

Raport kompilacji jądra systemu Linux

1. Maciej Mirosław, numer indeksu: 291060, grupa 2

2. Przebieg procesu kompilacji dla metody starej -
localmodconfig

Cały proces rozpoczynamy od pobrania paczki źródeł, które zapisujemy w odpowiednim folderze `/usr/src/`. Najnowsza wersja kernela dostępna w momencie przeprowadzania kompilacji (na dzień 30.05.2021) to **5.12.8**. Pobranie paczki umożliwia wykonanie komendy **wget** oraz podanie jako argument adresu do paczki kernela z oficjalnej strony **kernel.org**.

```
student@Slack:/usr/src$ sudo wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12.8.tar.xz
--2021-05-30 13:34:42-- https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12.8.tar.xz
Translacja cdn.kernel.org... 151.101.113.176, 2a04:4e42:3::432
##czenie si## z cdn.kernel.org[151.101.113.176]:443... po##czono.
##danie HTTP wys##ano, oczekiwanie na odpowied##... 200 OK
D##ugo##: 118153560 (113M) [application/x-xz]
Zapis do: `linux-5.12.8.tar.xz'

linux-5.12.8.tar.xz      100%[=====>] 112,68M   373KB/s   w 6m 46s
2021-05-30 13:41:28 (284 KB/s) - zapisano `linux-5.12.8.tar.xz' [118153560/118153560]

student@Slack:/usr/src$ ls
linux@ linux-4.4.261/ linux-5.12.8.tar.xz
student@Slack:/usr/src$
```

Następnie pobraną paczkę należy rozpakować. Skorzystałem do tego z komendy **tar** z parametrami **-xf**, pominąłem tutaj parametr **-v** (--verbose, pokazujący status rozpakowywania) ze względu na czytelność zrzutu ekranu. Przed procesem zmieniłem również konto użytkownika na **root** aby uniknąć sytuacji w których potrzebna będzie komenda **sudo** i wpisywanie hasła roota.

```
student@Slack:/usr/src$ su root
Has##o:
root@Slack:/usr/src# tar -xf linux-5.12.8.tar.xz
root@Slack:/usr/src# ls -l
razem 115400
lrwxrwxrwx  1 root root      13 kwi 18 17:04 linux -> linux-4.4.261
drwxr-xr-x 25 root root    4096 mar 14 03:17 linux-4.4.261
drwxrwxr-x 24 root root    4096 maj 28 13:18 linux-5.12.8
-rw-r--r--  1 root root 118153560 maj 28 13:33 linux-5.12.8.tar.xz
root@Slack:/usr/src#
```

Mając wypakowane pliki skorzystałem z komendy **zcat** w celu utworzenia pliku konfiguracyjnego **.config** w folderze rozpakowanych plików. Następnie wyświetliłem pierwszych 10 linijek tego pliku za pomocą komendy **head** aby udowodnić jego utworzenie i istnienie. Jak widać na ekranie jest to konfiguracja dla starszej wersji kernela systemu Linux. W naszym wypadku wersja różni się pewnie znacząco aczkolwiek nie pozwoliłem sobie na

modyfikację tego pliku z racji na brak znajomości większości parametrów jak i informacji “DO NOT EDIT.” w 2 linijce pliku. Następnie wywołuje komendę: “**make localmodconfig**”

```
root@Slack:/usr/src/linux-5.12.8# zcat /proc/config.gz > .config
root@Slack:/usr/src/linux-5.12.8# head -10 .config
#
# Automatically generated file; DO NOT EDIT.
# Linux/x86 4.4.261 Kernel Configuration
#
CONFIG_64BIT=y
CONFIG_X86_64=y
CONFIG_X86=y
CONFIG_INSTRUCTION_DECODER=y
CONFIG_PERF_EVENTS_INTEL_UNCORE=y
CONFIG_OUTPUT_FORMAT="elf64-x86-64"
root@Slack:/usr/src/linux-5.12.8# make localmodconfig_
```

Tak wyglądał efekt końcowy komendy.

```
Test find_bit functions (FIND_BIT_BENCHMARK) [N/m/y/?] (NEW)
Test firmware loading via userspace interface (TEST_FIRMWARE) [N/m/y/?] n
sysctl test driver (TEST_SYSCTL) [N/m/y/?] (NEW)
udelay test driver (TEST_UDELAY) [N/m/y/?] n
Test static keys (TEST_STATIC_KEYS) [N/m/?] n
kmod stress tester (TEST_KMOD) [N/m/?] (NEW)
Test memcat_p() helper function (TEST_MEMCAT_P) [N/m/y/?] (NEW)
Test level of stack variable initialization (TEST_STACKINIT) [N/m/y/?] (NEW)
Test heap/page initialization (TEST_MEMINIT) [N/m/y/?] (NEW)
Test freeing pages (TEST_FREE_PAGES) [N/m/y/?] (NEW)
Test floating point operations in kernel space (TEST_FPU) [N/m/y/?] (NEW)
Memtest (MEMTEST) [Y/n/?] y
#
# configuration written to .config
#
root@Slack:/usr/src/linux-5.12.8#
```

Podczas wykonywania tej komendy system pytał o wiele różnych opcji. Zostawiłem je domyślne (zatwierdzając wszystkie przyciskiem ENTER) ponieważ nie posiadam wiedzy pozwalającej na dokładne dostosowanie konfiguracji do mojego systemu.

Następnym krokiem będzie wykonanie kolejnych komend **make** widocznych na zrzutach.

```
root@Slack:/usr/src/linux-5.12.8# make -j4 bzImage
```

W tym przypadku omyłkowo skorzystałem z parametru -j4, który powinien wykorzystać 4 procesory niestety nie zorientowałem się na czas, że przydzielone są tylko dwa. Natomiast komenda wykonała się bez żadnych błędów.

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
CC arch/x86/boot/video-mode.o
CC arch/x86/boot/compressed/string.o
CC arch/x86/boot/video-uga.o
CC arch/x86/boot/version.o
CC arch/x86/boot/video-vesa.o
CC arch/x86/boot/compressed/cmdline.o
CC arch/x86/boot/video-bios.o
CC arch/x86/boot/compressed/error.o
HOSTCC arch/x86/boot/tools/build
OBJCOPY arch/x86/boot/compressed/vmlinux.bin
RELOCS arch/x86/boot/compressed/vmlinux.relocs
HOSTCC arch/x86/boot/compressed/mkpiggy
CC arch/x86/boot/compressed/cpuflags.o
CC arch/x86/boot/compressed/early_serial_console.o
CPUSTR arch/x86/boot/cpustr.h
CC arch/x86/boot/compressed/ident_map_64.o
CC arch/x86/boot/compressed/kaslr.o
CC arch/x86/boot/compressed/idt_64.o
AS arch/x86/boot/compressed/idt_handlers_64.o
AS arch/x86/boot/compressed/mem_encrypt.o
CC arch/x86/boot/compressed/pgtable_64.o
CC arch/x86/boot/compressed/acpi.o
AS arch/x86/boot/compressed/efi_thunk_64.o
CC arch/x86/boot/cpu.o
CC arch/x86/boot/compressed/misc.o
LZMA arch/x86/boot/compressed/vmlinux.bin.lzma
MKPIGGY arch/x86/boot/compressed/piggy.S
AS arch/x86/boot/compressed/piggy.o
LD arch/x86/boot/compressed/vmlinux
ZOFFSET arch/x86/boot/zoffset.h
OBJCOPY arch/x86/boot/vmlinux.bin
AS arch/x86/boot/header.o
LD arch/x86/boot/setup.elf
OBJCOPY arch/x86/boot/setup.bin
BUILD arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready (#1)
root@Slack:/usr/src/linux-5.12.8#
```

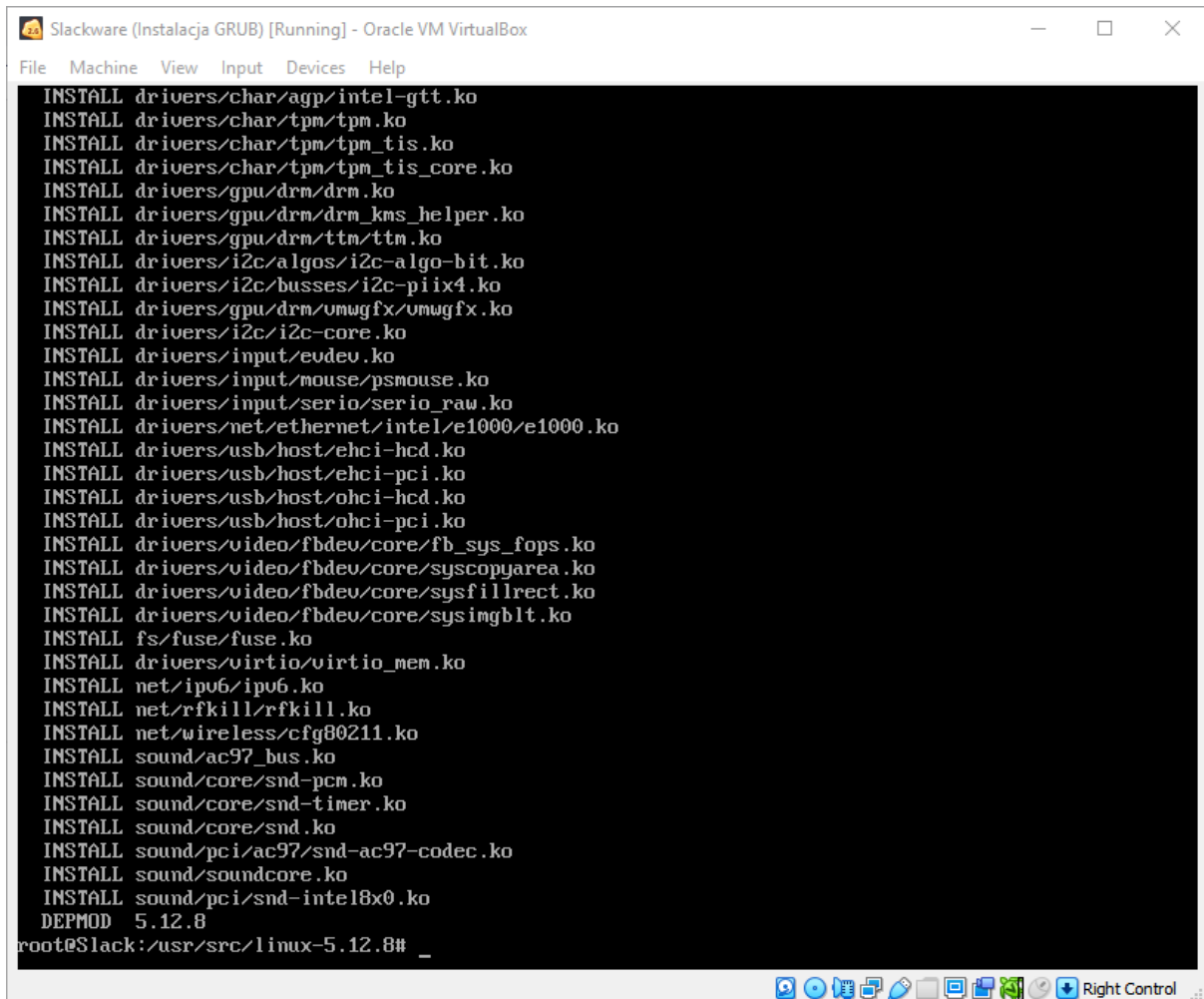
Następną komendą było wykonanie kompilacji modułów jądra.

