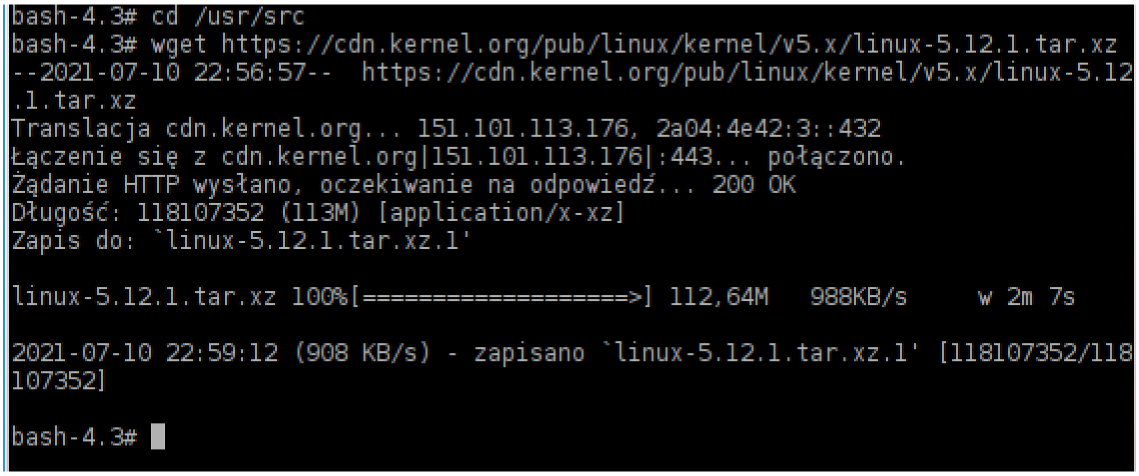
Tomasz Jąder

Kompilacja jądra linuxa

Przygotowanie do kompilacji jądra. Na początku pobrałem przeszedłem do określonego polder /usr/src i potem pobrałem wget jądro z oficjalnej strony.



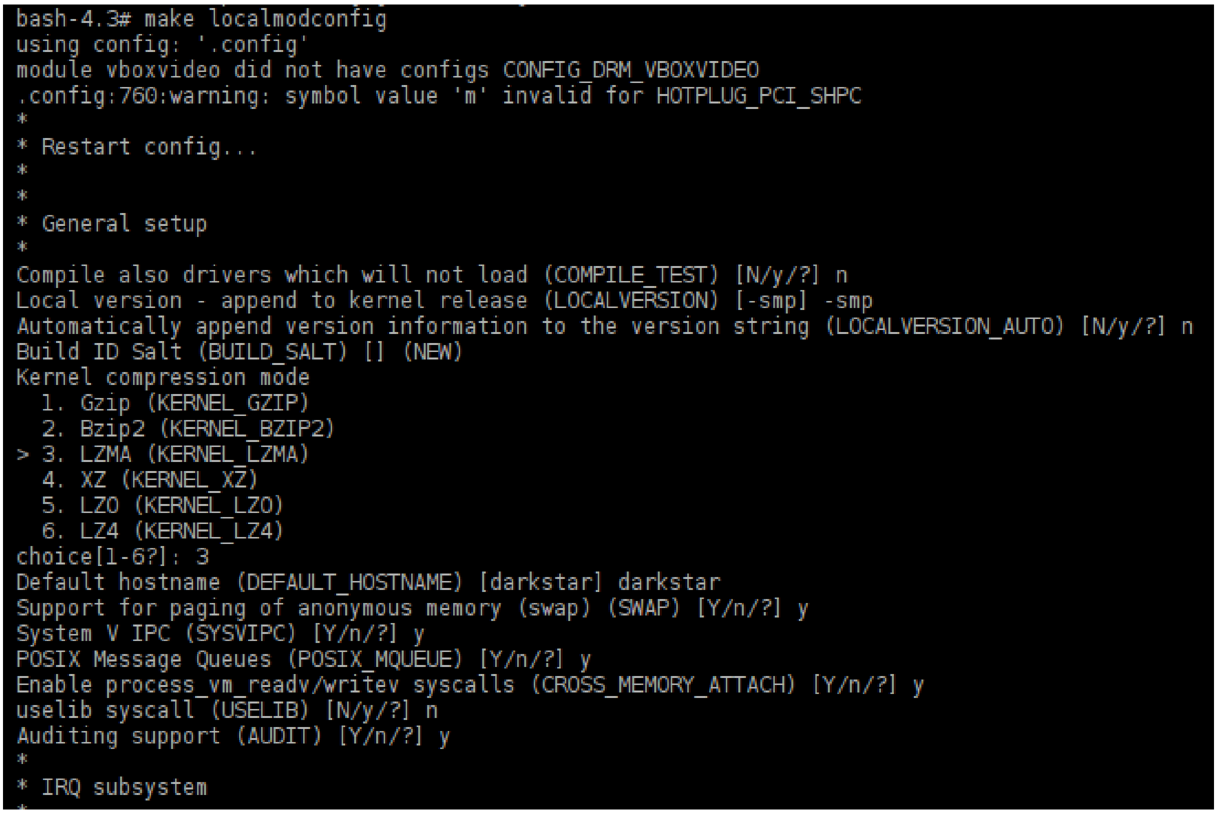
```
bash-4.3# cd /usr/src
bash-4.3# wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12.1.tar.xz
--2021-07-10 22:56:57--  https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12
.1.tar.xz
Translacja cdn.kernel.org... 151.101.113.176, 2a04:4e42:3::432
Łączenie się z cdn.kernel.org|151.101.113.176|:443... połączono.
Żądanie HTTP wysłano, oczekiwanie na odpowiedź... 200 OK
Długość: 118107352 (113M) [application/x-xz]
Zapis do: `linux-5.12.1.tar.xz.1'

linux-5.12.1.tar.xz 100%[===================>] 112,64M   988KB/s     w 2m 7s

2021-07-10 22:59:12 (908 KB/s) - zapisano `linux-5.12.1.tar.xz.1' [118107352/118
107352]

bash-4.3# 
```

