

VxWorks Programmierung

System und Anwendungen

Inhaltsverzeichnis

1	History	1
2	VxWorks 5.3 für MVME CPU	2
2.1	VxWorks Image Build	2
2.2	VxWorks Boot	2
2.3	Dokumentation 5.3	3
3	VxWorks 21.07 für Intel CPU	4
3.1	VxWorks Installation	4
3.2	VxWorks Lizenzserver	5
3.3	VxWorks Image Build	5
3.3.1	VxWorks Build mit Workbench	5
3.3.1.1	VxWorks VSB Project	6
3.3.1.2	VxWorks VIP Project	7
3.4	Boot Loader Build	9
3.4.1	UEFI Boot Loader	9
3.4.2	VxWorks Boot Loader	9
3.5	VxWorks Boot	10
3.6	Anwendungen	11
3.6.1	Programmquellen	11
3.6.2	Kompilierung	12
3.6.3	Dokumentation	12
3.6.4	Makefiles	12
3.7	VxWorks 21.07 Dokumentation und Support	13
3.8	Gut zu wissen...	14
3.8.1	Login	14
3.8.2	Kernel Shell	14
3.9	Bekannte Probleme	16
3.9.1	Start-Menü	16
3.9.2	Spy-App	16
3.10	Anhang	17
3.10.1	Parameter für PC-Console	17
3.10.2	Telnet mit Tera Term	17
3.10.3	Telnet mit xTerm	18
3.10.4	Navigating in VI	18
3.10.5	Intel CPU Generationen	18
3.10.6	VIP Includes und Parameter für das Core-Projekt	19
4	Neues Experiment	23
4.1	System-Einstellungen	23
4.2	Account-Konfiguration	24

1 History

VxWorks System

VxWorks 5.1 auf DECstation 5000/33 unter Ultrix (TUTNIX)

Tornado 1.0 auf PC unter Windows 98 (???????)

> in VBox unter Windows 7 (RADON)

> in VBox unter Windows 10 (RADON)

> in VBox unter Debian 11 (IONIX)

VxWorks 21.07 unter Ubuntu 1804 in VBox unter Debian 11 (IONIX)

VxWorks Anwendungen

VxWorks 5.1 auf DECstation 5000/33 unter Ultrix (TUTNIX)

VxWorks 5.3 auf DECstation 5000/33 unter Ultrix (TUTNIX)

!!!Baustelle!!!

2 VxWorks 5.3 für MVME CPU

2.1 VxWorks Image Build

Die Software zum Erzeugen von VxWorks 5.3 Systemen ist auf Windows 98 installiert. Dieses Win98 hat alle Systemwechsel in einer VBox überlebt (z.Z. auf huber@radon und huber@ionix).

Die Anpassung des VxWorks wurde durch Editieren der Konfigurationsdateien vorgenommen. Die Dokumentation dazu liegt in gedruckter Form vor (s.u.).

Alternativ kann durch die Systemgenerierung mittels WindRiver-Tornado geführt werden.

Für die vorhandenen, verschiedenen MVME-Laborrechner existieren bootbare VxWorks Images:

```
/usr/exp/vxw/MVME/
vx162          Standard für Motorola MVME162
vx162no_nfs    ohne NFS-Transfers
vx162no_passw  ohne Passwort bei Netzwerk-Logins
vx172          Standard für Motorola MVME172, MVME172P
vx172_16MB     mit 16 MB Memory
vx172no_nfs    ohne NFS-Transfers
vx172no_passw  ohne Passwort bei Netzwerk-Logins
*.sym          die zugehörigen Symbol Tables
```

2.2 VxWorks Boot

Der Boot Loader ist auf einem Boot-Rom installiert. Die Boot Parameter sind auf einem NV-RAM abgelegt. Nach Abbrechen des Boot können Sie mit dem Kommando c editiert werden.

Boot-Parameter für das 017-Experiment nach dem Umzug in die Physik:

```
boot device      ei
processor number 0
host name        boothost
file name        ~/ex_home/vxw/MVME/vx172
inet on ethernet (e) 134.176.21.35:ffffff00
inet on backplane (b)
host inet (h)      134.176.20.18
gateway inet (g)   134.176.21.1
user (u)          exeis
ftp password (pu)
flags (f)         [0x0]
target name (tn)   exeis
startup script (s) ex_home/ex_param/Startup.vxw
other (o)
```

2.3 Dokumentation 5.3

Die Dokumentation für VxWorks 5.3 liegt in gedruckter Form vor:

VxWorks Programmer's Guide 5.3

VxWorks Reference Manual 5.3

GNU ToolKit User's Guide 2.6

Und ist teilweise bei Wind River noch zu finden:

<https://docs.windriver.com>

> Discontinued Products (ganz unten)

> VxWorks

> VxWorks 5.3

3 VxWorks 21.07 für Intel CPU

Für die VxWorks-Systemgenerierung und die Entwicklung der Anwendungsprogramme stehen zur Zeit die Versionen 2103 und 2107 zur Verfügung. Da die VxWorks-Entwicklung nicht mit Debian verträglich ist, wurde sie in einer VBox unter Ubuntu installiert, unter

```
VXHOME/WindRiver2103
VXHOME/WindRiver2107
```

Zur Zeit mit

```
VXHOME = /home/huber
VXHOME/WindRiver --> VXHOME/WindRiver2107
```

in der VBox: huber-virtualbox

3.1 VxWorks Installation

Dear Justus-Liebig-University Giessen,

Thank you for your recent purchase of a Wind River solution. This email will be your guide for:

1. Registering for a Wind River support account
<https://wrsn.windriver.com/accounts/faces/userRegistration>
2. Downloading, installing, and activating Wind River software
<https://wrsn.windriver.com/esd/faces/esd.jsp?orderNo=4516827&val2567277-101-PU7W-AMMLZ>
3. Managing product licenses
<http://www.windriver.com/licensing>
4. Accessing product documentation
<https://knowledge.windriver.com/en-us>
5. Contacting support
https://knowledge.windriver.com/en-us/Wind_River_Support_Network

Please forward this email to other entitled end users of Wind River software to access any of the products purchased in this order. Each person will need to create their own personal Wind River support account using the information in this email.

Order Information

- Customer Name: Justus-Liebig-University Giessen
- Sales Order: 4516827
- Purchase Order: Uni Grant VxWorks
- License Number: 683907
- License Authorization Code (LAC): 683907-5FKR-XNJEY
- License Administration Token (LAT): YCTUOKY-683907-9OCYOFX
- Support Contract Number:
- Support Contract End Date:

3.2 VxWorks Lizenzserver

Für die Systemgenerierung und die Entwicklung der Anwendungen ist eine Lizenz erforderlich. Wir haben zur Zeit eine kostenlose Uni-Lizenz für VxWorks 2103 und 2107, die gelegentlich wieder verlängert werden muss. Dabei sollten wir uns mal nach einem Upgrade erkundigen...

