

Сборка ядра Thinstation

Хочу предупредить взыскательных читателей. Данные записки не претендуют на звание мануала. Это записки больше для себя, чтобы через год не вспоминать, напрягая до судорог синие, атрофированные мозги.

Итак. Постановка задачи. Имеем гигабитную сетевую карточку, не поддерживаемую в Thinstation 2.2, а именно Attansic L1, необходимо включить поддержку этой карточки. К счастью, производитель чипа данной карточки предоставляет с ней вполне рабочие дрова для Линукс. Причем драйвер может компилироваться как с ядром 2.4, так и с 2.6.

Читаем соответствующую [хауту](#)

И приступаем к работе

1) [Качаем исходники Thinstation](#)

2) Создаем каталог compile и build. В первый распаковываем закаченные исходники, во второй готовый "сборочный" пакет thinstation.

В моем случае это были /home/slava/compile и /home/slava/build

3) Готовим исходники к работе.

```
cd /home/slava/compile/thinstation_src-2.2
```

```
./RUNME
```

После этого с ужасом обнаруживаете, что находитесь в другом боксе (chroot), но не теряйтесь, а смело выполняйте

```
cd source/kernel-2.6.16.5/linux-2.6.16.5
```

```
cp .config ../.config.orig
```

```
make mrproper
```

```
make distclean
```

(запускаем menuconfig, ничего не меняем, просто сохраняем конфигурацию)

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
make menuconfig
```

```
cp ../config.orig .config
```

```
exit
```

Очутившись в своей родной оболочке, перемещаемся в каталог

```
cd /home/slava/build/Thinstation-2.2/utils
```

и, вооружившись любимым текстовым редактором, меняем первую и единственную строчку файла SOURCE_PATH на

```
/home/slava/compile/thinstation_src-2.2/source
```

Далее...

```
cd /home/slava/compile/thinstation_src-2.2
```

```
./RUNME
```

```
cd source/kernel-2.6.16.5/linux-2.6.16.5
```

(Опять конфигурируем ядро, может чего забыли добавить. А можно просто сохранить все, как есть)

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
make menuconfig
```

```
cd ..
```

(компилируем ядро и модули)

```
./setup.sh
```

(инсталлим модули и выходим)

```
cd linux-2.6.16.5
```

```
make modules_install
```

```
exit
```

Далее, необходимо установить некоторые пакеты. Признаюсь, не в курсе все ли они действительно необходимы, просто многие инструкции тупо скопированы мной из соответствующего хауту.

```
su
```

Password:

```
apt-get install sharutils
```

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
apt-get install libiw-dev
```

```
apt-get install pcmcia-cs
```

Далее, прячем системные компиляторы и подставляем соответствующие библиотеки

```
mv /usr/bin/gcc /usr/bin/gcc.orig
```

```
mv /usr/bin/g++ /usr/bin/g++.orig
```

```
mv /usr/bin/cc /usr/bin/cc.orig
```

```
mkdir -p /usr/local/lib/gcc/i686-pc-linux-gnu/3.4.4
```

```
cp -r /home/slava/compile/thinstation_src-2.2/usr/local/lib/gcc/i686-pc-linux-gnu/3.4.4/* /usr/local/lib/gcc/i686-pc-linux-gnu/3.4.4/
```

```
PATHORIG=$PATH
```

```
PATH=$PATH:/home/slava/compile/thinstation_src-2.2/usr/bin/
```

```
exit
```

Теперь можно приступать к компиляции драйверов.

Качаем дрова [отсюда](#) .

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

Распаковываем в любой каталог и меняем содержимое файла Makefile. Должно получится примерно следующее

```
CFILES = at_main.c at_hw.c at_param.c
at_test.c kcompat.c at_ethtool.c
HFILES = at.h at_hw.h at_osdep.h kcompat.h drv_ioctl.h
ifeq (,$(BUILD_KERNEL))
BUILD_KERNEL=$(shell uname -r)
endif

#####
# Environment tests

# Kernel Search Path
# All the places we look for kernel source
KSP := /lib/modules/$(BUILD_KERNEL)/source
/lib/modules/$(BUILD_KERNEL)/build
/usr/src/linux-$(BUILD_KERNEL)
/usr/src/linux-$(BUILD_KERNEL) | sed 's/-.*/'
/usr/src/kernel-headers-$(BUILD_KERNEL)
/usr/src/kernel-source-$(BUILD_KERNEL)
/usr/src/linux-$(BUILD_KERNEL) | sed 's/([0-9]*[0-9]*).**/1/'
/usr/src/linux