Następnie rozpakowałem jądro.

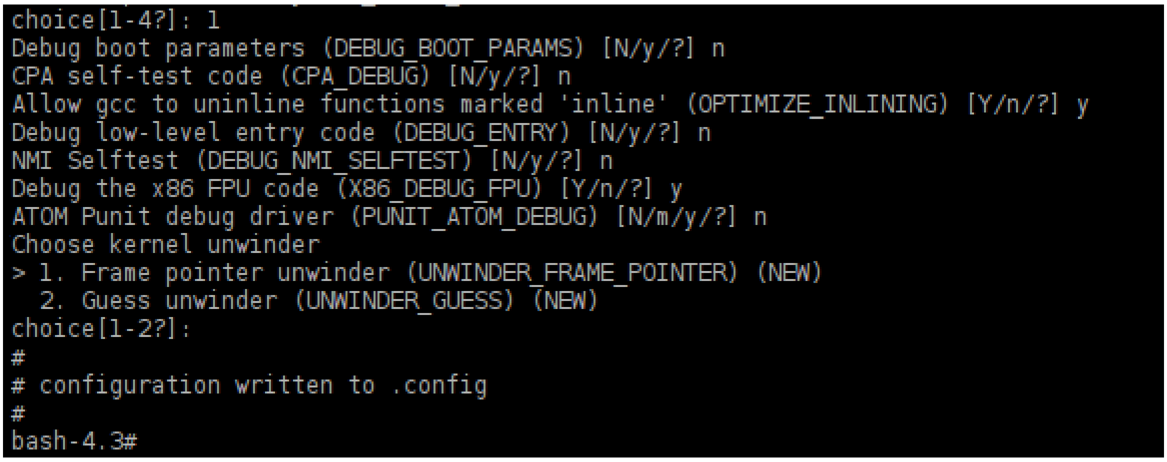
Potem przeszedłem do folderu Linux-5.12.1 i wywołanie polecenia do skopiowania starej konfiguracji.

1)Stara metoda:

Wywołuje polecenie do utworzenia konfiguracji.