Der laufende Betrieb von VxWorks und Anwendungen erfolgt lizenzfrei.

Der Lizenzserver ist lokal installiert, zur Zeit unter WindRiver2103. Er vergibt bis zu 8? Lizenzen gleichzeitig. Er muss vor Beginn der Arbeiten gestartet werden. Für Start und Stop gibt es zwei Scripts:

```
VXHOME/WindRiver2103/license/startFlexNet  
VXHOME/WindRiver2103/license/stopFlexNet
```

3.3 VxWorks Image Build

Zum Erstellen eines VxWorks Betriebssystems stehen zwei Wege zur Verfügung:

- Über vxprj-Kommandozeilen
Diese Variante hat den Vorteil, dass man alles in einem Shell-Script zusammenfassen kann und damit alle Parametereinstellungen gleich dokumentiert hat. Der Nachteil: die passenden Parameter muss man erst wissen.
Beispiele dazu findet man unter: VXHOME/WindRiver2103/workspace
- Menügeführt mit dem Workbench-Programm
Der Vorteil von Workbench ist, dass alle Parameter zur Auswahl stehen. Der Nachteil: es sind schrecklich viele Parameter und am Ende hat man den Überblick verloren, welche man verändert hat.

Eine VxWorks-Generierung erfolgt in zwei getrennten Schritten:

- VxWorks Source Bild (VSB)
In diesem ersten Schritt wird festgelegt welche CPU unterstützt werden soll und welche Bibliotheken gebraucht werden. Die Bibliotheken werden kompiliert.
- VxWorks Image Project (VIP)
In diesem zweiten Schritt werden die notwendigen Parameter angepasst und das fertige, bootbare Image erzeugt.

Jeder der beiden Schritte kann wiederholt werden um Änderungen einzubringen.

3.3.1 VxWorks Build mit Workbench

Das Arbeiten mit Workbench verlangt einige Erfahrung. Deshalb vorab ein kurze Einführung in die Bedienung um VSB und VIP Projekte zu erstellen.

- Starten Sie ggf. den Lizenzserver.
- Starten Sie Workbench4 über den

Workbench(2107) Icon

oder das Kommando

```
VXHOME/WindRiver2107/workbench-4/startWorkbench.sh
```

- Starten Sie Ihr Projekt mit

File -> New -> Wind River Workbench Project...

In diesem Menüfenster wählen Sie den Projekttyp, also VSB bzw VIP

3.3.1.1 VxWorks VSB Project

Im VxWorks-Source-Build-Projekt werden die notwendigen Bibliotheken zusammengestellt und kompiliert.

Die Anpassungen an den CPU-Typ und das CPU-Board erfolgen über das Board Support Package (BSP) und die CPU-Typangabe.

Für Intel-Prozessoren gemeinsam ist das BSP itl-generic.

Mit der Wahl von CORE (Nachfolger von Pentium, Vorgänger von CORE-i, Empfehlung vom WR-Support) als CPU erhält man ein VxWorks-Image, das auch für den OptiPlex 5000 (12. Generation) im Labor 017 und etliche ältere geeignet ist. Dies in der Hoffnung, dass ein Rechnertauch ohne große Probleme erfolgen kann. Neuere Entwicklungen bleiben dabei jedoch möglicherweise ungenutzt. Mehr Info zu diesem Thema findet man unter

VXHOME/WindRiver2107/vxworks/21.07/os/board/intel/itl-generic/*_readme.md

und der Liste der Intel-CPU-Generationen im Anhang.

Nachtrag:

Sandy-Bridge (2. Gen.) und Sky-Lake (6. Gen.) sind ebenfalls für den OptiPlex 5000 geeignet, möglicherweise aber nicht für ältere Generationen.

Starten Sie das VSB-Projekt mit:

File > New > Wind River Workbench Project...

Im Menüfenster wählen Sie den Projekttyp: VxWorks Source Build

Next >

Projekt name: Core-VSB (z.B.)

Next >

Based on: a board support package (BSW)

BSW: itl-generic....

CPU: CORE (z.B.)

Processor mode: SMP support

Finish >

Anschließend können Ergänzungen vorgenommen werden. Der Default enthält bereits weitgehend alle benötigten Funktionen:

Öffnen Sie Ihr VSB-Projekt in der Projektliste

Doppelklicken Sie auf: Source Build Configuration

Folgende Layer wurden ergänzt:

Source Build Configuration

- (find: user)
Enable user management libraries: Y
- (find: editor)
Enable vi editor libraries: Y

Im letzten Schritt werden die Bibliotheken kompiliert:

rechte Maustaste auf Ihr VSB-Projekt > Build

Das dauert...

3.3.1.2 VxWorks VIP Project

Im VxWorks-Image-Build-Projekt wird das Image konfiguriert und anschließend fertig gestellt. Die BSP-Parameter sind bereits voreingestellt, es gibt aber noch viel zu tun...

Starten Sie das VIP-Projekt mit:

```
File > New > Wind River Workbench Project...
  Im Menüfenster wählen Sie den Projekttyp: VxWorks Image Project
Next >
  Project name: Core-VIP (z.B.)
Next >
  Based on: a Source Build Project (VSB)
  Projekt: Core-VSB (z.B.)
Next >
  Profile: (no profile)
  ggf. verwenden eines zuvor erstellten Profiles eines bereits
  passend konfigurierten VIP (z.B. PROFILE_CORE-VIP).
Finish >
```

Anschließend müssen die VIP-Parameter angepasst werden. Dazu öffnen Sie Ihr VIP-Projekt im Project Explorer und Doppelklicken auf: Kernel Configuration.

Wenn Sie einen VIP-Profile verwendet haben (s.o.), dann können Sie in der Datei

```
VXHOME/WindRiver2107/vxworks/21.07/os/vip_profiles/profiles/
vip_profile_PROFILE_CORE-VIP.cdf (z.B.)
```

sehen, mit welchen Includes und Parameter die BSP-Einstellungen durch das Profile ergänzt wurden. Das Core-VIP Profile finden Sie auch im Anhang.