# prune the list down to only values that exist
# and have an include/linux sub-directory
test_dir = $(shell [ -e $(dir)/include/linux ] && echo $(dir))
KSP := $(foreach dir, $(KSP), $(test_dir))

# we will use this first valid entry in the search path
ifeq (,$(KSRC))
KSRC := $(firstword $(KSP))
endif

ifeq (,$(KSRC))
$(error Linux kernel source not found)
else
ifeq (/lib/modules/$(shell uname -r)/source, $(KSRC))
KOBJ := /lib/modules/$(shell uname -r)/build
else
KOBJ := $(KSRC)
endif
endif
```

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
# check for version.h and autoconf.h for running kernel in /boot (SUSE)
ifneq (,$(wildcard /boot/vmlinuz.version.h))
VERSION_FILE := /boot/vmlinuz.version.h
CONFIG_FILE := /boot/vmlinuz.autoconf.h
KVER := $(shell $(CC) $(CFLAGS) -E -dM $(VERSION_FILE) |
grep UTS_RELEASE | awk '{ print $$$3 }' | sed 's/"//g')
ifeq ($(KVER),$(shell uname -r))
# set up include path to override headers from kernel source
x=$(shell rm -rf include)
x=$(shell mkdir -p include/linux)
x=$(shell cp /boot/vmlinuz.version.h include/linux/version.h)
x=$(shell cp /boot/vmlinuz.autoconf.h include/linux/autoconf.h)
CFLAGS += -I./include
else
VERSION_FILE := $(KOBJ)/include/linux/version.h
CONFIG_FILE := $(KSRC)/include/linux/config.h
endif
else
VERSION_FILE := $(KOBJ)/include/linux/version.h
CONFIG_FILE := $(KSRC)/include/linux/config.h
endif

ifeq (,$(wildcard $(VERSION_FILE)))
$(error Linux kernel source not configured - missing version.h)
endif

ifeq (,$(wildcard $(CONFIG_FILE)))
$(error Linux kernel source not configured - missing config.h)
endif

# pick a compiler
ifneq (,$(findstring egcs-2.91.66, $(shell cat /proc/version)))
CC := kgcc gcc cc
else
CC := gcc cc
endif
test_cc = $(shell $(cc) --version > /dev/null 2>&1 && echo $(cc))
CC := $(foreach cc, $(CC), $(test_cc))
CC := $(firstword $(CC))
ifeq (,$(CC))
$(error Compiler not found)
endif

# we need to know what platform the driver is being built on
# some additional features are only built on Intel platforms
ARCH := $(shell uname -m | sed 's/i.86/i386/')
```

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
ifeq ($(ARCH),alpha)
CFLAGS += -ffixed-8 -mno-fp-regs
endif
ifeq ($(ARCH),x86_64)
CFLAGS += -mcmmodel=kernel -mno-red-zone
endif
ifeq ($(ARCH),ppc)
CFLAGS += -msoft-float
endif
ifeq ($(ARCH),ppc64)
CFLAGS += -m64 -msoft-float
LDFLAGS += -melf64ppc
endif

# standard flags for module builds
#CFLAGS += -DLINUX -D__KERNEL__ -DMODULE -DDBG -O2 -pipe -Wall
CFLAGS += -DLINUX -D__KERNEL__ -DMODULE -O2 -pipe -Wall
CFLAGS += -I$(KSRC)/include -l.
CFLAGS += $(shell [ -f $(KSRC)/include/linux/modversions.h ] &&
echo "-DMODVERSIONS -DEXPORT_SYMTAB
-include $(KSRC)/include/linux/modversions.h")

CFLAGS += $(CFLAGS_EXTRA)

RHC := $(KSRC)/include/linux/rhconfig.h
ifneq (,$(wildcard $(RHC)))
# 7.3 typo in rhconfig.h
ifneq (,$(shell $(CC) $(CFLAGS) -E -dM $(RHC) | grep __module__bigmem))
CFLAGS += -D__module__bigmem
endif
endif

# get the kernel version - we use this to find the correct install path
#KVER := $(shell $(CC) $(CFLAGS) -E -dM $(VERSION_FILE) | grep UTS_RELEASE |
# awk '{ print $$$3 }' | sed 's//g')
# Added by Brackett, for install driver on Red flag dist.