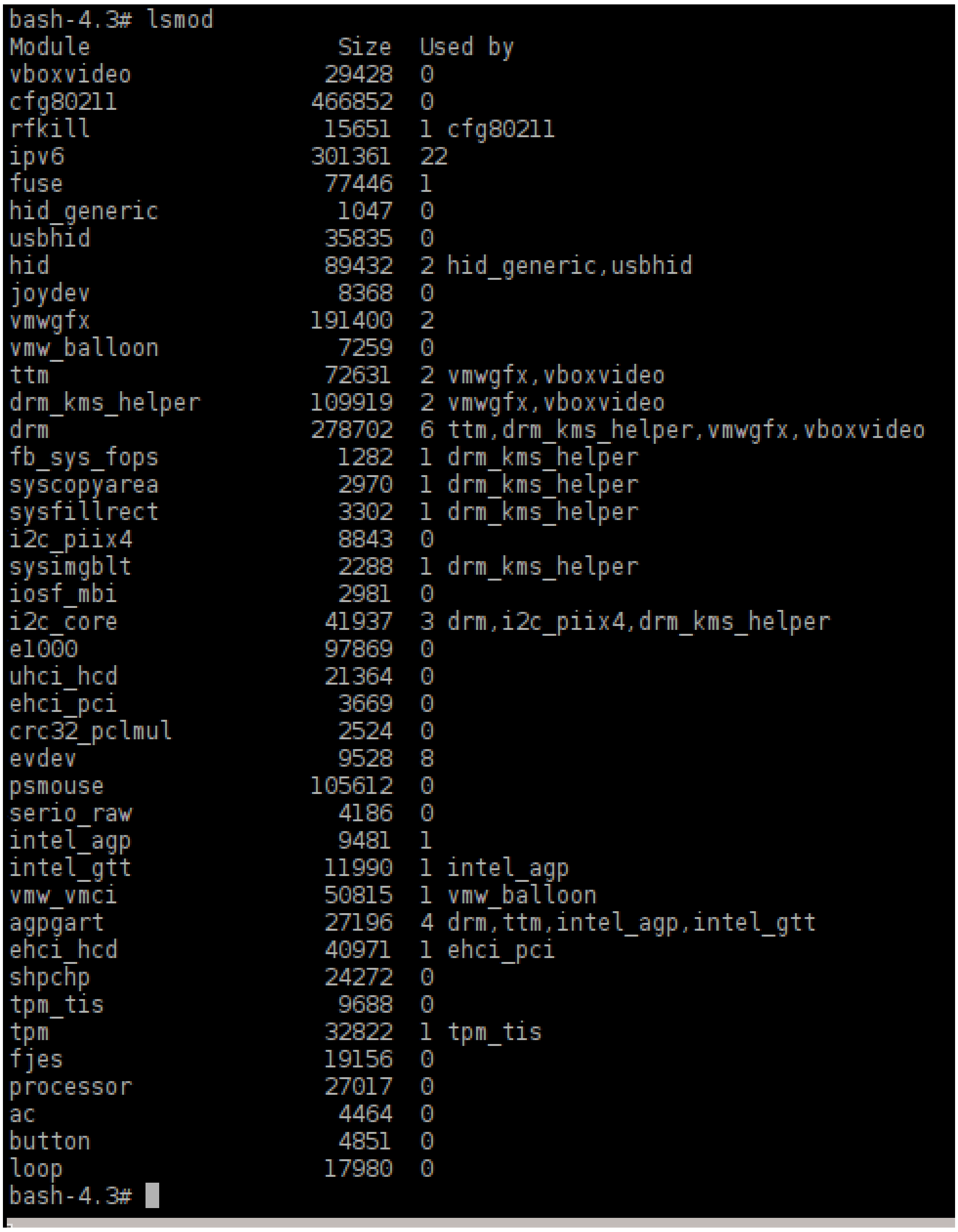


```
bash-4.3# make localmodconfig
using config: '.config'
module vboxvideo did not have configs CONFIG_DRM_VBOXVIDEO
.config:760:warning: symbol value 'm' invalid for HOTPLUG_PCI_SHPC
*
* Restart config...
*
*
* General setup
*
Compile also drivers which will not load (COMPILE_TEST) [N/y/?] n
Local version - append to kernel release (LOCALVERSION) [-smp] -smp
Automatically append version information to the version string (LOCALVERSION_AUTO) [N/y/?] n
Build ID Salt (BUILD_SALT) [] (NEW)
Kernel compression mode
  1. Gzip (KERNEL_GZIP)
  2. Bzip2 (KERNEL_BZIP2)
> 3. LZMA (KERNEL_LZMA)
  4. XZ (KERNEL_XZ)
  5. LZO (KERNEL_LZO)
  6. LZ4 (KERNEL_LZ4)
choice[1-6?]: 3
Default hostname (DEFAULT_HOSTNAME) [darkstar] darkstar
Support for paging of anonymous memory (swap) (SWAP) [Y/n/?] y
System V IPC (SYSVIPC) [Y/n/?] y
POSIX Message Queues (POSIX_MQUEUE) [Y/n/?] y
Enable process_vm_readv/writev syscalls (CROSS_MEMORY_ATTACH) [Y/n/?] y
uselib syscall (USELIB) [N/y/?] n
Auditing support (AUDIT) [Y/n/?] y
*
* IRQ subsystem
```



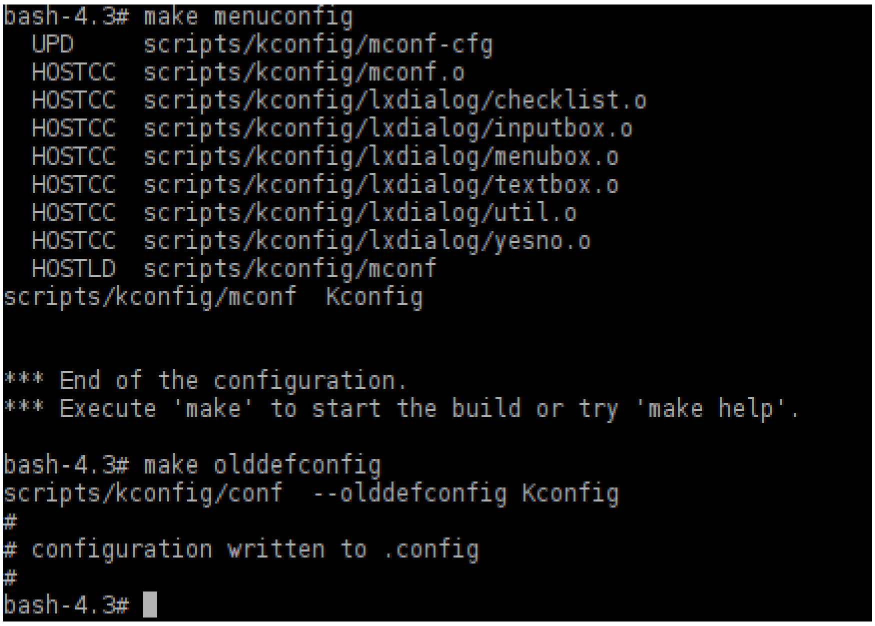
```
choice[1-4?]: 1
Debug boot parameters (DEBUG_BOOT_PARAMS) [N/y/?] n
CPA self-test code (CPA_DEBUG) [N/y/?] n
Allow gcc to uninline functions marked 'inline' (OPTIMIZE_INLINING) [Y/n/?] y
Debug low-level entry code (DEBUG_ENTRY) [N/y/?] n
NMI Selftest (DEBUG_NMI_SELFTEST) [N/y/?] n
Debug the x86 FPU code (X86_DEBUG_FPU) [Y/n/?] y
ATOM Punit debug driver (PUNIT_ATOM_DEBUG) [N/m/y/?] n
Choose kernel unwinder
> 1. Frame pointer unwinder (UNWINDER_FRAME_POINTER) (NEW)
  2. Guess unwinder (UNWINDER_GUESS) (NEW)
choice[1-2?]:
#
# configuration written to .config
#
bash-4.3#
```

Następnie wywołałem polecenie do sprawdzenia aktualnych modułów.