```
root@Slack:/usr/src/linux-5.12.8# make -j2 modules
DESCEND objtool
CALL scripts/atomic/check-atomics.sh
CALL scripts/checksyscalls.sh
CC [M] fs/fuse/dev.o
CC [M] fs/fuse/dir.o
```

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
LD [M] drivers/char/agp/intel-agp.ko
LD [M] drivers/char/agp/intel-gtt.ko
LD [M] drivers/char/tpm/tpm.ko
LD [M] drivers/char/tpm/tpm_tis.ko
LD [M] drivers/char/tpm/tpm_tis_core.ko
LD [M] drivers/gpu/drm/drm.ko
LD [M] drivers/gpu/drm/drm_kms_helper.ko
LD [M] drivers/gpu/drm/ttm/ttm.ko
LD [M] drivers/i2c/algos/i2c-algo-bit.ko
LD [M] drivers/gpu/drm/umugfx/umugfx.ko
LD [M] drivers/i2c/busses/i2c-piix4.ko
LD [M] drivers/i2c/i2c-core.ko
LD [M] drivers/input/mouse/psmouse.ko
LD [M] drivers/input/evdev.ko
LD [M] drivers/input/serio/serio_raw.ko
LD [M] drivers/net/ethernet/intel/e1000/e1000.ko
LD [M] drivers/usb/host/ehci-hcd.ko
LD [M] drivers/usb/host/ohci-hcd.ko
LD [M] drivers/usb/host/ehci-pci.ko
LD [M] drivers/usb/host/ohci-pci.ko
LD [M] drivers/video/fbdev/core/fb_sys_fops.ko
LD [M] drivers/video/fbdev/core/syscopyarea.ko
LD [M] drivers/video/fbdev/core/sysfillrect.ko
LD [M] drivers/virtio/virtio_mem.ko
LD [M] drivers/video/fbdev/core/sysimgblt.ko
LD [M] fs/fuse/fuse.ko
LD [M] net/ipv6/ipv6.ko
LD [M] net/rfkill/rfkill.ko
LD [M] net/wireless/cfg80211.ko
LD [M] sound/ac97_bus.ko
LD [M] sound/core/snd-pcm.ko
LD [M] sound/core/snd-timer.ko
LD [M] sound/pci/ac97/snd-ac97-codec.ko
LD [M] sound/pci/snd-intel8x0.ko
LD [M] sound/core/snd.ko
LD [M] sound/soundcore.ko
root@Slack:/usr/src/linux-5.12.8# _
```

Skompilowane moduły należało następnie zainstalować również komendą **make**.

```
root@Slack:/usr/src/linux-5.12.8# make -j2 modules_install
```

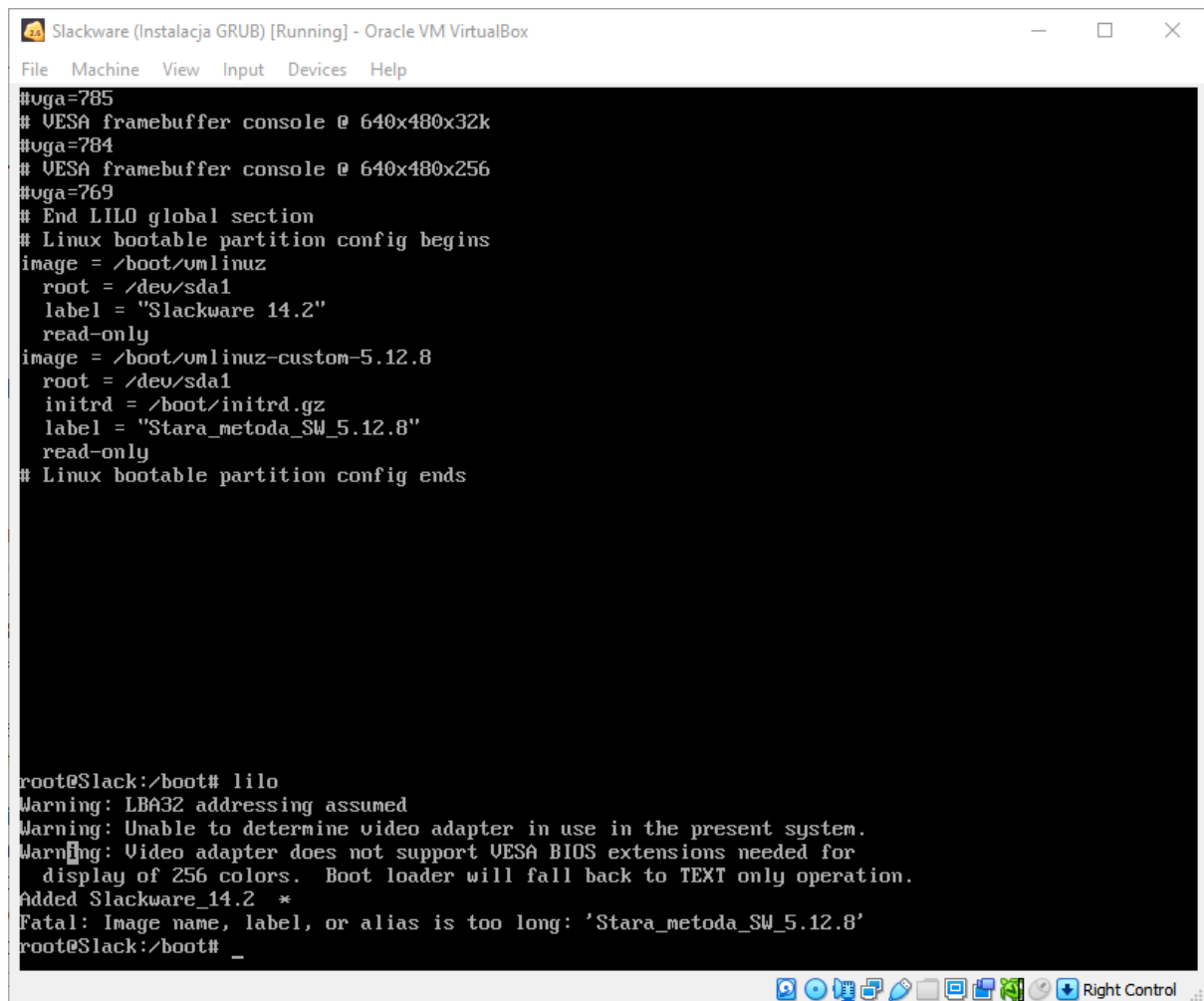


```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
INSTALL drivers/char/agp/intel-gtt.ko
INSTALL drivers/char/tpm/tpm.ko
INSTALL drivers/char/tpm/tpm_tis.ko
INSTALL drivers/char/tpm/tpm_tis_core.ko
INSTALL drivers/gpu/drm/drm.ko
INSTALL drivers/gpu/drm/drm_kms_helper.ko
INSTALL drivers/gpu/drm/ttm/ttm.ko
INSTALL drivers/i2c/algos/i2c-algo-bit.ko
INSTALL drivers/i2c/busses/i2c-piix4.ko
INSTALL drivers/gpu/drm/umugfx/umugfx.ko
INSTALL drivers/i2c/i2c-core.ko
INSTALL drivers/input/evdev.ko
INSTALL drivers/input/mouse/psmouse.ko
INSTALL drivers/input/serio/serio_raw.ko
INSTALL drivers/net/ethernet/intel/e1000/e1000.ko
INSTALL drivers/usb/host/ehci-hcd.ko
INSTALL drivers/usb/host/ehci-pci.ko
INSTALL drivers/usb/host/ohci-hcd.ko
INSTALL drivers/usb/host/ohci-pci.ko
INSTALL drivers/video/fbdev/core/fb_sys_fops.ko
INSTALL drivers/video/fbdev/core/syscopyarea.ko
INSTALL drivers/video/fbdev/core/sysfillrect.ko
INSTALL drivers/video/fbdev/core/sysimgblt.ko
INSTALL fs/fuse/fuse.ko
INSTALL drivers/virtio/virtio_mem.ko
INSTALL net/ipv6/ipv6.ko
INSTALL net/rfkill/rfkill.ko
INSTALL net/wireless/cfg80211.ko
INSTALL sound/ac97_bus.ko
INSTALL sound/core/snd-pcm.ko
INSTALL sound/core/snd-timer.ko
INSTALL sound/core/snd.ko
INSTALL sound/pci/ac97/snd-ac97-codec.ko
INSTALL sound/soundcore.ko
INSTALL sound/pci/snd-intel8x0.ko
DEPMOD 5.12.8
root@Slack:/usr/src/linux-5.12.8#
```