Für das Core-VIP wurden folgende Parameter angepasst (mit 'Synopsys' erhalten Sie eine Beschreibung der Parameter):

- (find: TELNET_IDLE)
IPCOM_TELNET_IDLE_TIMEOUT "0"
- (find: TIMEZONE)
TIMEZONE_ENV_VAR "TZ=CET-1CEST,M3.5.0/02,M10.5.0/03"
- (find: VX_SMP)
VX_SMP_NUM_CPUS VX_MAX_SMP_CPUS
- (find: SHELL_COMP)
SHELL_COMPATIBLE FALSE
- (find: SHELL_DEFAULT)
SHELL_DEFAULT_CONFIG ("INTERPRETER=,LINE_EDIT_MODE=emacs,
LINE_LENGTH=1024,STRING_FREE>manual,
VXE_PATH=.;/bin;/usr/bin;/romfs")
Command-line Editor: emacs
- (find: DNSC_DOM)
DNSC_DOMAIN_NAME "IAMP"

- (find: DNSC_PRIM)
DNSC_PRIMARY_NAME_SERVER "134.176.2.7"
- (find: DNSC_SEC)
DNSC_SECONDARY_NAME_SERVER "134.176.2.5"
- (find: NTP_INIT)
NTP_INIT_PARAM NULL
- (find: NTP_CONF)
NTP_CONFIG_PARAM NULL
- (find: NTP_CONF)
NTP_CONFIG_FILENAME NULL
- (find: NO_SEC)
SYSFLG_NO_SECURITY_ENABLE FALSE
- (find: UDB_HASH)
UDB_HASH_KEY
256 Byte Hash-Code, notwendig für UserDataBase
- (find: UDB_STOR)
UDB_STORAGE_PATH "/ram0/cxUserDB.txt"
UserDataBase
- (find: UDB_PROM)
UDB_PROMPT_INITIAL_USER TRUE
Allerersten User anfordern, falls noch keiner da
- (find: SYS_CLK_RATE)
SYS_CLK_RATE (100)
- (find: RING_BUFFER)
USB_GEN2_SERIAL_IOS_RING_BUFFER_SIZE 131072
!!! notwendige Anpassung um Datenverlust zu vermeiden !!!

Im letzten Schritt wird das VxWorks Image erstellt:

rechte Maustaste auf Ihr VIP-Projekt > Build

Das bootbare VxWorks Image finden Sie z.B. unter:

VXHOME/WindRiver2107/workspace/Core-VIP/default/vxWorks

Damit außerhalb der VBox unter /usr/exp/vxw/INTEL darauf zugegriffen werden kann, muss es dahin noch übertragen werden. Durch die mehrfachen Wechsel der Unix-Systeme, ist dies zur Zeit ein ziemlich verschlungener Pfad:

cp vxWorks /media/sf_servix/usr2s/users/huber/ex_tools/vxw/INTEL/vxWorks-Core

Wobei /media/sf_servix/ das Shared Folder /home/servix mit dem VBox-Host (z.Z. Ionix) ist.

Weitere Hilfen:

Der Makefile zeigt die Details:

VXHOME/WindRiver2107/workspace/Core-VIP/Makefile.mk

Mit Workbench lassen sich Log-Files erstellen:

Help -> Collect log Files

3.4 Boot Loader Build

3.4.1 UEFI Boot Loader

VxWorks bietet mehrere unterschiedliche Boot-Möglichkeiten. Für die Zukunft ist jedoch UEFI vermutlich eine gute Wahl.

Die UEFI Boot Loader sollten bereits kompiliert sein:

Für 32 Bit CPU:

```
VXHOME/WindRiver2107/vxworks/21.07/workspace/uefi_ia32/BOOTIA32.EFI
```

Für 64 Bit CPU:

```
VXHOME/WindRiver2107/vxworks/21.07/workspace/uefi_x86_64/BOOTX64.EFI
```

Falls noch nicht kompiliert, erledigt dies folgendes Make:

```
cd VXHOME/WindRiver2107/vxworks/21.07/boot/uefi
make
```

3.4.2 VxWorks Boot Loader

Der VxWorks Boot Loader ist ein VxWorks-Betriebssystem mit einer speziellen Konfiguration.

- Im VSB-Projekt wird als Prozessor Mode UP statt SMP gewählt.
UP: Uniprocessor
SMP: Symmetric multiprocessing
- Im VIP-Projekt sorgt PROFILE_BOOTAPP für die passende Konfiguration.

Starten Sie das VSB-Projekt mit:

```
File > New > Wind River Workbench Project...
  Im Menüfenster wählen Sie den Projekttyp: VxWorks Source Build
Next >
  Projekt name: CoreBoot-VSB (z.B.)
Next >
  Based on: a board support package (BSW)
  BSW: itl-generic....
  CPU: CORE (z.B.)
  Processor mode: UP support
Finish >
```

Im letzten Schritt werden die Bibliotheken kompiliert:

```
rechte Maustaste auf Ihr VSB-Projekt > Build
```

Starten Sie das VIP-Projekt mit:

```
File > New > Wind River Workbench Project...
  Im Menüfenster wählen Sie den Projekttyp: VxWorks Image Project
Next >
  Project name: CoreBoot-VIP (z.B.)
Next >
  Based on: a Source Build Project (VSB)
```

```

    Projekt: CoreBoot-VSB (z.B.)
Next >
    Profile: PROFILE_BOOTAPP
Finish >

```

Im letzten Schritt wird das VxWorks Boot Image erstellt:

```
rechte Maustaste auf Ihr VIP-Projekt > Build
```

3.5 VxWorks Boot

VxWorks bietet unterschiedliche Boot-Möglichkeiten: von ROM, HD, USB-Stick, Netzwerk usw...

Die zur Zeit bevorzugte Variante:

- Booten des VxWorks Image von einem Server im Netzwerk.
- Boot-Loader und Boot-Parameter auf einem USB-Stick.
- Booten über UEFI.

Diese Boot-Konfiguration bietet folgende Eigenschaften:

- Das Anpassen an ein Experiment erfolgt durch Editieren der Boot-Parameterdatei auf dem USB-Stick.
- Software-Updates stehen für jeden Laborrechner ohne neue Installation zur Verfügung.
- Jeder PC kann auf seine Eignung als Laborrechner geprüft werden, ohne seine Festplatte zu verändern.

Der USB-Stick muss folgendermaßen vorbereitet werden:

Angenommen er ist als <USBstick> gemountet.