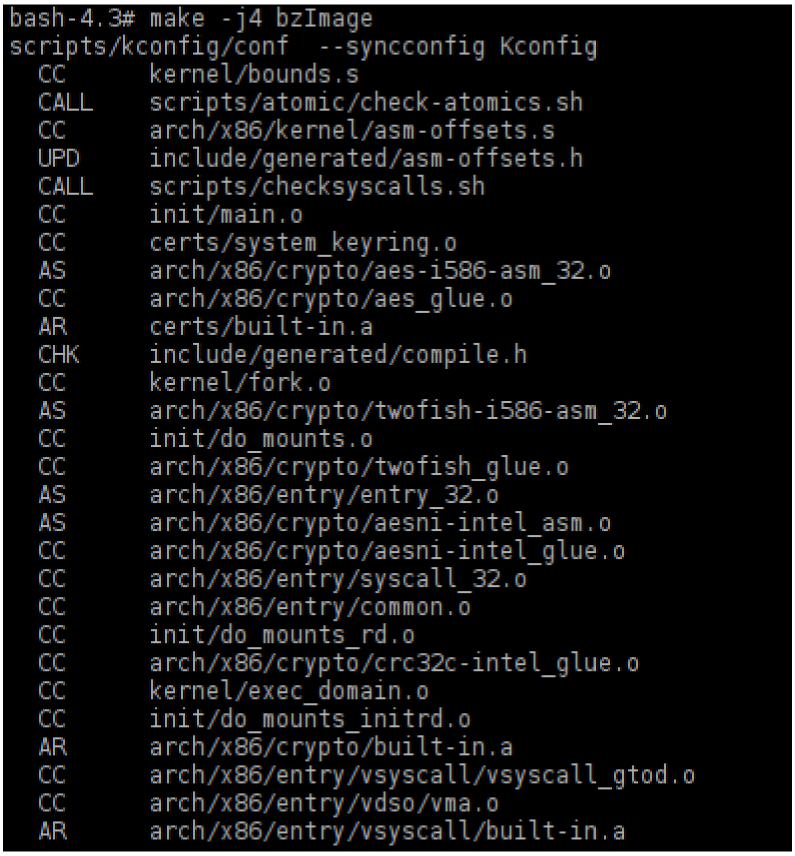
```
bash-4.3# lsmod
Module                  Size  Used by
vboxvideo              29428  0
cfg80211              466852  0
rfkill                 15651  1 cfg80211
ipv6                  301361  22
fuse                   77446  1
hid_generic             1047  0
usbhid                 35835  0
hid                    89432  2 hid_generic,usbhid
joydev                  8368  0
vmwgfx                191400  2
vmw_balloon             7259  0
ttm                    72631  2 vmwgfx,vboxvideo
drm_kms_helper        109919  2 vmwgfx,vboxvideo
drm                   278702  6 ttm,drm_kms_helper,vmwgfx,vboxvideo
fb_sys_fops             1282  1 drm_kms_helper
syscopyarea             2970  1 drm_kms_helper
sysfillrect             3302  1 drm_kms_helper
i2c_piix4               8843  0
sysimgblt               2288  1 drm_kms_helper
iosf_mbi                2981  0
i2c_core               41937  3 drm,i2c_piix4,drm_kms_helper
e1000                  97869  0
uhci_hcd               21364  0
ehci_pci                3669  0
crc32_pclmul            2524  0
evdev                   9528  8
psmouse               105612  0
serio_raw               4186  0
intel_agp               9481  1
intel_gtt              11990  1 intel_agp
vmw_vmci               50815  1 vmw_balloon
agpgart                27196  4 drm,ttm,intel_agp,intel_gtt
ehci_hcd               40971  1 ehci_pci
shpchp                 24272  0
tpm_tis                 9688  0
tpm                    32822  1 tpm_tis
fjes                   19156  0
processor              27017  0
ac                      4464  0
button                  4851  0
loop                   17980  0
bash-4.3#
```

Potem sprawdzam konfiguracje kernela.