#KVER := $(shell uname -r | awk '{ print $$1 }')
KVER := 2.6.16.5

# KKVER := $(shell echo $(KVER) |
# awk '{ if ($$0 ~ /2.[4-9]./) print "1"; else print "0"}')

ifeq ($(KKVER), 0)
```

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
$(error *** Aborting the build.
*** This driver is not supported on kernel versions older than 2.4.0)
endif

# set the install path
#INSTDIR := /lib/modules/$(KVER)/kernel/drivers/net/atl1
INSTDIR := /home/slava/compile/thinstation_src-2.2/lib/modules/$(KVER)/kernel/drivers/net/atl1

# look for SMP in config.h
SMP := $(shell $(CC) $(CFLAGS) -E -dM $(CONFIG_FILE) |
grep CONFIG_SMP | awk '{ print $$3 }')
ifneq ($(SMP),1)
SMP := 0
endif

ifneq ($(SMP),$(shell uname -a | grep SMP > /dev/null 2>&1 && echo 1 || echo 0))
$(warning ***
ifeq ($(SMP),1)
$(warning *** Warning: kernel source configuration (SMP))
$(warning *** does not match running kernel (UP))
else
$(warning *** Warning: kernel source configuration (UP))
$(warning *** does not match running kernel (SMP))
endif
$(warning *** Continuing with build,)
$(warning *** resulting driver may not be what you want)
$(warning ***
endif

ifeq ($(SMP),1)
CFLAGS += -D__SMP__
endif

#####
# 2.4.x & 2.6.x Specific rules

#K_VERSION:=$(shell uname -r | cut -c1-3 | sed 's/2.[56]/2.6/')
K_VERSION:=2.6

ifeq ($(K_VERSION), 2.6)

# Makefile for 2.6.x kernel
TARGET = atl1.ko

# man page
MANSECTION = 7
```


Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
MANFILE = $(TARGET:.ko=.$(MANSECTION))

ifneq ($(PATCHLEVEL),)
EXTRA_CFLAGS += $(CFLAGS_EXTRA)
obj-m += atl1.o
atl1-objs := $(CFILES:.c=.o)
else
default:
ifeq ($(KOBJ),$(KSRC))
make -C $(KSRC) SUBDIRS=$(shell pwd) modules
else
make -C $(KSRC) O=$(KOBJ) SUBDIRS=$(shell pwd) modules
endif
endif

else # ifeq ($(K_VERSION),2.6)

# Makefile for 2.4.x kernel
TARGET = atl1.o

# man page
MANSECTION = 7
MANFILE = $(TARGET:.o=.$(MANSECTION))

# Get rid of compile warnings in kernel header files from SuSE
ifneq (,$(wildcard /etc/SuSE-release))
CFLAGS += -Wno-sign-compare -fno-strict-aliasing
endif

# Get rid of compile warnings in kernel header files from fedora
ifneq (,$(wildcard /etc/fedora-release))
CFLAGS += -fno-strict-aliasing
endif