Kopieren der UEFI Boot Loader:

```

mkdir <USBstick>/EFI
mkdir <USBstick>/EFI/BOOT
cd VXHOME/WindRiver2107/vxworks/21.07/workspace/
cp uefi_ia32/BOOTIA32.EFI <USBstick>/EFI/BOOT
cp uefi_x86_64/BOOTX64.EFI <USBstick>/EFI/BOOT

```

Kopieren des VxWorks Boot Loaders (z.B. Core-Boot-VIP) auf den USB-Stick:

```

cd VXHOME/WindRiver2107/workspace/Core-Boot-VIP/default/
cp vxWorks <USBstick>/EFI/BOOT/bootapp.sys
umount <USBstick>

```

Editieren der Boot-Parameter-Datei EFI/BOOT/bootapp.cfg. Diese enthält in einer einzigen Zeile alle zum Booten notwendigen Informationen.

Boot-Zeile, aktuelle Testversion (eine Zeile!):

```

bootline:gei(0,0)bootHost:/usr/exp/vxw/INTEL/vxWorks-Core
          e=134.176.20.74:ffffff00 h=134.176.20.18 g=134.176.20.40
          u=huber tn=vmtest

```

```
s=/host.bootHost/home/huber/ex_home/ex_param/startup.vxwi
```

Boot-Zeile für das 017-Experiment nach Umzug in die Physik:

```
bootline:gei(0,0)bootHost:/usr/exp/vxw/INTEL/vxWorks-Core
e=134.176.21.35:ffffff00 h=134.176.20.18 g=134.176.21.1
u=exeis tn=exeis
s=/host.bootHost/home/exeis/ex_home/ex_param/startup.vxwi
```

Das Shell-Script startup.vxwi wird am Ende des Boot-Prozesses ausgeführt und kann editiert werden zur Anpassung an die Erfordernisse des Experimentes.

Die Boot-Parameter können beim Booten nochmals geändert werden durch Abbrechen des Boot-Vorgangs (s.h. Help). Die Änderungen werden allerdings nicht auf dem Stick gespeichert.

Boot-Parameter

boot device	gei (bzw. rtg für RealTek Ethernet)
processor number	0
host name	bootHost
file name	/usr/exp/vxw/INTEL/vxWorks-Core (z.B.)
inet on ethernet (e)	134.176.21.35:ffffff00
inet on backplane (b)	
host inet (h)	134.176.20.18
gateway inet (g)	134.176.21.1
user (u)	exeis
ftp password (pu)	
flags (f)	[0x0]
target name (tn)	exeis
startup script (s)	/host.bootHost/home/exeis/ex_home/ex_param/startup.vxwi
other (o)	

3.6 Anwendungen

Die Mess- und Auswerteprogramme sind aus historischen Gründen in C geschrieben. Die zunächst für VxWorks 5.3 entwickelten Programme konnten nach einigen Ergänzungen auch unter VxWorks 210x kompiliert werden (`#ifdef VXINTL` bzw. `VXMVME`). In wenigen Fällen war es angebracht Teile neu zu schreiben.

3.6.1 Programmquellen

Die Programmquellen sind zu finden unter

```
/usr/exp/ex_src/Progs/
Applib  Bibliothek für spezielle Auswertungen
Daclib  Bibliothek für Messprogramme
Danlib  Bibliothek für Auswerteprogramme
Daxlib  Bibliothek für Nicht-Experiment-Programme
Exlib   Bibliothek für gemeinsame Funktionen
esa22   Elektronenspektrometer
```

```

ex016    Labor 016
ex017    Labor 017 (EXEIS)
pipe     Pipe-Experiment
misc     allgemein genutzte Programme (MCA, PEAK usw...)

```

3.6.2 Kompilierung

Die Anwendungen können ebenfalls mit Workbench kompiliert werden. Ich habe jedoch die vorhandene Makefile-Struktur beibehalten, da nur eine geringfügige Anpassung notwendig war (s.h. Makefile.dac).

Die Anwendungen werden beim Kompilieren mit den Bibliotheken des VxWorks-Projektes (Board Support Package, CPU) verbunden (s.h. VSB), so dass eigentlich für alle Projekte mit unterschiedlicher CPU jeweils eine eigene Kompilation nötig ist. Es hat sich jedoch (mit Vorbehalt!) gezeigt, dass für das Intel-BSP die Wahl der CPU keine auffälligen Unterschiede macht. In den Makefiles (Makefile.dac, Makefile.dan) ist zur Zeit als Bezug Core-VSB angegeben.

Die ladbaren Programme sind unter /usr/exp/progs zu finden, und für jeden Experiment-Account auch unter ~/ex_home/ex_progs:

```

*.o VxWorks 5.3 Version
*.out VxWorks 2107 Version

```

Wird ein Programm neu installiert, so erhält der Vorgänger das Datum angehängt, an dem er ersetzt wurde.

3.6.3 Dokumentation

Die Mess- und Auswertprogramme sind mit dem Info-System (texinfo) von GNU dokumentiert. Mit Info können die gebräuchlichen Ausgabeformate erzeugt werden (HTML, PDF, DVI, Info, Text).

Die Quellen für texinfo finden Sie in dem Unterverzeichnis texi von jedem Mess- und Auswertprogramm. Die zu editierenden Quellen haben die Endung .txi und nachdem in einem ersten Schritt die Umlaute umgesetzt wurden, erhalten sie die Endung .texi.

Das Kompilieren der verschiedenen Ausgabeformate erfolgt mit

```
make info ....
```

Die Ausgaben erfolgen (z.B. für MCA) in das Verzeichnis

```
/usr/exp/ex_help/mca
```

3.6.4 Makefiles

Makefile-Hierarchie für das Messprogramm MCA

```

/usr/exp/ex_src/Progs/misc/mca
--> ./Makefile
--> ../../Makefile.dac (Cross Compiler Auswahl)
--> ./Makefile.mca (Cross Compiling)

```

Makefile-Hierarchie für das Auswertprogramm PEAK

```

/usr/exp/ex_src/Progs/misc/peak
--> ./Makefile
--> ../../Makefile.dan (Cross Compiler Auswahl)

```

```
--> ./Makefile.peak (Cross Compiling)
```

Makefile-Hierarchie für die MCA Info-Dokumentation

```
/usr/exp/ex_src/Progs/misc/mca
--> ./Makefile
--> ../../../../Makefile.info.top.V2
--> ../../../../Makefile.info.OSF1make.V2 (Servix)
--> ../../../../Makefile.info.GNUMake.V2 (Ionix)
```

Makefile-Hierarchie für HTML-Index und -FAQs

```
/usr/exp/ex_src/Progs
--> ./Makefile
--> ./Makefile.index
--> ./Makefile.FAQs
```