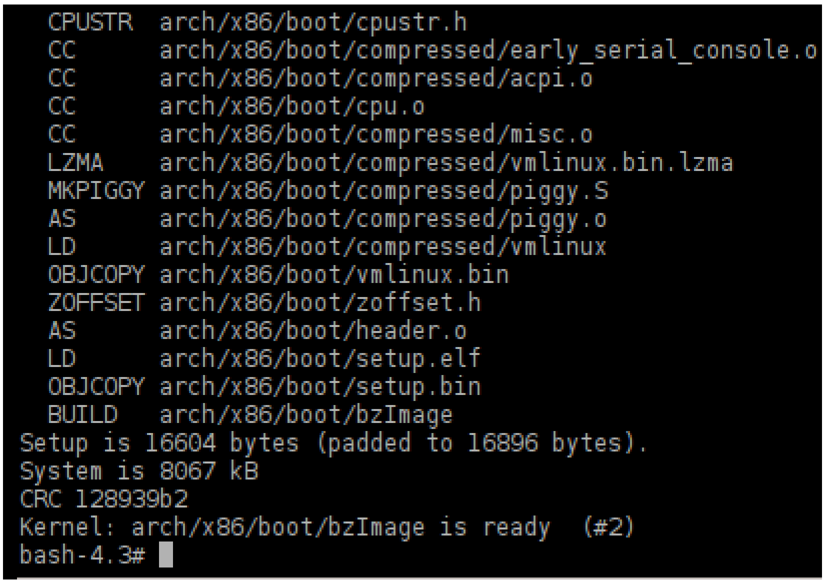


```
bash-4.3# make menuconfig
  UPD     scripts/kconfig/mconf-cfg
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

bash-4.3# make olddefconfig
scripts/kconfig/conf  --olddefconfig Kconfig
#
# configuration written to .config
#
bash-4.3# 
```

Przechodzę do kompilacji jądra.

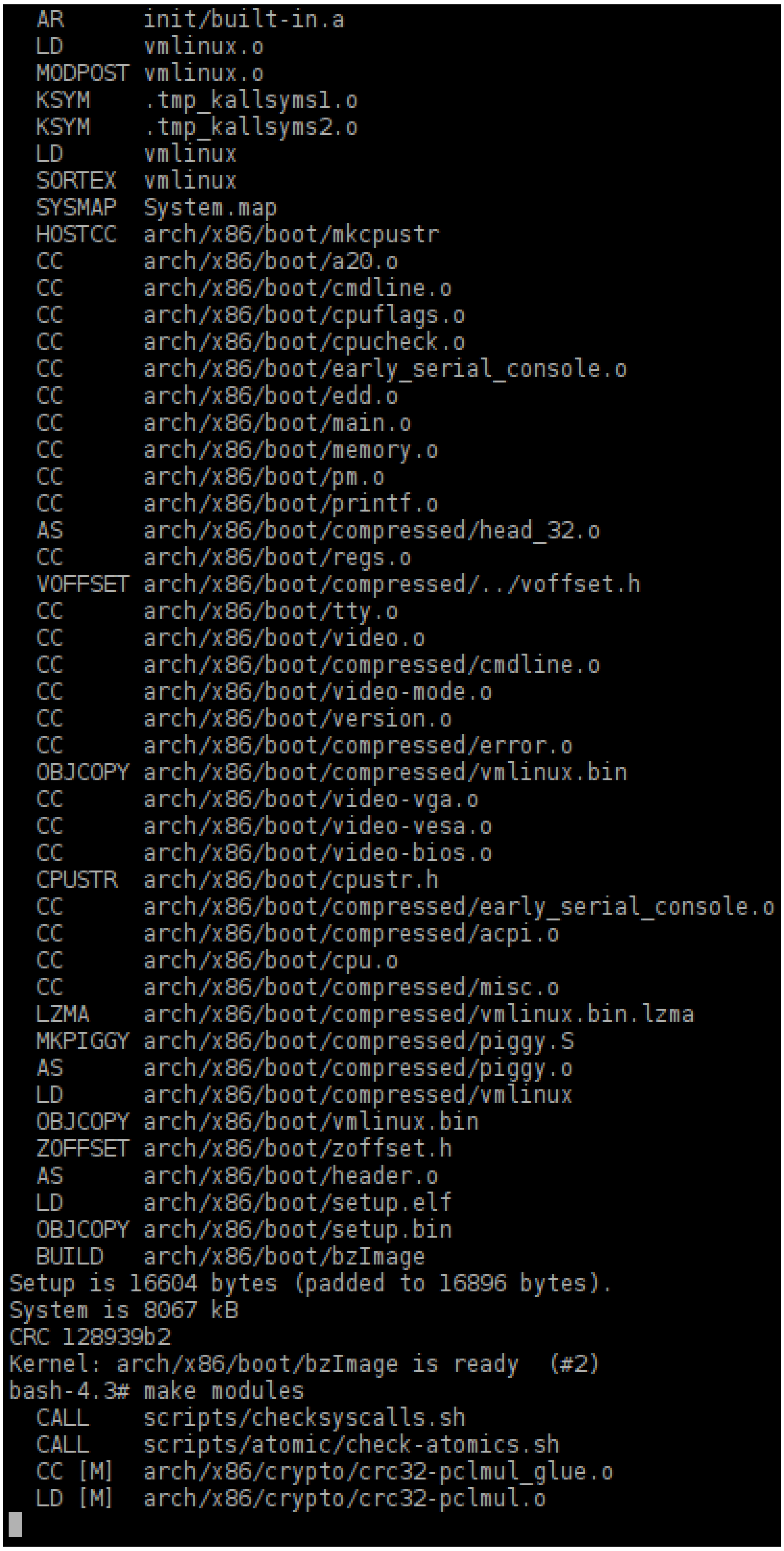


```
bash-4.3# make -j4 bzImage
scripts/kconfig/conf  --syncconfig Kconfig
  CC      kernel/bounds.s
  CALL    scripts/atomic/check-atomics.sh
  CC      arch/x86/kernel/asm-offsets.s
  UPD     include/generated/asm-offsets.h
  CALL    scripts/checksyscalls.sh
  CC      init/main.o
  CC      certs/system_keyring.o
  AS      arch/x86/crypto/aes-i586-asm_32.o
  CC      arch/x86/crypto/aes_glue.o
  AR      certs/built-in.a
  CHK     include/generated/compile.h
  CC      kernel/fork.o
  AS      arch/x86/crypto/twofish-i586-asm_32.o
  CC      init/do_mounts.o
  CC      arch/x86/crypto/twofish_glue.o
  AS      arch/x86/entry/entry_32.o
  AS      arch/x86/crypto/aesni-intel_asm.o
  CC      arch/x86/crypto/aesni-intel_glue.o
  CC      arch/x86/entry/syscall_32.o
  CC      arch/x86/entry/common.o
  CC      init/do_mounts_rd.o
  CC      arch/x86/crypto/crc32c-intel_glue.o
  CC      kernel/exec_domain.o
  CC      init/do_mounts_initrd.o
  AR      arch/x86/crypto/built-in.a
  CC      arch/x86/entry/vsyscall/vsyscall_gtod.o
  CC      arch/x86/entry/vdso/vma.o
  AR      arch/x86/entry/vsyscall/built-in.a
```

