak daemon

rtp Real Time Process commands

3 Platform

3.1 Environment

WIND_HOME Installation directory

WIND_BASE Base directory of VxWorks

WIND_HOST_TYPE Host type, z. B. `x86-win32`

WIND_PLATFORM Platform, z. B. `vxworks-6.5`

WIND_COMPONENTS Components path

TOOL Compiler, z. B. `diab`, `sfdiab`, `gnu`, `gnube`, ...

TOOL_FAMILY Compiler-Familie (`diab`, `gnu`)

CPU Processor, z. B. `PPC603`, `PPC32`, `XSCALE`, `ARM`, `SPARC`, ...

3.2 Build Variables (-D)

VX_OS_LIBS OS libraries in `LD_LINK_PATH` and `$(TGT_DIR)/lib/lib$(CPU)$(TOOL)vx.a`

LIBS Customer's libraries and `VX_OS_LIBS`

LD_LINK_PATH `ld` search path[9]

TGT_DIR `$(WIND_BASE)/target`

_WRS_KERNEL Kernel build

__RTP__ RTP build

3.3 Project Build Macros

CFLAGS_PROJECT Common CFLAGS (all build specs)

3.4 Build

1. `make -C %WIND_BASE%/target/[usr/]src CPU=SPARC TOOL=sfdiab [ADDED_CFLAGS+=...] rclean`
2. `make -C %WIND_BASE%/target/[usr/]src CPU=SPARC TOOL=sfdiab OPT=-fr [ADDED_CFLAGS+=...]` [10]
3. `make -C %WIND_BASE% CPU=XSCALE TOOL=gnube [FEATURE_SET=pne] (comp-kernel|comp-usr) TARGET=rclean`[11]
4. `make -C %WIND_BASE% CPU=XSCALE TOOL=gnube [FEATURE_SET=pne] (comp-kernel|comp-usr) ADDED_CFLAGS+="-DIP_DEBUG -g" ADDED_C++FLAGS+="-DIP_DEBUG -g"`

3.5 Files

Kernel mode config:	`$(WIND_BASE)/config/(pne|pad|pcd|pid)/config.mk`[12]
User mode config:	`$(WIND_BASE)/config/(pne|pad|pcd|pid)/configRtp.mk`
Architecture compile flags:	`$(WIND_BASE)/target/[usr/]h/tool/$(TOOL_FAMILY)/defs.$(CPU)`
Compiler family flags:	`$(WIND_BASE)/target/[usr/]h/tool/$(TOOL_FAMILY)/defs.$(TOOL_FAMILY)`
Architecture/Family Makefile:	`$(WIND_BASE)/target/[usr/]h/tool/$(TOOL_FAMILY)/make.$(CPU)$(TOOL)`[13]
Common makefiles:	`$(WIND_BASE)/target/[usr/]h/make`
statTbl makefile:	`$(WIND_BASE)/target/src/usr/Makefile`[14]
Components objects/libraries:	`$(WIND_COMPONENTS)/obj`
VxWorks objects/libraries:	`$(WIND_BASE)/target/lib/$(CPU), $(WIND_BASE)/target/usr/lib/$(CPU)`
Environment variables:	`$(WIND_HOME)/install.properties`
Workbench config:	`workspace/.metadata/.plugins/org.eclipse.ui.workbench/workbench.xml`

---

[8]For *severities* and *facilities* see „ipcom_syslog_data.c".

[9]Includes the tool independant and dependant directories, as well as any additional directory specified via LD_LINK_PATH_EXTRA.

[10]usr ≡ user space libraries, -fr ≡w/o *WindView* instrumentation (`-D_FREE_VERSION -UWV_INSTRUMENTATION`)

[11]sf ≡ software floating point

[12]pad ≡ Platform for Automotive Device, pne≡ Platform Networking Equipment, pcd≡ Platform for Consumer Devices, pid≡ Platform for Industrial Devices

[13]Includes file `defs.$(TOOL_FAMILY)`.

[14]`perror`, `printErrno` database makefile (see VxWorks Kernel API Reference Vol. 1, errnoLib)