3.7 VxWorks 21.07 Dokumentation und Support

Für VxWorks gibt es unglaublich viel Dokumentation, so dass es nicht einfach ist das Passende zu finden. Der Zugriff zur Doku erfordert ein Login:

```
User: schippers@jlug.de
Passw: Stefan fragen
```

Eine Einstiegsmöglichkeit zur VxWorks Dokumentation ist folgende:

```
https://docs.windriver.com
```

oder nach 'WindRiver Login' suchen. Weiter mit:

```
> Support (rechts oben)

- Lesen der Dokumentation:
> Documentation
> VxWorks
> VxWorks 21.07
> Configuration & Build (z.B.)

- oder Kontakt mit Wind River aufnehmen (bisher mit Note 2+):
> Submit a Case
Für ein 'Support Case' ist eine persönliche Anmeldung
bei WindRiver notwendig.

- oder das Forum befragen:
> Knowledge Forum

- und Einiges mehr...
```

Weitere Startmöglichkeiten:

```
https://docs.windriver.com/category/os\_vxworks\_21\_07
https://knowledge.windriver.com/en-us/Wind\_River\_Support\_Network
```

3.8 Gut zu wissen...

3.8.1 Login

Die aktuelle Konfiguration des User Managements:

- Für ein Login ist ein User-Name und ein Passwort nötig. Es sind mehrere Logins gleichzeitig möglich:
 - local, seriell, 9,6 kbaud: VT200, TeraTerm
 - remote, viel schneller: Telnet
- Die Passwort-Datei befindet sich auf der Ram-Disk0 und geht deshalb mit jedem Runterfahren des PC verloren, da geplant ist, User und Passwort über die Experiment-eigene 'startup.vxwi' Datei zu setzen (add user). Alternativ könnte die Passwort-Datei auch permanent auf dem USB-Stick eingerichtet werden.
- Wenn in der 'startup.vxwi' Datei keine User vergeben werden, dann wird nach dem Booten ein erster User mit freiem Passwort angefordert, der Admin-Rechte hat und mit 'add user' weitere User eintragen kann.
- Wenn in der 'startup.vxwi' Datei bereits User definiert sind dann kann nach dem Booten ein Login nur mit einem dieser User erfolgen.
- Dadurch dass mehrere Logins gleichzeitig erlaubt sind, kann das System während einer laufenden Messung überwacht werden (z.B. Last) und es besteht die Möglichkeit, bei einer evtl. abgebrochene Verbindung zum laufenden Experiment die Kontrolle wieder zu erlangen.

Es besteht jedoch auch die Gefahr, eine laufende Messung durch einen zweiten Start zum Scheitern zu bringen. Um dies zu verhindern gibt das 'Start'-Menue eine Warnung aus, wenn es erkennt, dass es bereits aktiv ist. Wenn jedoch durch eine abgebrochene Verbindung die Kontrolle über die laufende Messung verloren gegangen ist, so kann sie auf diesem Wege möglicherweise wieder erlangt werden.

Man kann diesen Weg aber auch dazu benutzen, eine bestehende Verbindung raus zu werfen und die Kontrolle zu übernehmen. Das kann aber je nach Status der Messung auch schief gehen, Vorsicht!!!

3.8.2 Kernel Shell

Die Kernel Shell hat einen C- und einen CMD-Interpreter. Nach dem Login gelangt man zunächst in den C-Interpreter, von dem aus man in den mächtigeren CMD-Interpreter wechseln kann und auch wieder zurück. Und man kann Kommandos des jeweils anderen Interpreters ausführen ohne zu wechseln:

- C-Interpreter
 - > help Liste der C-Interpreter Kommandos
 - > spyHelp z.B.
 - > CMD Wechsel zum CMD-Interpreter
 - > CMD ... Kommando im CMD-Interpreter ausführen ohne Wechsel
- CMD-Interpreter
 - [...]# help Liste der CMD-Interpreter Kommandos
 - [...]# help spy z.B.


```
[... ]# C          Wechsel zum C-Interpreter
[... ]# C ...      Kommando im C-Interpreter ausführen ohne Wechsel
```

Editieren der Kommandozeile:

Es stehen zwei Command-line Editoren zur Verfügung, vi und emacs. Aktuell ist emacs-like voreingestellt, wegen der größeren Ähnlichkeit mit Linux.

- Editoren:
 - + vi-like Editor
 - Mit ESC erfolgt der Wechsel in die Navigation.
 - Mit a, A, i, I erfolgt der Wechsel in den Insert-Mode.
 - (siehe Anhang)
 - Zeichen löschen (rückwärts) mit CTRL+h
 - + emacs-like Editor
 - Die Cursor-Tasten können wie gewohnt benutzt werden.
 - Ungewohnt ist, dass Backspace, DEL und CTRL+d vorwärts löschen!
 - Nur CTRL+h löscht rückwärts.
- Wechsel der Editoren:
 - C-Interpreter:
 - > CMD vi Wechsel zum vi-Editor
 - > CMD emacs Wechsel zum emacs-Editor
 - CMD-Interpreter:
 - [...]# vi Wechsel zum vi-Editor
 - [...]# emacs Wechsel zum emacs-Editor

Command History:

- vi Editor
 - h History
 - ESC-nG geh zur History-Zeile n
 - ESC-K zurück in History
 - ESC-J vor in History
- emacs Editor
 - h History
 - !n führe History-Zeile n aus
 - Cursor up zurück in History
 - Cursor down vor in History

Die CTRL-Funktionen der Kernel Shell:

- CTRL+C Aborts and restarts the shell.
However, if a process is launched with the command interpreter (using rtp exec), the function of CTRL+C changes. It is used to interrupt the process.
- CTRL+D Action depends on the location of the cursor.
At the beginning of a line (that is, next to the shell prompt):
Exits the shell or logs out of the session (if login is enabled). Logs out with remote

sessions (telnet, for example) when INCLUDE_SECURITY is included; or with the console shell when INCLUDE_SECURITY is included and SHELL_SECURE is set to TRUE.

At the end of a fragment of a command, file name, or path name:

Lists all possible matches when the fragment is ambiguous, or completes the fragment when it is not. The TAB key also completes a fragment when it is not ambiguous. Note that file name completion only works with file systems that differentiate between files and directories (netDrv is does not).