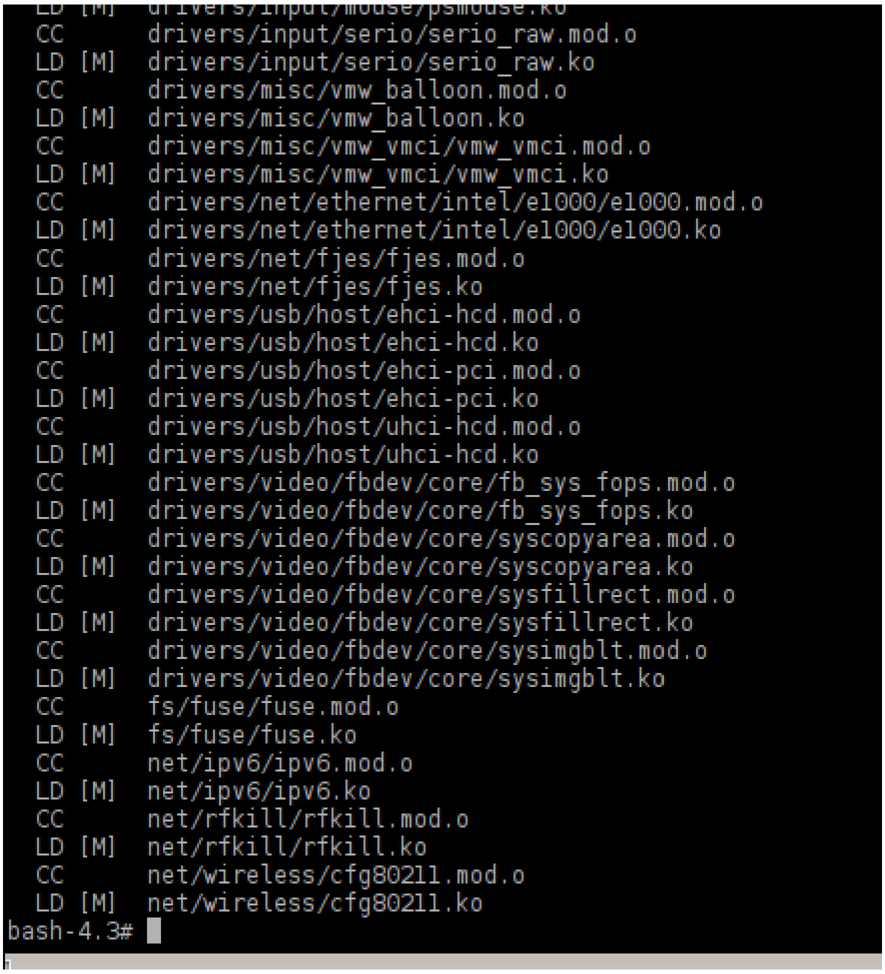


```
  CPUSTR  arch/x86/boot/cpustr.h
  CC      arch/x86/boot/compressed/early_serial_console.o
  CC      arch/x86/boot/compressed/acpi.o
  CC      arch/x86/boot/cpu.o
  CC      arch/x86/boot/compressed/misc.o
  LZMA    arch/x86/boot/compressed/vmlinux.bin.lzma
  MKPIGGY arch/x86/boot/compressed/piggy.S
  AS      arch/x86/boot/compressed/piggy.o
  LD      arch/x86/boot/compressed/vmlinux
  OBJCOPY arch/x86/boot/vmlinux.bin
  ZOFFSET arch/x86/boot/zoffset.h
  AS      arch/x86/boot/header.o
  LD      arch/x86/boot/setup.elf
  OBJCOPY arch/x86/boot/setup.bin
  BUILD   arch/x86/boot/bzImage
Setup is 16604 bytes (padded to 16896 bytes).
System is 8067 kB
CRC 128939b2
Kernel: arch/x86/boot/bzImage is ready  (#2)
bash-4.3#
```

Po wszystkim używam polecania make modules do zbudowania modułów.