Następnym krokiem było przekopiowanie skompilowanych plików do folderu **/boot**.
Utworzenie im dowiązań do odpowiednich plików i następnie wygenerowanie komendy **ramdisk** i uruchomienie jej w celu wygenerowania pliku **initrd.gz**

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Documentation README fs modules-only.symvers security vmlinux.symvers
Kbuild System.map include modules.builtin sound
Kconfig arch init modules.builtin.modinfo tools
LICENSES block ipc modules.order usr
MAINTAINERS certs kernel net virt
root@Slack:/usr/src/linux-5.12.8# cp arch/x86/boot/bzImage /boot/vmlinuz-custom-5.12.8
root@Slack:/usr/src/linux-5.12.8# cp System.map /boot/System.map-custom-5.12.8
root@Slack:/usr/src/linux-5.12.8# cp .config /boot/config-custom-5.12.8
root@Slack:/usr/src/linux-5.12.8# cd /boot
root@Slack:/boot# ls
README.initrd config inside.dat vmlinuz-custom-5.12.8
System.map config-custom-5.12.8 map vmlinuz-generic
System.map-custom-5.12.8 config-generic-4.4.261.x64 onlyblue.bmp vmlinuz-generic-4.4.261
System.map-generic-4.4.261 config-huge-4.4.261.x64 onlyblue.dat vmlinuz-huge
System.map-huge-4.4.261 elilo-ia32.efi slack.bmp vmlinuz-huge-4.4.261
boot.0800 elilo-x86_64.efi tuxlogo.bmp
boot_message.txt grub tuxlogo.dat
coffee.dat inside.bmp vmlinuz
root@Slack:/boot# ln -sf vmlinuz-custom-5.12.8 vmlinuz
root@Slack:/boot# ln -sf System.map-custom-5.12.8 System.map
root@Slack:/boot# ln -sf config-custom-5.12.8 config
root@Slack:/boot# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.12.8
#
# mkinitrd_command_generator.sh revision 1.45
#
# This script will now make a recommendation about the command to use
# in case you require an initrd image to boot a kernel that does not
# have support for your storage or root filesystem built in
# (such as the Slackware 'generic' kernels').
# A suitable 'mkinitrd' command will be:
mkinitrd -c -k 5.12.8 -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd.gz
root@Slack:/boot# mkinitrd -c -k 5.12.8 -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd.gz
33415 bloków
/boot/initrd.gz created.
Be sure to run lilo again if you use it.
root@Slack:/boot#
```

Jak widać na powyższym screenie komenda **mkinitrd** wyrzuciła nam informacje o uruchomieniu programu lilo w przypadku gdy go używamy. Zanim jednak wykonamy komendę lilo należy dodać wpis dla naszego nowego kernela.



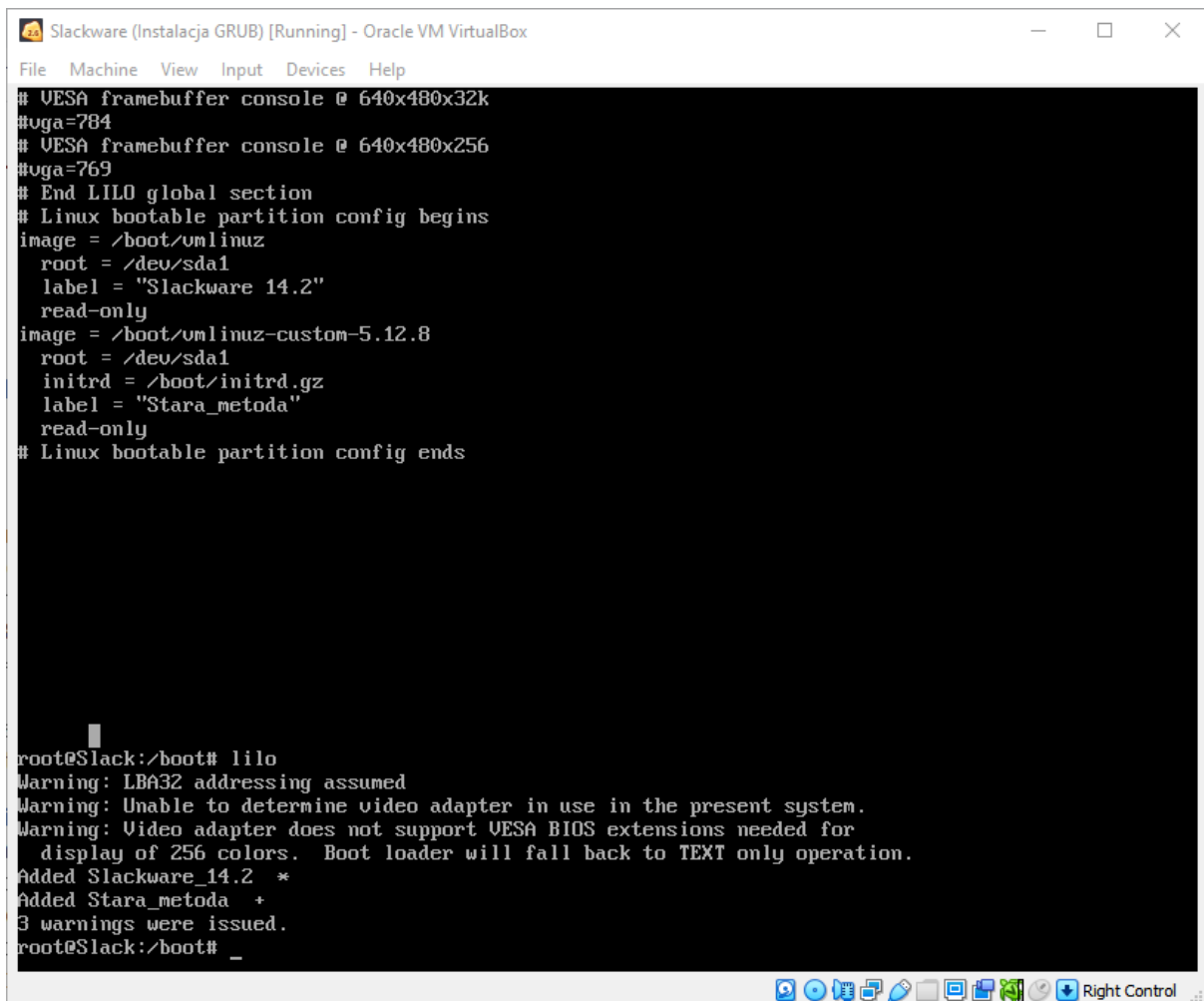
The screenshot shows a terminal window titled "Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox". The terminal displays the following commands and output:

```
#vga=785
# UESA framebuffer console @ 640x480x32k
#vga=784
# UESA framebuffer console @ 640x480x256
#vga=769
# End LILO global section
# Linux bootable partition config begins
image = /boot/vmlinuz
  root = /dev/sda1
  label = "Slackware 14.2"
  read-only
image = /boot/vmlinuz-custom-5.12.8
  root = /dev/sda1
  initrd = /boot/initrd.gz
  label = "Stara_metoda_SW_5.12.8"
  read-only
# Linux bootable partition config ends

root@Slack:/boot# lilo
Warning: LBA32 addressing assumed
Warning: Unable to determine video adapter in use in the present system.
Warning: Video adapter does not support VESA BIOS extensions needed for
display of 256 colors. Boot loader will fall back to TEXT only operation.
Added Slackware_14.2 *
Fatal: Image name, label, or alias is too long: 'Stara_metoda_SW_5.12.8'
root@Slack:/boot# _
```

The error message "Fatal: Image name, label, or alias is too long: 'Stara_metoda_SW_5.12.8'" indicates that the label for the second kernel image is too long for the boot loader to handle.

Tutaj natrafiłem na błędy, podałem zbyt długą nazwę dla obrazu - chciałem długą i opisową nazwę wraz z wersją kernela niestety musiałem ją zmienić na coś krótszego.

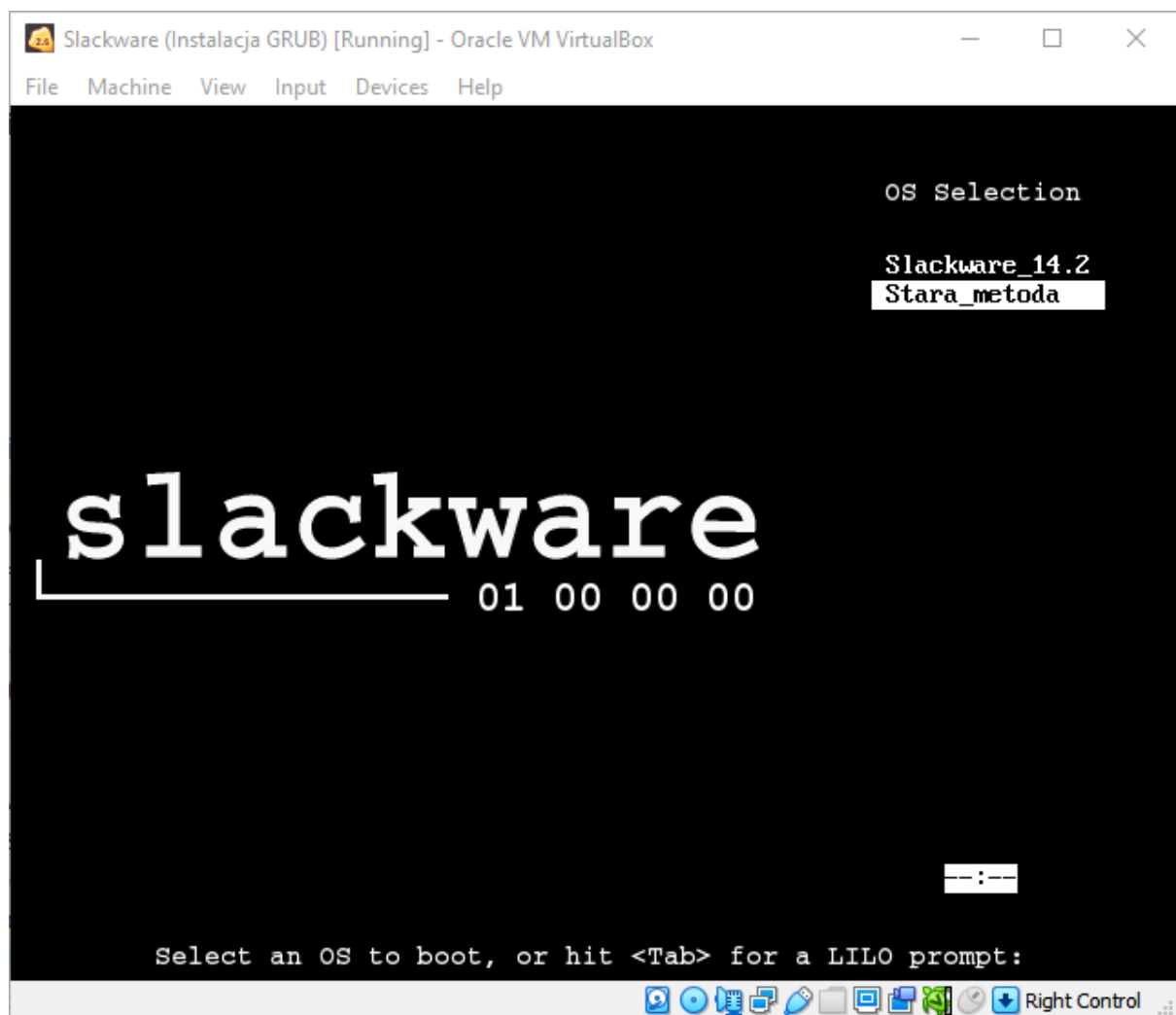


```
# VESA framebuffer console @ 640x480x32k
#vga=784
# VESA framebuffer console @ 640x480x256
#vga=769
# End LIL0 global section
# Linux bootable partition config begins
image = /boot/umlunuz
  root = /dev/sda1
  label = "Slackware 14.2"
  read-only
image = /boot/umlunuz-custom-5.12.8
  root = /dev/sda1
  initrd = /boot/initrd.gz
  label = "Stara_metoda"
  read-only
# Linux bootable partition config ends

root@Slack:/boot# lilo
Warning: LBA32 addressing assumed
Warning: Unable to determine video adapter in use in the present system.
Warning: Video adapter does not support VESA BIOS extensions needed for
display of 256 colors. Boot loader will fall back to TEXT only operation.
Added Slackware_14.2 *
Added Stara_metoda +
3 warnings were issued.
root@Slack:/boot# _
```

Jak widać poprawiłem nazwę, lilo wypuściło na standardowe wyjście informację o 3 ostrzeżeniach.

Następnym krokiem było zrestartowanie maszyny i zweryfikowanie czy maszyna z nowym kernelem się uruchomi i czy będzie w stanie się do niej zalogować.



Jak widać nowy wpis wyświetla się poprawnie.

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
eth0: waiting for carrier
eth0: carrier acquired
all: IPv6 kernel autoconf disabled
DUID 00:01:00:01:28:0e:e1:30:08:00:27:a3:9e:be
eth0: IAID 27:a3:9e:be
eth0: rebinding lease of 10.0.2.15
eth0: leased 10.0.2.15 for 86400 seconds
eth0: adding route to 10.0.2.0/24
eth0: adding default route via 10.0.2.2
forked to background, child pid 655
Polling for DHCP server on interface eth1:
all: IPv6 kernel autoconf disabled
eth1: adding address fe80::a717:14d8:b215:c904
if_addaddress6: Operation not supported
eth1: waiting for carrier
eth1: carrier acquired
all: IPv6 kernel autoconf disabled
DUID 00:01:00:01:28:0e:e1:30:08:00:27:a3:9e:be
eth1: IAID 27:49:55:3e
eth1: rebinding lease of 192.168.56.101
eth1: leased 192.168.56.101 for 600 seconds
eth1: adding route to 192.168.56.0/24
forked to background, child pid 712
Starting system message bus: /usr/bin/dbus-uuidgen --ensure ; /usr/bin/dbus-daemon --system
Starting Internet super-server daemon: /usr/sbin/inetd
Starting OpenSSH SSH daemon: /usr/sbin/sshd
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
Starting ACPI daemon: /usr/sbin/acpid
Updating MIME database: /usr/bin/update-mime-database /usr/share/mime &
Starting ConsoleKit daemon: /usr/sbin/console-kit-daemon
Updating gtk.immodules:
  /usr/bin/update-gtk-immodules &
Updating gdk-pixbuf loaders:
  /usr/bin/update-gdk-pixbuf-loaders &
Compiling GSettings XML schema files:
  /usr/bin/glib-compile-schemas /usr/share/glib-2.0/schemas &
Loading /usr/share/kbd/keymaps/i386/german/pl.map.gz
Starting gpm: /usr/sbin/gpm -m /dev/mouse -t imps2

Welcome to Linux 5.12.8 (tty1)

Slack login: root
Password:
Linux 5.12.8.
Last login: Sun May 30 13:03:00 +0200 2021 on /dev/tty1.
You have mail.
root@Slack:~#
```