- CTRL+H Deletes a character (backspace).
- CTRL+Q Resumes output.
- CTRL+S Temporarily suspends output.
- CTRL+U Deletes an entire line.
- CTRL+W If a process is launched with the command interpreter (using rtp exec), this key sequence suspends the process running in the foreground.
- CTRL+X Reboots (trap to the ROM monitor).
- ESC Toggles between input mode and edit mode (vi mode only).

siehe auch:

Kernel Shell User's Guide

https://docs.windriver.com/bundle/Kernel_Shell_Users_Guide_1/page/1643959.html

Command-line Editing

<https://www.vxworks6.com/command-line-editing-and-object-module-loader.html>

3.9 Bekannte Probleme

3.9.1 Start-Menü

Der Transfer der Kontrolle von einem Login zum anderen funktioniert wie geplant, solange ein Anwenderprogramm (MCA, PEAK) aktiv ist. Wird dieses verlassen und danach der Bildschirm (Login) gewechselt, dann erfolgt bei einem neuen Start eines Anwenderprogramms die Ausgabe auf den alten Bildschirm. Mit einem neuen Start-Kommando auf dem alten Bildschirm ist dies zu reparieren.

3.9.2 Spy-App

Spy zeigt die Aktivität von CPUs und Tasks an. In der aktuellen Core-VIP Konfiguration lässt sich Spy starten, auf dem Alder Lake (12. Generation) erfolgt aber keine Ausgabe. Einigen Hinweisen zufolge steht möglicherweise die benötigte auxClock nicht zur Verfügung.

Lösung: Wechsel der auxClock

```
Auxiliary clock source select
```

```
alt: I8253
```

```
neu: MC146818
```

Funktioniert auch auf älteren Generationen, eventuelle Nebenwirkungen noch unbekannt.

3.10 Anhang

3.10.1 Parameter für PC-Console

Das vxWorks-Core-Projekt ist für die Bedienung über einen seriellen Anschluss und über Telnet konfiguriert und kann damit mittels 'Tera Term' (Windows) und 'xTerm' (linux) gesteuert werden. Die lokalen Komponenten des VxWorks-Laborrechners (Monitor, Tastatur und Maus) werden daher nicht benötigt.

Alternativ ist für einige PC-Modelle eine lokale Bedienung mittels USB-Tastatur und Monitor möglich (PC-Console-Betrieb), der jedoch nicht für die Mess- und Auswerteprogramme geeignet ist, da er keine VT240-Kontrollsequenzen und -Grafik versteht.

Für den PC-Console-Betrieb sind folgende Parametereinstellungen notwendig, man findet sie mit dem Stichwort 'Console':

- ser. Console (exclude)
- USB GEN2 Serial Device Attach to PC Console Init (include)
- PC console (include)
- PC console keyboard (include)
- PC console video (VGA oder UEFI, probieren...)

3.10.2 Telnet mit Tera Term

Mittels Tera Term kann der VxWorks-Laborrechner sowohl über den seriellen Port (COM) als auch über Telnet gesteuert werden. Vorzugsweise wird Telnet verwendet, da damit die Grafikausgaben deutlich schneller dargestellt werden als über die serielle Leitung.

Damit die Telnet-Verbindung nicht wegen Inaktivität abgebrochen wird, sind folgende Einstellungen an Windows und Telnet vorzunehmen:

Windows ist üblicherweise so eingestellt, dass es nach einer gewissen Zeit der Inaktivität in den Energiesparmodus wechselt.

Die Telnet-Verbindung geht dadurch verloren.

Notwendige Änderung:

Einstellungen

-> Netzbetrieb und Energiesparmodus

-> Energiesparmodus: nie

Tera Term hat einen 'Keep alive' Parameter, der dafür sorgt, dass periodisch eine minimale Übertragung zum Telnet-Server erfolgt, um eine Inaktivität zu vermeiden.

Ein 'Keep alive' Takt von 1s hat funktioniert:

Setup -> TCP/IP -> Keep alive: 1

Weitere nützliche Tera Term Setups:

Terminal

-> ID: VT320

-> Autoswitch VT <-> TEK: yes

-> Termsize = WINSIZE: yes

Das Grafikprogramm (Peak) muss für Tera Term konfiguriert sein:

```
Analyse Spectrum
-> ^Z
-> Set configuration
-> Terminals and printers
-> Select terminal
-> TeraTerm (TEK4010)
```

3.10.3 Telnet mit xTerm

Alternativ zu Tera Term unter Windows bietet xTerm unter Linux die selben Funktionen, jedoch kein 'Keep alive'. Da der Telnet-Server unter VxWorks so konfiguriert ist, dass er Inaktivität ignoriert, sollte dies kein Problem sein. Entsprechende Tests stehen noch aus...

Das Grafikprogramm (Peak) muss für xTerm konfiguriert sein:

```
Analyse Spectrum
-> ^Z
-> Set configuration
-> Terminals and printers
-> Select terminal
-> xTerm (TEK4010)
```

3.10.4 Navigating in VI

To start working with the VI editor, you need to press ESC to enter line-editing mode. Here is a list of those commands.

```
k - Backwards through history buffer
j - Forwards through history buffer
h - Move cursor to the left
l - Move cursor to the right
a - Append after cursor
A - Append at end of line
i - Insert before cursor
I - Insert at beginning of line
rc - Replace current character with character c
x - Delete current character
dd - Delete entire line
nG - Go to history line /string Search for string
```

3.10.5 Intel CPU Generationen

Die CPU-Generation g bzw. gg ist der ix-ggxxx Bezeichnung zu entnehmen bei Abfrage der Prozessor Info (z.B. g = 7):

Intel Core i7-7700K CPU @ 4.20GHz x 8

3 Mikroarchitekturen

3.1 Vorgänger: Pentium M, Core- und Core-2-Mikroarchitektur

3.1.1 Core-Mikroarchitektur

3.1.2 Core-2-Mikroarchitektur

- 3.2 Nehalem-Mikroarchitektur
 - 3.2.1 Nehalem (1. Generation)
 - 3.2.2 Westmere (1. Generation)
- 3.3 Sandy-Bridge-Mikroarchitektur
 - 3.3.1 Sandy-Bridge (2. Generation)
 - 3.3.2 Ivy-Bridge (3. Generation)
- 3.4 Haswell-Mikroarchitektur
 - 3.4.1 Haswell (4. Generation)
 - 3.4.2 Broadwell (5. Generation)
- 3.5 Skylake-Mikroarchitektur
 - 3.5.1 Skylake (6. Generation)
 - 3.5.2 Kaby-Lake (7. Generation)
 - 3.5.3 Coffee Lake (8. Generation)
 - 3.5.4 Coffee Lake Refresh (9. Generation)
 - 3.5.5 Cannon Lake (8. Generation)
 - 3.5.6 Whiskey Lake (8. Generation)
 - 3.5.7 Comet Lake (10. Generation)
 - 3.5.8 Cascade Lake (10. Generation)
 - 3.5.9 Rocket Lake (11. Generation)
- 3.6 Ice-Lake-Mikroarchitektur
 - 3.6.1 Ice Lake (10. Generation)
 - 3.6.2 Lakefield (10. Generation, hybrid: 1 Core i-Kern und 4 Atom-Kerne)
 - 3.6.3 Tiger Lake (11. Generation)
 - 3.6.4 Alder Lake (12. Generation)
 - 3.6.5 Raptor Lake (13. Generation)
 - 3.6.6 Raptor Lake Refresh (14. Generation)