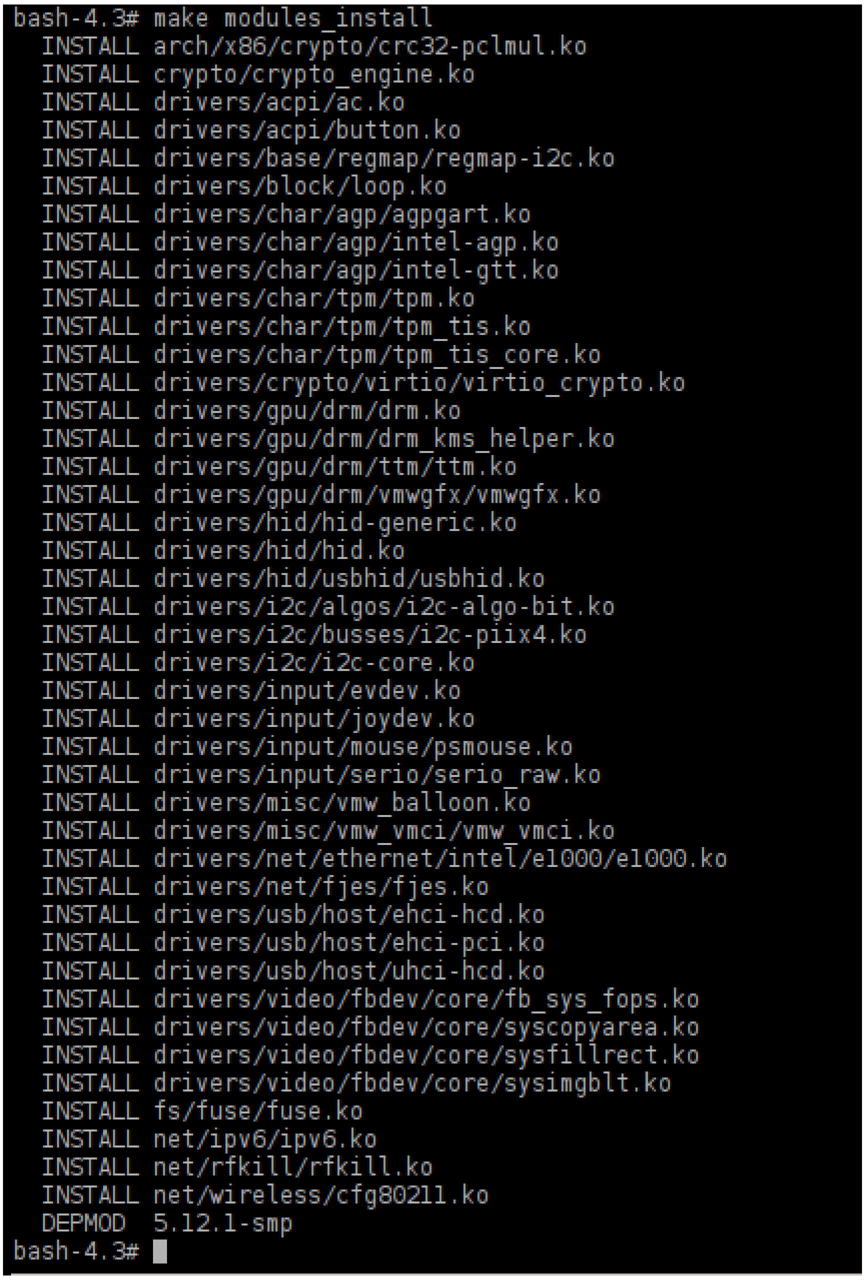


```
  AR      init/built-in.a
  LD      vmlinux.o
  MODPOST vmlinux.o
  KSYM    .tmp_kallsyms1.o
  KSYM    .tmp_kallsyms2.o
  LD      vmlinux
  SORTEX  vmlinux
  SYSMAP  System.map
  HOSTCC  arch/x86/boot/mkcpustr
  CC      arch/x86/boot/a20.o
  CC      arch/x86/boot/cmdline.o
  CC      arch/x86/boot/cpuflags.o
  CC      arch/x86/boot/cpucheck.o
  CC      arch/x86/boot/early_serial_console.o
  CC      arch/x86/boot/edd.o
  CC      arch/x86/boot/main.o
  CC      arch/x86/boot/memory.o
  CC      arch/x86/boot/pm.o
  CC      arch/x86/boot/printf.o
  AS      arch/x86/boot/compressed/head_32.o
  CC      arch/x86/boot/regs.o
  VOFFSET arch/x86/boot/compressed/../voffset.h
  CC      arch/x86/boot/tty.o
  CC      arch/x86/boot/video.o
  CC      arch/x86/boot/compressed/cmdline.o
  CC      arch/x86/boot/video-mode.o
  CC      arch/x86/boot/version.o
  CC      arch/x86/boot/compressed/error.o
  OBJCOPY arch/x86/boot/compressed/vmlinux.bin
  CC      arch/x86/boot/video-vga.o
  CC      arch/x86/boot/video-vesa.o
  CC      arch/x86/boot/video-bios.o
  CPUSTR  arch/x86/boot/cpustr.h
  CC      arch/x86/boot/compressed/early_serial_console.o
  CC      arch/x86/boot/compressed/acpi.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
  OBJCOPY arch/x86/boot/vmlinux.bin
  ZOFFSET arch/x86/boot/zoffset.h
  AS      arch/x86/boot/header.o
  LD      arch/x86/boot/setup.elf
  OBJCOPY arch/x86/boot/setup.bin
  BUILD   arch/x86/boot/bzImage
Setup is 16604 bytes (padded to 16896 bytes).
System is 8067 kB
CRC 128939b2
Kernel: arch/x86/boot/bzImage is ready  (#2)
bash-4.3# make modules
  CALL    scripts/checksyscalls.sh
  CALL    scripts/atomic/check-atomics.sh
  CC [M]  arch/x86/crypto/crc32-pclmul_glue.o
  LD [M]  arch/x86/crypto/crc32-pclmul.o
```