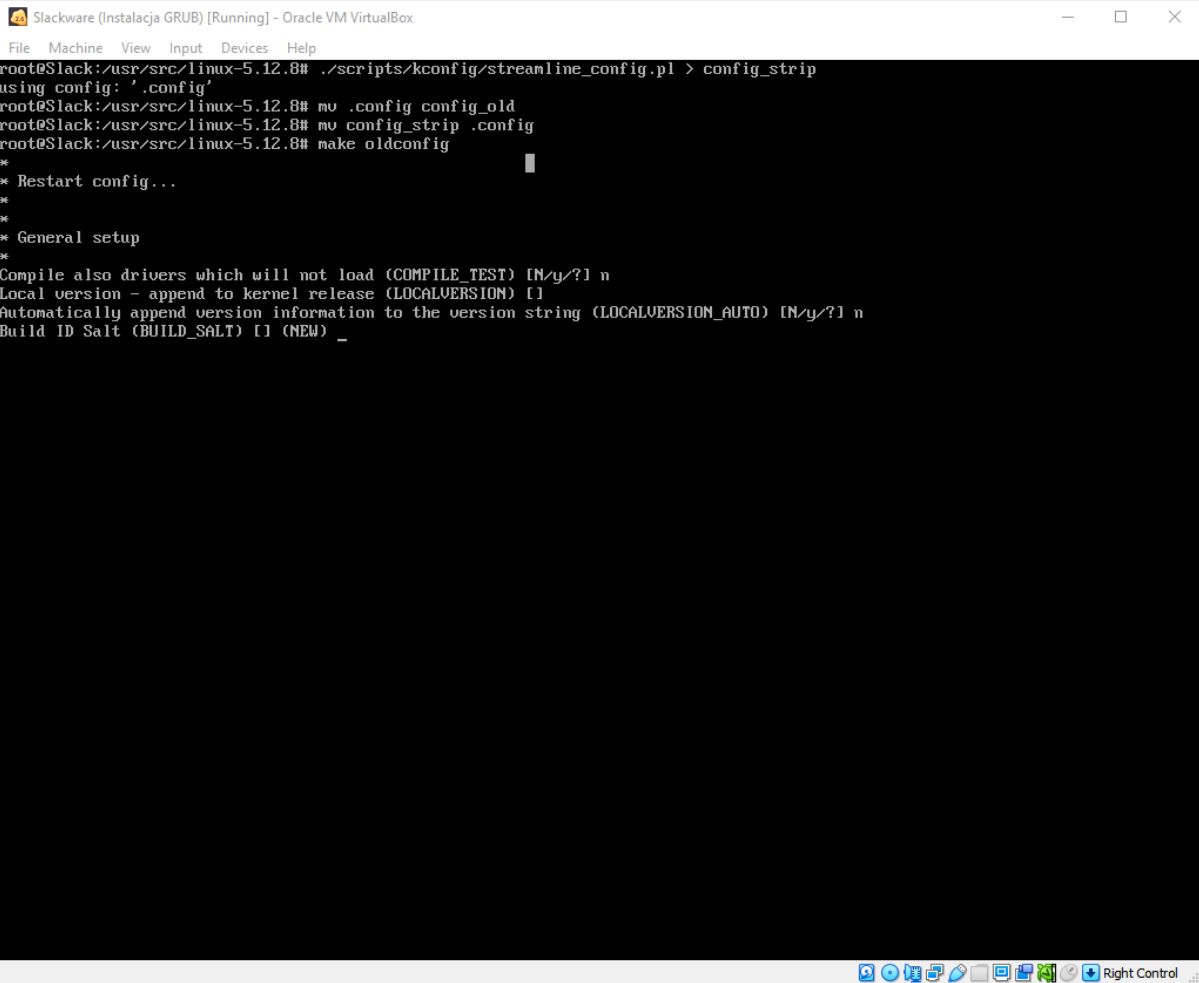
Zalogować udało się także bez problemu. Pod linijką z Password widnieje linijka z wersją naszego kernela.

3. Nowa metoda - streamline_config.pl

Kompiłacje kernels metodą streamline rozpoczynamy od utworzenia pliku konfiguracyjnego. Korzystamy do tego ze skryptu **streamline_config.pl**, który znajduje się w folderze **/scripts/kconfig/**

```
root@Slack:/usr/src/linux-5.12.8# ls
COPYING      LICENSES/    System.map  drivers/    kernel/      modules.builtin.modinfo  security/  umlinux*
CREDITS      MAINTAINERS  arch/       fs/         lib/         modules.order            sound/     umlinux.o
Documentation/ Makefile     block/     include/    mm/         net/                    tools/     umlinux.symvers
Kbuild       Module.symvers  certs/    init/       modules-only.symvers  samples/  usr/
Kconfig      README        crypto/    ipc/        modules.builtin  scripts/  virt/
root@Slack:/usr/src/linux-5.12.8# ./scripts/kconfig/streamline_config.pl > config_strip
using config: '.config'
root@Slack:/usr/src/linux-5.12.8#
```

Następnie zapisałem nasz plik konfiguracyjny **.config** jako **config_old** aby zachować jego kopię. Zamieniłem nawę pliku **config_strip** na **.config** i wykonałem komendę **make oldconfig**.

A screenshot of a Slackware VM window titled "Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox". The window shows a terminal session with the following commands and output:

```
root@Slack:/usr/src/linux-5.12.8# ./scripts/kconfig/streamline_config.pl > config_strip
using config: '.config'
root@Slack:/usr/src/linux-5.12.8# mv .config config_old
root@Slack:/usr/src/linux-5.12.8# mv config_strip .config
root@Slack:/usr/src/linux-5.12.8# make oldconfig

*
* Restart config...
*
*
* General setup
*
Compile also drivers which will not load (CONFIG_COMPILE_TEST) [N/y/?] n
Local version - append to kernel release (LOCALVERSION) []
Automatically append version information to the version string (LOCALVERSION_AUTO) [N/y/?] n
Build ID Salt (BUILD_SALT) [] (NEW) _
```

The window has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". At the bottom right, there is a "Right Control" button.

Jak widać komenda podobnie jak w starej metodzie pyta nas o pewne ustawienia. W tym przypadku również zostawiłem je domyślne.

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Test functions located in the string_helpers module at runtime (TEST_STRING_HELPERS) [N/m/y/?] n
Test strscpy*() family of functions at runtime (TEST_STRSCPY) [N/m/y/?] (NEW)
Test kstrto*() family of functions at runtime (TEST_KSTRTOX) [N/m/y/?] n
Test printf() family of functions at runtime (TEST_PRINTF) [N/m/y/?] n
Test bitmap_*() family of functions at runtime (TEST_BITMAP) [N/m/y/?] (NEW)
Test functions located in the uuid module at runtime (TEST_UUID) [N/m/y/?] (NEW)
Test the XArray code at runtime (TEST_XARRAY) [N/m/y/?] (NEW)
Test check_*_overflow() functions at runtime (TEST_OVERFLOW) [N/m/y/?] (NEW)
Perform selftest on resizable hash table (TEST_RHASHTABLE) [N/m/y/?] n
Perform selftest on hash functions (TEST_HASH) [N/m/y/?] n
Perform selftest on IDA functions (TEST_IDA) [N/m/y/?] (NEW)
Test module loading with 'hello world' module (TEST_LKM) [N/m/?] n
Test module for compilation of bitops operations (TEST_BITOPS) [N/m/?] (NEW)
Test module for stress/performance analysis of umalloc allocator (TEST_UMALLOC) [N/m/?] (NEW)
Test user/kernel boundary protections (TEST_USER_COPY) [N/m/?] n
Test BPF filter functionality (TEST_BPF) [N/m/?] n
Test blackhole netdev functionality (TEST_BLACKHOLE_DEV) [N/m/?] (NEW)
Test find_bit functions (FIND_BIT_BENCHMARK) [N/m/y/?] (NEW)
Test firmware loading via userspace interface (TEST_FIRMWARE) [N/m/y/?] n
sysctl test driver (TEST_SYSCTL) [N/m/y/?] (NEW)
udelay test driver (TEST_UDELAY) [N/m/y/?] n
Test static keys (TEST_STATIC_KEYS) [N/m/?] n
kmod stress tester (TEST_KMOD) [N/m/?] (NEW)
Test memcat_p() helper function (TEST_MEMCAT_P) [N/m/y/?] (NEW)
Test level of stack variable initialization (TEST_STACKINIT) [N/m/y/?] (NEW)
Test heap/page initialization (TEST_MEMINIT) [N/m/y/?] (NEW)
Test freeing pages (TEST_FREE_PAGES) [N/m/y/?] (NEW)
Test floating point operations in kernel space (TEST_FPU) [N/m/y/?] (NEW)
Memtest (MEMTEST) [Y/n/?] y
#
# configuration written to .config
#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
root@Slack:/usr/src/linux-5.12.8#
Right Control
```

Następnym krokiem będzie wykonanie komendy **make** dla: bzImage, modules oraz modules_install - analogicznie jak w starej metodzie. Tym razem dla każdej komendy zastosowałem parametr **-j2**.

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@Slack:/usr/src/linux-5.12.8# make -j2 bzImage
SYSHDR arch/x86/include/generated/uapi/asm/unistd_32.h
WRAP arch/x86/include/generated/uapi/asm/bpf_perf_event.h
WRAP arch/x86/include/generated/uapi/asm/errno.h
WRAP arch/x86/include/generated/uapi/asm/fcntl.h
WRAP arch/x86/include/generated/uapi/asm/ioctl.h
WRAP arch/x86/include/generated/uapi/asm/ioctls.h
WRAP arch/x86/include/generated/uapi/asm/ipcbuf.h
WRAP arch/x86/include/generated/uapi/asm/param.h
WRAP arch/x86/include/generated/uapi/asm/poll.h
SYSHDR arch/x86/include/generated/uapi/asm/unistd_64.h
WRAP arch/x86/include/generated/uapi/asm/resource.h
WRAP arch/x86/include/generated/uapi/asm/socket.h
WRAP arch/x86/include/generated/uapi/asm/sockios.h
WRAP arch/x86/include/generated/uapi/asm/termbits.h
WRAP arch/x86/include/generated/uapi/asm/termios.h
WRAP arch/x86/include/generated/uapi/asm/types.h
SYSHDR arch/x86/include/generated/uapi/asm/unistd_x32.h
HOSTCC arch/x86/tools/relocs_32.o
SYSTBL arch/x86/include/generated/asm/syscalls_32.h
HOSTCC arch/x86/tools/relocs_64.o
HOSTCC arch/x86/tools/relocs_common.o
HOSTLD arch/x86/tools/relocs
UPD include/config/kernel.release
WRAP arch/x86/include/generated/asm/early_ioremap.h
WRAP arch/x86/include/generated/asm/export.h
WRAP arch/x86/include/generated/asm/mcs_spinlock.h
WRAP arch/x86/include/generated/asm/irq_regs.h
WRAP arch/x86/include/generated/asm/kmap_size.h
WRAP arch/x86/include/generated/asm/local64.h
WRAP arch/x86/include/generated/asm/mmioub.h
WRAP arch/x86/include/generated/asm/module.lds.h
WRAP arch/x86/include/generated/asm/ruonce.h
UPD include/generated/uapi/linux/version.h
UPD include/generated/utsrelease.h
HOSTCC scripts/bin2c
HOSTCC scripts/kallsyms
HOSTCC scripts/sorttable
SYSHDR arch/x86/include/generated/asm/unistd_32_ia32.h
SYSHDR arch/x86/include/generated/asm/unistd_64_x32.h
SYSTBL arch/x86/include/generated/asm/syscalls_64.h
HOSTCC scripts/asn1_compiler
```