3.10.6 VIP Includes und Parameter für das Core-Projekt

In der VIP-Profile Datei für das Core-Projekt

```
VXHOME/WindRiver2107/vxworks/21.07/os/vip_profiles/profiles/
vip_profile_PROFILE_CORE-VIP.cdf
```

ist dokumentiert, mit welchen Includes und Parametern die BSP-Einstellungen ergänzt wurden um die benötigten Funktionen zu konfigurieren:

```
Profile PROFILE_CORE-VIP
```

```
PROFILES BSP_DEFAULT
```

```
COMPONENTS INCLUDE_IPTELNETS\
INCLUDE_IPARP_CMD\
INCLUDE_IPCOM_SYSLOGD_CMD\
INCLUDE_IPCOM_SYSLOGD_USE_LOG_FILE\
INCLUDE_IPCOM_SYSVAR_CMD\
INCLUDE_IPIFCONFIG_CMD\
INCLUDE_IPD_CMD\
INCLUDE_IPNETSTAT_CMD\
```

```
INCLUDE_IPPING_CMD\  
INCLUDE_IPROUTE_CMD\  
INCLUDE_IPSNTP_CMD\  
INCLUDE_IPNTP_CMD\  
INCLUDE_IPNTP\  
INCLUDE_IPSYSCTL_CMD\  
INCLUDE_IPCOM_USE_TIME_CMD\  
INCLUDE_NETPOOLSHOW\  
INCLUDE_NET_HOST_SHOW\  
INCLUDE_NETSTAT\  
INCLUDE_PING\  
INCLUDE_RLOGIN\  
INCLUDE_TELNET_CLIENT\  
INCLUDE_NFS2_CLIENT\  
INCLUDE_NFS3_CLIENT\  
INCLUDE_ANSI_ASSERT\  
INCLUDE_ANSI_COMPLEX\  
INCLUDE_ANSI_CTYPE\  
INCLUDE_ANSI_ERRNO\  
INCLUDE_ANSI_FENV\  
INCLUDE_ANSI_FLOAT\  
INCLUDE_ANSI_INTTYPES\  
INCLUDE_ANSI_ISO646\  
INCLUDE_ANSI_LIMITS\  
INCLUDE_ANSI_LOCALE\  
INCLUDE_ANSI_MATH\  
INCLUDE_ANSI_SETJMP\  
INCLUDE_ANSI_SIGNAL\  
INCLUDE_ANSI_STDALIGN\  
INCLUDE_ANSI_STDARG\  
INCLUDE_ANSI_STDATOMIC\  
INCLUDE_ANSI_STDBOOL\  
INCLUDE_ANSI_STDDEF\  
INCLUDE_ANSI_STDINT\  
INCLUDE_ANSI_STDIO\  
INCLUDE_ANSI_STDLIB\  
INCLUDE_ANSI_STDNORETURN\  
INCLUDE_ANSI_STRING\  
INCLUDE_ANSI_TGMATH\  
INCLUDE_ANSI_THREADS\  
INCLUDE_ANSI_TIME\  
INCLUDE_ANSI_UCHAR\  
INCLUDE_ANSI_WCHAR\  
INCLUDE_ANSI_WCTYPE\  
INCLUDE_C_STDLIB\  
INCLUDE_SHOW_ROUTINES\  
INCLUDE_DEFAULT_TIMEZONE\  

```

```
INCLUDE_POSIX_AIO_SHOW\  
INCLUDE_POSIX_MQ_SHOW\  
INCLUDE_POSIX_SEM_SHOW\  
INCLUDE_TRIGGER_SHOW\  
INCLUDE_RBUFF_SHOW\  
INCLUDE_DATACollector_SHOW\  
INCLUDE_EXC_SHOW\  
INCLUDE_ISR_SHOW\  
INCLUDE_POOL_SHOW\  
INCLUDE_SYSCTL_CLI\  
INCLUDE_ACPI_SHOW\  
INCLUDE_RTP_SHELL_CMD\  
INCLUDE_RTP_SHOW_SHELL_CMD\  
INCLUDE_SHL_SHELL_CMD\  
INCLUDE_DEV_NULL\  
INCLUDE_SHELL_EMACS_MODE\  
INCLUDE_SHELL_HISTORY_FILE\  
INCLUDE_STAT_SYM_TBL\  
INCLUDE_RAM_DISK\  
INCLUDE_RAM_DISK_FORMAT_HRFS\  
INCLUDE_FSINFO_SHOW\  
INCLUDE_NFS_CLIENT_ALL\  
INCLUDE_NFS_CLIENT_RPC_TCP\  
INCLUDE_RAMDRV\  
INCLUDE_STARTUP_SCRIPT\  
INCLUDE_IPNSLOOKUP_CMD\  
INCLUDE_MEM_PART_CACHE_SHOW\  
INCLUDE_DISK_UTIL_SHELL_CMD\  
INCLUDE_EDR_SHELL_CMD\  
INCLUDE_TASK_SHELL_CMD\  
INCLUDE_SYM_SHELL_CMD\  
INCLUDE_VM_SHOW_SHELL_CMD\  
INCLUDE_ADR_SPACE_SHELL_CMD\  
INCLUDE_SHARED_DATA_SHOW_SHELL_CMD\  
INCLUDE_MAPPED_FILES_SHOW_SHELL_CMD\  
INCLUDE_MEM_EDR_SHELL_CMD\  
INCLUDE_MEM_EDR_RTP_SHELL_CMD\  
INCLUDE_MODULE_SHELL_CMD\  
INCLUDE_UNLOADER_SHELL_CMD\  
INCLUDE_HISTORY_FILE_SHELL_CMD\  
INCLUDE_SPY_SHELL_CMD\  
INCLUDE_TIP_CMD\  
INCLUDE_USER_MGT_SHELL_CMD\  
INCLUDE_TIMER_SYS_SHOW\  
INCLUDE_USB_GEN2_STORAGE_INIT\  
INCLUDE_EHCI_INIT\  
INCLUDE_OHCI_INIT
```

```

INCLUDE_UHCI_INIT\
INCLUDE_USB_XHCI_HCD_INIT\
INCLUDE_USB_SHOW\
INCLUDE_MC146818_AUX_CLK\
INCLUDE_INTEL_CPU_SHOW\
INCLUDE_EHCI\
INCLUDE_DEV_DRV\
INCLUDE_DEV_ZERO\
INCLUDE_DEV_RANDOM\
INCLUDE_DEV_URANDOM\
INCLUDE_DEV_ARANDOM\
INCLUDE_DEV_TTY\
INCLUDE_DEV_CONSOLE\
INCLUDE_DEV_FULL\
INCLUDE_TMP_DIR\
INCLUDE_USB_GEN2_STORAGE\
INCLUDE_USB_GEN2_MOUSE_INIT\
INCLUDE_USB_GEN2_SER_FTDI232\
INCLUDE_USB_GEN2_SER_PL2303\
INCLUDE_OHCI\
INCLUDE_UHCI\
INCLUDE_USB_XHCI_HCD\
INCLUDE_MPTBL_SHOW\
INCLUDE_MP_APIC_DATA_SHOW