.SILENT: $(TARGET)
$(TARGET): $(filter-out $(TARGET), $(CFILES:.c=.o))
$(LD) $(LDFLAGS) -r $^ -o $@
echo; echo
echo "*****"
echo "*** $(TARGET) built for $(KVER)"
echo -n "*** SMP "
if [ "$(SMP)" = "1" ];
then echo "Enabled"; else echo "Disabled"; fi
echo "*****"
echo
```

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
$(CFILES:.c=.o): $(HFILES) Makefile
```

```
default:
```

```
make
```

```
endif # ifeq ($(K_VERSION),2.6)
```

```
ifeq (,$(MANDIR))
```

```
# find the best place to install the man page
```

```
MANPATH := $(shell (manpath 2>/dev/null || echo $MANPATH) | sed 's:/ /g')
```

```
ifneq (,$(MANPATH))
```

```
# test based on inclusion in MANPATH
```

```
test_dir = $(findstring $(dir), $(MANPATH))
```

```
else
```

```
# no MANPATH, test based on directory existence
```

```
test_dir = $(shell [ -e $(dir) ] && echo $(dir))
```

```
endif
```

```
# our preferred install path
```

```
# should /usr/local/man be in here ?
```

```
MANDIR := /usr/share/man /usr/man
```

```
MANDIR := $(foreach dir, $(MANDIR), $(test_dir))
```

```
MANDIR := $(firstword $(MANDIR))
```

```
endif
```

```
ifeq (,$(MANDIR))
```

```
# fallback to /usr/man
```

```
MANDIR := /usr/man
```

```
endif
```

```
# depmod version for rpm builds
```

```
DEPVER := $(shell /home/slava/compile/thinstation_src-2.2/bin/depmod -V 2>/dev/null |
```

```
awk 'BEGIN {FS="."} NR==1 {print $$2}')
```

```
#####
```

```
# Build rules
```

```
$(MANFILE).gz: ..$(MANFILE)
```

```
gzip -c $< > $@
```

```
install: default $(MANFILE).gz
```

```
# remove all old versions of the driver
```

```
find /home/slava/compile/thinstation_src-2.2/lib/modules/$(KVER) -name $(TARGET) -exec rm -f {} ; || true
```

```
find /home/slava/compile/thinstation_src-2.2/lib/modules/$(KVER) -name $(TARGET).gz -exec rm -f {} ; || true
```

```
install -D -m 644 $(TARGET) $(INSTALL_MOD_PATH)$(INSTDIR)/$(TARGET)
```

```
ifeq (,$(INSTALL_MOD_PATH))
```

```
/home/slava/compile/thinstation_src-2.2/bin/depmod -a || true
```

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
else
ifeq ($(DEPVER),1 )
/home/slava/compile/thinstation_src-2.2/bin/depmod -r $(INSTALL_MOD_PATH) -a || true
else
/home/slava/compile/thinstation_src-2.2/bin/depmod -b $(INSTALL_MOD_PATH) -a -n >
/dev/null || true
endif
endif
install -D -m 644 $(MANFILE).gz
$(INSTALL_MOD_PATH)$(MANDIR)/man$(MANSECTION)/$(MANFILE).gz
man -c -P'cat > /dev/null' $(MANFILE:.$(MANSECTION)=) || true

uninstall:
if [ -e $(INSTDIR)/$(TARGET) ] ; then
rm -f $(INSTDIR)/$(TARGET) ;
fi
/home/slava/compile/thinstation_src-2.2/bin/depmod -a
if [ -e $(MANDIR)/man$(MANSECTION)/$(MANFILE).gz ] ; then
rm -f $(MANDIR)/man$(MANSECTION)/$(MANFILE).gz ;
fi

.PHONY: clean install

clean:
rm -rf $(TARGET) $(TARGET:.ko=.o) $(TARGET:.ko=.mod.c) $(TARGET:.ko=.mod.o)
$(CFILES:.c=.o) $(MANFILE).gz .*cmd .tmp_versions
```

Перекрестились и запускаем

make install

```
KSRC=/home/slava/compile/thinstation_src-2.2/source/kernel-2.6.16.5/linux-2.6.16.5  
BUILD_KERNEL=2.6.16.5
```

Если ошибок нет, то воздаем хвалу афтару, иначе пишем в комментах свое фе. По идее волшебным образом должна возникнуть папка с дровами

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

```
/home/slava/compile/thinstation_src-2.2/lib/modules/2.6.16.5/kernel/drivers/net/atl1
```

Остались сущие пустяки

```
cd /home/slava/build/Thinstation-2.2/utls
```

```
./update_kernel.sh
```

```
cd /home/slava/build/Thinstation-2.2/
```

И в файл build.conf добавляем вожденную строчку

```
module atl1
```

Собираем рабочий образ

```
./build
```

И пробуем!

Автор:

06.04.12 21:42 - Последнее обновление 06.04.12 21:46

Не забудьте вернуть на родину "родные" компиляторы, они еще могут пригодиться))

```
mv /usr/bin/gcc.orig /usr/bin/gcc
```

```
mv /usr/bin/g++.orig /usr/bin/g++
```

```
mv /usr/bin/cc.orig /usr/bin/cc
```

```
rm -R /usr/local/lib/gcc-lib/i586-pc-linux-gnu/2.95.3
```

```
PATH=$PATHORIG
```

Tags: [thin notes](#)

ИСТОЧНИК: <http://huefisher.livejournal.com/119932.html>

ссылка на материал: http://thin.kiev.ua/index.php?option=com_content&view=article&id=586:-thinstation&catid=43:os&Itemid=67 {jcomments on}