A taki był wynik wykonania komendy:

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
AS arch/x86/boot/compressed/kernel_info.o
CC arch/x86/boot/memory.o
AS arch/x86/boot/compressed/head_64.o
VOFFSET arch/x86/boot/compressed/..voffset.h
CC arch/x86/boot/pm.o
AS arch/x86/boot/pmjump.o
CC arch/x86/boot/printf.o
CC arch/x86/boot/regs.o
CC arch/x86/boot/compressed/string.o
CC arch/x86/boot/string.o
CC arch/x86/boot/compressed/cmdline.o
CC arch/x86/boot/tty.o
CC arch/x86/boot/video.o
CC arch/x86/boot/compressed/error.o
CC arch/x86/boot/video-mode.o
CC arch/x86/boot/version.o
CC arch/x86/boot/video-vga.o
OBJCOPY arch/x86/boot/compressed/vmlinux.bin
CC arch/x86/boot/video-vesa.o
CC arch/x86/boot/video-bios.o
RELOCS arch/x86/boot/compressed/vmlinux.relocs
HOSTCC arch/x86/boot/tools/build
HOSTCC arch/x86/boot/compressed/mkpiggy
CC arch/x86/boot/compressed/cpuflags.o
CPUSTR arch/x86/boot/cpustr.h
CC arch/x86/boot/cpu.o
CC arch/x86/boot/compressed/early_serial_console.o
CC arch/x86/boot/compressed/kaslr.o
CC arch/x86/boot/compressed/ident_map_64.o
CC arch/x86/boot/compressed/idt_64.o
AS arch/x86/boot/compressed/idt_handlers_64.o
AS arch/x86/boot/compressed/mem_encrypt.o
CC arch/x86/boot/compressed/pgtable_64.o
CC arch/x86/boot/compressed/acpi.o
AS arch/x86/boot/compressed/efi_thunk_64.o
CC arch/x86/boot/compressed/misc.o
LZMA arch/x86/boot/compressed/vmlinux.bin.lzma
MKPIGGY arch/x86/boot/compressed/piggy.S
AS arch/x86/boot/compressed/piggy.o
LD arch/x86/boot/compressed/vmlinux
ZOFFSET arch/x86/boot/zoffset.h
OBJCOPY arch/x86/boot/vmlinux.bin
AS arch/x86/boot/header.o
LD arch/x86/boot/setup.elf
OBJCOPY arch/x86/boot/setup.bin
BUILD arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready (#1)
root@Slack:/usr/src/linux-5.12.8#
```

Następnie wykonałem komendę: **make -j2 modules**. Taki był jej wynik:

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
CC [M] net/ipv6/ipv6.mod.o
CC [M] net/rfkill/rfkill.mod.o
CC [M] net/wireless/cfg80211.mod.o
CC [M] sound/ac97_bus.mod.o
CC [M] sound/core/snd-pcm.mod.o
CC [M] sound/core/snd-timer.mod.o
CC [M] sound/core/snd.mod.o
CC [M] sound/pci/ac97/snd-ac97-codec.mod.o
CC [M] sound/pci/snd-intel8x0.mod.o
CC [M] sound/soundcore.mod.o
LD [M] drivers/acpi/ac.ko
LD [M] drivers/acpi/button.ko
LD [M] drivers/acpi/video.ko
LD [M] drivers/block/loop.ko
LD [M] drivers/char/agp/agpgart.ko
LD [M] drivers/char/agp/intel-agp.ko
LD [M] drivers/char/agp/intel-gtt.ko
LD [M] drivers/gpu/drm/drm.ko
LD [M] drivers/gpu/drm/drm_kms_helper.ko
LD [M] drivers/gpu/drm/ttm/ttm.ko
LD [M] drivers/gpu/drm/umugfx/umugfx.ko
LD [M] drivers/i2c/algos/i2c-algo-bit.ko
LD [M] drivers/i2c/busses/i2c-piix4.ko
LD [M] drivers/i2c/i2c-core.ko
LD [M] drivers/input/evdev.ko
LD [M] drivers/input/mouse/psmouse.ko
LD [M] drivers/input/serio/serio_raw.ko
LD [M] drivers/net/ethernet/intel/e1000/e1000.ko
LD [M] drivers/usb/host/ehci-hcd.ko
LD [M] drivers/usb/host/ehci-pci.ko
LD [M] drivers/usb/host/ohci-hcd.ko
LD [M] drivers/usb/host/ohci-pci.ko
LD [M] drivers/video/fbdev/core/fb_sys_fops.ko
LD [M] drivers/video/fbdev/core/syscopyarea.ko
LD [M] drivers/video/fbdev/core/sysfillrect.ko
LD [M] drivers/video/fbdev/core/sysimgblt.ko
LD [M] fs/fuse/fuse.ko
LD [M] net/ipv6/ipv6.ko
LD [M] net/wireless/cfg80211.ko
LD [M] net/rfkill/rfkill.ko
LD [M] sound/ac97_bus.ko
LD [M] sound/core/snd-pcm.ko
LD [M] sound/core/snd-timer.ko
LD [M] sound/core/snd.ko
LD [M] sound/pci/ac97/snd-ac97-codec.ko
LD [M] sound/pci/snd-intel8x0.ko
LD [M] sound/soundcore.ko
root@Slack:/usr/src/linux-5.12.8#
```

Następnie instalacja modułów komendą: **make -j2 modules_install**.

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
LD [M] net/rfkill/rfkill.ko
LD [M] sound/ac97_bus.ko
LD [M] sound/core/snd-pcm.ko
LD [M] sound/core/snd-timer.ko
LD [M] sound/core/snd.ko
LD [M] sound/pci/ac97/snd-ac97-codec.ko
LD [M] sound/pci/snd-intel8x0.ko
LD [M] sound/soundcore.ko
root@Slack:/usr/src/linux-5.12.8# make -j2 modules_install
INSTALL drivers/acpi/ac.ko
INSTALL drivers/acpi/button.ko
INSTALL drivers/acpi/video.ko
INSTALL drivers/block/loop.ko
INSTALL drivers/char/agp/agpgart.ko
INSTALL drivers/char/agp/intel-agp.ko
INSTALL drivers/char/agp/intel-gtt.ko
INSTALL drivers/gpu/drm/drm.ko
INSTALL drivers/gpu/drm/drm_kms_helper.ko
INSTALL drivers/gpu/drm/ttm/ttm.ko
INSTALL drivers/gpu/drm/vmwgfx/vmwgfx.ko
INSTALL drivers/i2c/algos/i2c-algo-bit.ko
INSTALL drivers/i2c/busses/i2c-piix4.ko
INSTALL drivers/i2c/i2c-core.ko
INSTALL drivers/input/evdev.ko
INSTALL drivers/input/mouse/psmouse.ko
INSTALL drivers/input/serio/serio_raw.ko
INSTALL drivers/net/ethernet/intel/e1000/e1000.ko
INSTALL drivers/usb/host/ehci-hcd.ko
INSTALL drivers/usb/host/ehci-pci.ko
INSTALL drivers/usb/host/ohci-hcd.ko
INSTALL drivers/usb/host/ohci-pci.ko
INSTALL drivers/video/fbdev/core/fb_sys_fops.ko
INSTALL drivers/video/fbdev/core/syscopyarea.ko
INSTALL drivers/video/fbdev/core/sysimgblt.ko
INSTALL drivers/video/fbdev/core/sysfillrect.ko
INSTALL fs/fuse/fuse.ko
INSTALL net/ipv6/ipv6.ko
INSTALL net/rfkill/rfkill.ko
INSTALL net/wireless/cfg80211.ko
INSTALL sound/ac97_bus.ko
INSTALL sound/core/snd-pcm.ko
INSTALL sound/core/snd-timer.ko
INSTALL sound/core/snd.ko
INSTALL sound/pci/ac97/snd-ac97-codec.ko
INSTALL sound/pci/snd-intel8x0.ko
INSTALL sound/soundcore.ko
DEPMOD 5.12.8
root@Slack:/usr/src/linux-5.12.8#
```

Po zainstalowaniu modułów należy utworzyć odpowiednie dowiązania oraz przeniesienie odpowiednich plików do folderu /boot.