```

BUNDLES

```

PARAMETER_VALUES IPCOM_TELNET_IDLE_TIMEOUT "0"
TIMEZONE_ENV_VAR "TZ=CET-1CEST,M3.5.0/02,M10.5.0/03"
VX_SMP_NUM_CPUS VX_MAX_SMP_CPUS
SHELL_COMPATIBLE FALSE
SHELL_DEFAULT_CONFIG ("INTERPRETER=,LINE_EDIT_MODE=emacs,LINE_LENGTH=1024,STRING_FREE
DNSC_DOMAIN_NAME "IAMP"
DNSC_PRIMARY_NAME_SERVER "134.176.2.7"
DNSC_SECONDARY_NAME_SERVER "134.176.2.5"
NTP_INIT_PARAM NULL
NTP_CONFIG_PARAM NULL
NTP_CONFIG_FILENAME NULL
SYSFLG_NO_SECURITY_ENABLE FALSE
UDB_HASH_KEY .....
UDB_STORAGE_PATH "/ram0/cxUserDB.txt"
UDB_PROMPT_INITIAL_USER TRUE
SYS_CLK_RATE (100)
USB_GEN2_SERIAL_IOS_RING_BUFFER_SIZE 131072

```

PROFILES_EXCLUSIONS

4 Neues Experiment

Am Beispiel des Experimentes EXEIS wird im Folgenden gezeigt welche System- und Account-Einstellungen notwendig sind für den Anschluss eines Experiments.

4.1 System-Einstellungen

siehe auch: /root/firewall.sh, /root/Umzug017

*** Firewall anpassen ***

* Allow rlogin and rsh for EXEIS only

```
sudo ufw allow proto tcp from 134.176.21.35 to any port 513 comment 'rlogin'
sudo ufw allow proto tcp from 134.176.21.35 to any port 514 comment 'rsh'
```

* Allow NFS for EXEIS only

```
sudo ufw allow from 134.176.21.35 to any port 111 comment 'Portmapper'
sudo ufw allow from 134.176.21.35 to any port nfs comment 'NFS'
# Der folgende mountd-Port wurde auf 33333 festgelegt (s.u. NFS)
sudo ufw allow from 134.176.21.35 to any port 33333 comment 'mountd'
```

* Show Firewall settings

```
sudo ufw status verbose
sudo ufw reload
```

* Hilfen

```
sudo ufw delete nn      #delete rule nn (1...)
sudo cat /var/log/ufw.log -> ...[UFW BLOCK]... SRC=... DST=... DPT=<Port>
```

* Problem

Obwohl der Port 514 frei gegeben ist, taucht beim Booten von VxWorks eine unterschiedliche Zahl (3-10?) von Block-Meldungen im Log-File auf. Offensichtlich ohne weitere Auswirkungen:

```
Oct  8 11:50:54 ionix kernel: [1191732.334599] [UFW BLOCK]
IN=enp0s31f6 OUT=
MAC=54:bf:64:6e:c7:d0:34:17:eb:c8:17:04:08:00
SRC=134.176.21.35
DST=134.176.21.37
LEN=40 TOS=0x00 PREC=0x00 TTL=64 ID=61970
PROTO=TCP SPT=1009 DPT=514
WINDOW=0 RES=0x00 RST URGP=0
```

*** /etc/hosts ergänzen ***

```
134.176.21.35   exeis.strz.uni-giessen.de           exeis
```

*** NFS ***

* NFS Einstellungen

Der normalerweise dynamische mountd-Port muss für die Firewall festgelegt werden. Dazu

muss NFS V4 abgeschaltet werden, sonst gibt es Fehler beim Restart.
 /etc/default/nfs-kernel-server ergänzen:

```
RPCMOUNTDOPTS="--manage-gids"
RPCMOUNTDOPTS="--no-nfs-version 4"
RPCMOUNTDOPTS="--port 33333"
```

* /etc/exports (NFS) ergänzen

```
/home/exeis 134.176.21.35(rw,root_squash,sync,no_subtree_check)
```

* Restart und Check NFS

```
sudo systemctl restart nfs-server
sudo systemctl is-enabled nfs-server
sudo systemctl status nfs-server
```

4.2 Account-Konfiguration

Für das Experiment EXEIS z.B. sind folgende Verzeichnisse anzulegen:

```
Messdaten:      /home/exeis/ex_home/ex_data
Dokumentation:  /home/exeis/ex_home/ex_help -> /usr/exp/ex_help
Programmparameter: /home/exeis/ex_home/ex_param
Programme:      /home/exeis/ex_home/ex_prog -> /usr/exp/ex_prog
Quellen:        /home/exeis/ex_home/ex_src -> /usr/exp/ex_src
VxWorks-Systeme: /home/exeis/ex_home/vxw -> /usr/exp/vxw
```

Um rsh und rlogin zu erlauben, wird für EXEIS in .rhosts folgender Eintrag benötigt:

```
/home/exeis/.rhosts
134.176.21.35
exeis.strz.uni-giessen.de
```

VxWorks lädt nach dem Booten zum Konfigurieren die Datei

```
/home/exeis/ex_home/ex_param/startup.vxw   für MVME-Rechner
bzw.
/home/exeis/ex_home/ex_param/startup.vxwi   für INTEL-Rechner
```

Nach dem Kopieren einer solchen, bereits vorhandenen Datei sind einige Änderungen zur Anpassung notwendig.