```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
LD [M] sound/soundcore.ko
root@Slack:/usr/src/linux-5.12.8# make -j2 modules_install
INSTALL drivers/acpi/ac.ko
INSTALL drivers/acpi/button.ko
INSTALL drivers/acpi/video.ko
INSTALL drivers/block/loop.ko
INSTALL drivers/char/agp/agpgart.ko
INSTALL drivers/char/agp/intel-agp.ko
INSTALL drivers/char/agp/intel-gtt.ko
INSTALL drivers/gpu/drm/drm.ko
INSTALL drivers/gpu/drm/drm_kms_helper.ko
INSTALL drivers/gpu/drm/ttm/ttm.ko
INSTALL drivers/gpu/drm/vmwgfx/vmwgfx.ko
INSTALL drivers/i2c/algos/i2c-algo-bit.ko
INSTALL drivers/i2c/busses/i2c-piix4.ko
INSTALL drivers/i2c/i2c-core.ko
INSTALL drivers/input/evdev.ko
INSTALL drivers/input/mouse/psmouse.ko
INSTALL drivers/input/serio/serio_raw.ko
INSTALL drivers/net/ethernet/intel/e1000/e1000.ko
INSTALL drivers/usb/host/ehci-hcd.ko
INSTALL drivers/usb/host/ehci-pci.ko
INSTALL drivers/usb/host/ohci-hcd.ko
INSTALL drivers/usb/host/ohci-pci.ko
INSTALL drivers/video/fbdev/core/fb_sys_fops.ko
INSTALL drivers/video/fbdev/core/syscopyarea.ko
INSTALL drivers/video/fbdev/core/sysimgblt.ko
INSTALL drivers/video/fbdev/core/sysfillrect.ko
INSTALL fs/fuse/fuse.ko
INSTALL net/ipv6/ipv6.ko
INSTALL net/rfkill/rfkill.ko
INSTALL net/wireless/cfg80211.ko
INSTALL sound/ac97_bus.ko
INSTALL sound/core/snd-pcm.ko
INSTALL sound/core/snd-timer.ko
INSTALL sound/core/snd.ko
INSTALL sound/pci/ac97/snd-ac97-codec.ko
INSTALL sound/pci/snd-intel8x0.ko
INSTALL sound/soundcore.ko
DEPMOD 5.12.8
root@Slack:/usr/src/linux-5.12.8# cp arch/x86/boot/bzImage /boot/vmlinuz-new-5.12.8
root@Slack:/usr/src/linux-5.12.8# cp System.map /boot/System.map-new-5.12.8
root@Slack:/usr/src/linux-5.12.8# cp .config /boot/config-new-5.12.8
root@Slack:/usr/src/linux-5.12.8# cd /boot
root@Slack:/boot# ln -sf vmlinuz-new-5.12.8 vmlinuz
root@Slack:/boot# ln -sf System.map-new-5.12.8 System.map
root@Slack:/boot# ln -sf config-new-5.12.8 config
root@Slack:/boot#
```

Następnie tworzę ramdisk.

```

INSTALL drivers/input/eudev.ko
INSTALL drivers/input/mouse/psmouse.ko
INSTALL drivers/input/serio/serio_raw.ko
INSTALL drivers/net/ethernet/intel/e1000/e1000.ko
INSTALL drivers/usb/host/ehci-hcd.ko
INSTALL drivers/usb/host/ehci-pci.ko
INSTALL drivers/usb/host/ohci-hcd.ko
INSTALL drivers/usb/host/ohci-pci.ko
INSTALL drivers/video/fbdev/core/fb_sys_fops.ko
INSTALL drivers/video/fbdev/core/syscopyarea.ko
INSTALL drivers/video/fbdev/core/sysimgblt.ko
INSTALL drivers/video/fbdev/core/sysfillrect.ko
INSTALL fs/fuse/fuse.ko
INSTALL net/ipv6/ipv6.ko
INSTALL net/rfkill/rfkill.ko
INSTALL net/wireless/cfg80211.ko
INSTALL sound/ac97_bus.ko
INSTALL sound/core/snd-pcm.ko
INSTALL sound/core/snd-timer.ko
INSTALL sound/core/snd.ko
INSTALL sound/pci/ac97/snd-ac97-codec.ko
INSTALL sound/pci/snd-intel8x0.ko
INSTALL sound/soundcore.ko
DEPMOD 5.12.8
root@Slack:/usr/src/linux-5.12.8# cp arch/x86/boot/bzImage /boot/vmlinuz-new-5.12.8
root@Slack:/usr/src/linux-5.12.8# cp System.map /boot/System.map-new-5.12.8
root@Slack:/usr/src/linux-5.12.8# cp .config /boot/config-new-5.12.8
root@Slack:/usr/src/linux-5.12.8# cd /boot
root@Slack:/boot# ln -sf vmlinuz-new-5.12.8 vmlinuz
root@Slack:/boot# ln -sf System.map-new-5.12.8 System.map
root@Slack:/boot# ln -sf config-new-5.12.8 config
root@Slack:/boot# cd /usr/src/linux-5.12.8
root@Slack:/usr/src/linux-5.12.8# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.12.8
#
# mkinitrd_command_generator.sh revision 1.45
#
# This script will now make a recommendation about the command to use
# in case you require an initrd image to boot a kernel that does not
# have support for your storage or root filesystem built in
# (such as the Slackware 'generic' kernels').
# A suitable 'mkinitrd' command will be:
mkinitrd -c -k 5.12.8 -f ext4 -r /dev/sda1 -u -o /boot/initrd.gz

```

Wykonuje komendę mkinitrd.

```

root@Slack:/usr/src/linux-5.12.8# mkinitrd -c -k 5.12.8 -f ext4 -r /dev/sda1 -u -o /boot/initrd_new.gz
33371 bloków
/boot/initrd_new.gz created.
Be sure to run lilo again if you use it.

```

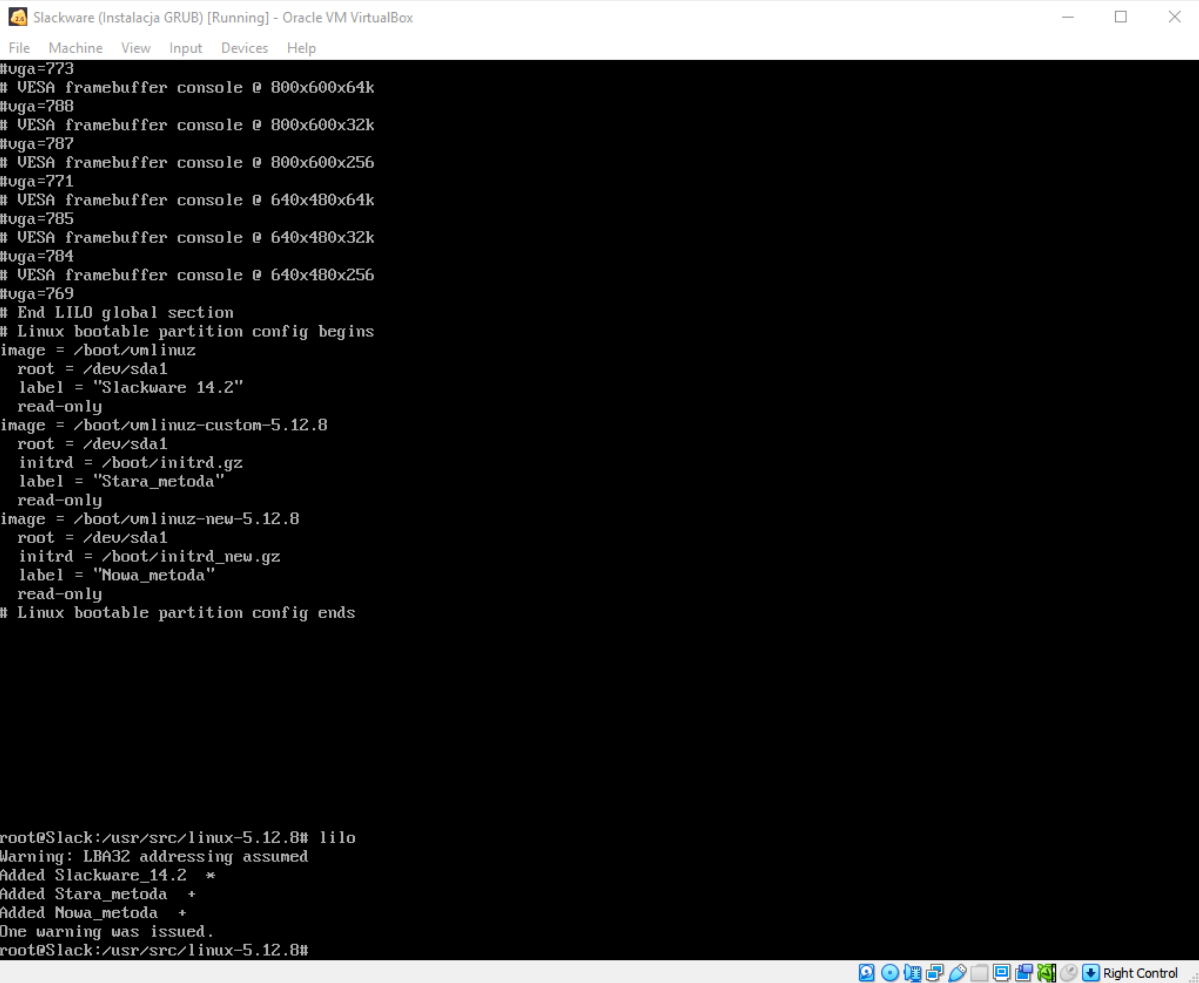
W kolejnym kroku należy dodać wpis do konfiguracji lilo w pliku /etc/lilo.conf. Wykonuje to za pomocą komendy **nano**. **nano /etc/lilo.conf**

```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
nano 2.6.0 Plik: /etc/lilo.conf

vga=791
# UESA framebuffer console @ 1024x768x32k
#vga=790
# UESA framebuffer console @ 1024x768x256
#vga=773
# UESA framebuffer console @ 800x600x64k
#vga=788
# UESA framebuffer console @ 800x600x32k
#vga=787
# UESA framebuffer console @ 800x600x256
#vga=771
# UESA framebuffer console @ 640x480x64k
#vga=785
# UESA framebuffer console @ 640x480x32k
#vga=784
# UESA framebuffer console @ 640x480x256
#vga=769
# End LILO global section
# Linux bootable partition config begins
image = /boot/vmlinuz
  root = /dev/sda1
  label = "Slackware 14.2"
  read-only
image = /boot/vmlinuz-custom-5.12.8
  root = /dev/sda1
  initrd = /boot/initrd.gz
  label = "Stara_metoda"
  read-only
image = /boot/vmlinuz-new-5.12.8
  root = /dev/sda1
  initrd = /boot/initrd_new.gz
  label = "Nowa_metoda"
  read-only
# Linux bootable partition config ends

root@Slack:/usr/src/linux-5.12.8#
```

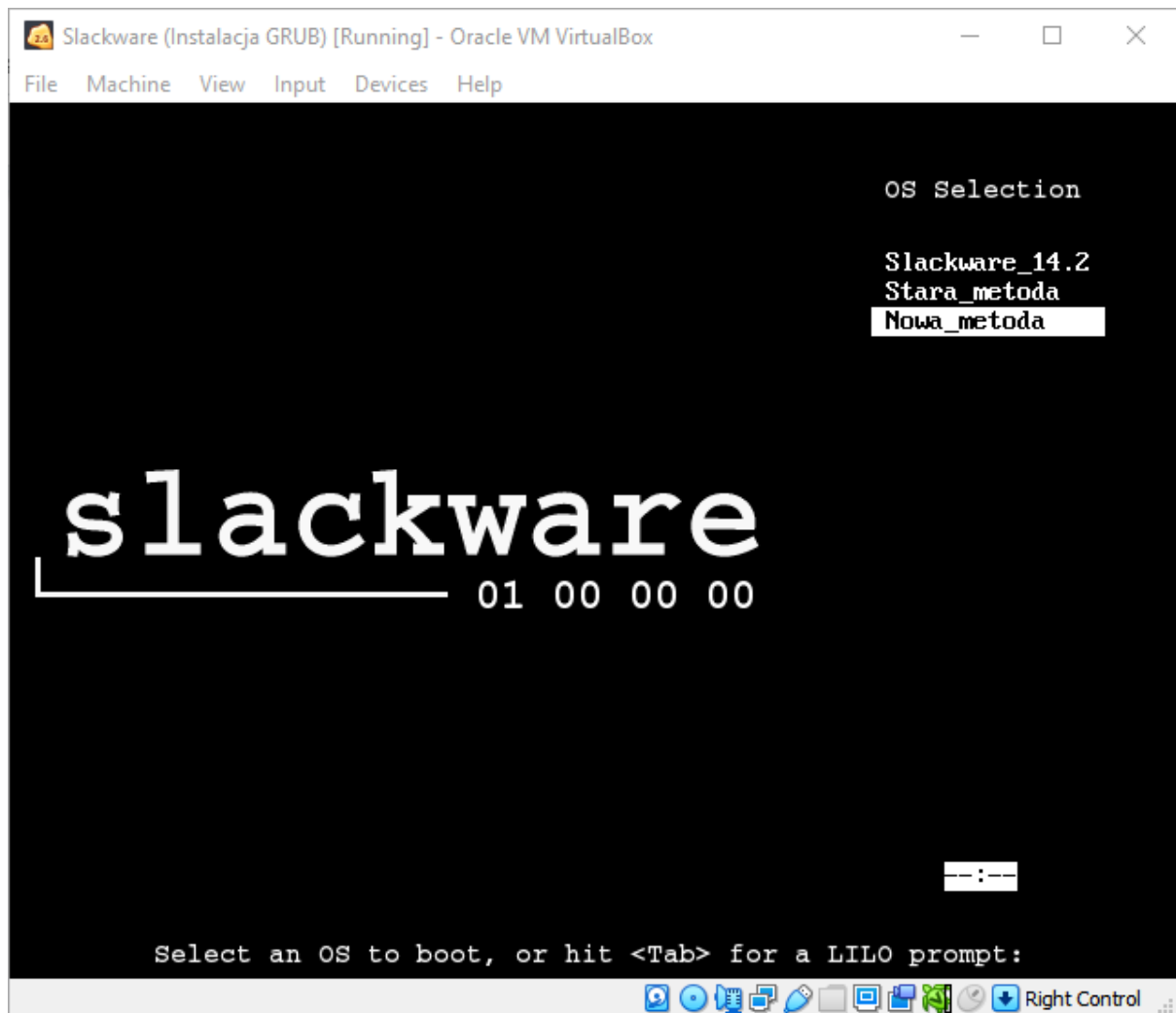
Po dodaniu wpisu w pliku konfiguracyjnego lilo wywołuje komendę **lilo**.



```
Slackware (Instalacja GRUB) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
#vga=773
# UESA framebuffer console @ 800x600x64k
#vga=788
# UESA framebuffer console @ 800x600x32k
#vga=787
# UESA framebuffer console @ 800x600x256
#vga=771
# UESA framebuffer console @ 640x480x64k
#vga=785
# UESA framebuffer console @ 640x480x32k
#vga=784
# UESA framebuffer console @ 640x480x256
#vga=769
# End LILO global section
# Linux bootable partition config begins
image = /boot/vmlinuz
  root = /dev/sda1
  label = "Slackware 14.2"
  read-only
image = /boot/vmlinuz-custom-5.12.8
  root = /dev/sda1
  initrd = /boot/initrd.gz
  label = "Stara_metoda"
  read-only
image = /boot/vmlinuz-new-5.12.8
  root = /dev/sda1
  initrd = /boot/initrd_new.gz
  label = "Nowa_metoda"
  read-only
# Linux bootable partition config ends

root@Slack:/usr/src/linux-5.12.8# lilo
Warning: LBA32 addressing assumed
Added Slackware_14.2 *
Added Stara_metoda +
Added Nowa_metoda +
One warning was issued.
root@Slack:/usr/src/linux-5.12.8#
```

Jak widać wpis nowej metody został dodany. Teraz pora na restart maszyny i sprawdzenie czy system działa.



Jak widać pojawił się nowy wpis.

4.WNIOSKI

W moim odczuciu stara metoda (localmod config) jest o wiele przyjemniejsza od metody nowszej. Jest to spowodowane błędami, które pojawiły się podczas mojej pierwszej próby kompilacji jądra nową metodą. Podczas restartowania maszyny otrzymałem kernel panic, szczerze nie potrafię dotrzeć dlaczego taki błąd występował natomiast po ponownym przejściu całego procesu udało się wszystko naprawić (być może to była jakaś literówka). Jeżeli chodzi o inne błędy, które wystąpiły podczas kompilacji w przypadku pierwszej metody sugerowałem się nazwami, które zapamiętałem z lekcji. Na końcu nazwy wersji kernela na lekcji pojawiało się -smp, dodając taką nazwę do końcówki plików dla starszej metody system nie widział zainstalowanych modułów. Dopiero po zmianie nazwy plików (usunięcia końcówki -smp) wszystko zadziało i udało mi się uruchomić system z najnowszą wersją kernela.