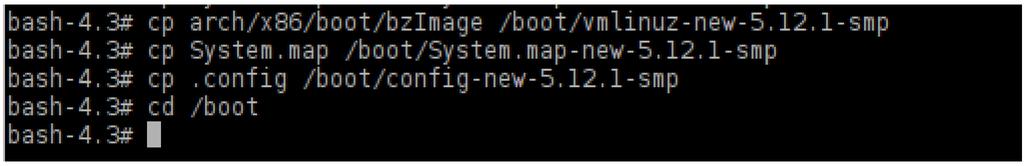
```
  LD [M]  drivers/input/mouse/psmouse.ko
  CC      drivers/input/serio/serio_raw.mod.o
  LD [M]  drivers/input/serio/serio_raw.ko
  CC      drivers/misc/vmw_balloon.mod.o
  LD [M]  drivers/misc/vmw_balloon.ko
  CC      drivers/misc/vmw_vmci/vmw_vmci.mod.o
  LD [M]  drivers/misc/vmw_vmci/vmw_vmci.ko
  CC      drivers/net/ethernet/intel/e1000/e1000.mod.o
  LD [M]  drivers/net/ethernet/intel/e1000/e1000.ko
  CC      drivers/net/fjes/fjes.mod.o
  LD [M]  drivers/net/fjes/fjes.ko
  CC      drivers/usb/host/ehci-hcd.mod.o
  LD [M]  drivers/usb/host/ehci-hcd.ko
  CC      drivers/usb/host/ehci-pci.mod.o
  LD [M]  drivers/usb/host/ehci-pci.ko
  CC      drivers/usb/host/uhci-hcd.mod.o
  LD [M]  drivers/usb/host/uhci-hcd.ko
  CC      drivers/video/fbdev/core/fb_sys_fops.mod.o
  LD [M]  drivers/video/fbdev/core/fb_sys_fops.ko
  CC      drivers/video/fbdev/core/syscopyarea.mod.o
  LD [M]  drivers/video/fbdev/core/syscopyarea.ko
  CC      drivers/video/fbdev/core/sysfillrect.mod.o
  LD [M]  drivers/video/fbdev/core/sysfillrect.ko
  CC      drivers/video/fbdev/core/sysimgblt.mod.o
  LD [M]  drivers/video/fbdev/core/sysimgblt.ko
  CC      fs/fuse/fuse.mod.o
  LD [M]  fs/fuse/fuse.ko
  CC      net/ipv6/ipv6.mod.o
  LD [M]  net/ipv6/ipv6.ko
  CC      net/rfkill/rfkill.mod.o
  LD [M]  net/rfkill/rfkill.ko
  CC      net/wireless/cfg80211.mod.o
  LD [M]  net/wireless/cfg80211.ko
bash-4.3# 
```

Po tym poleceniu zabieram się do instalacji modułów



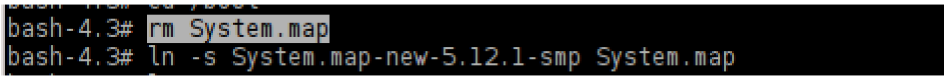
```
bash-4.3# make modules_install
  INSTALL arch/x86/crypto/crc32-pclmul.ko
  INSTALL crypto/crypto_engine.ko
  INSTALL drivers/acpi/ac.ko
  INSTALL drivers/acpi/button.ko
  INSTALL drivers/base/regmap/regmap-i2c.ko
  INSTALL drivers/block/loop.ko
  INSTALL drivers/char/agp/agpgart.ko
  INSTALL drivers/char/agp/intel-agp.ko
  INSTALL drivers/char/agp/intel-gtt.ko
  INSTALL drivers/char/tpm/tpm.ko
  INSTALL drivers/char/tpm/tpm_tis.ko
  INSTALL drivers/char/tpm/tpm_tis_core.ko
  INSTALL drivers/crypto/virtio/virtio_crypto.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/hid/hid-generic.ko
  INSTALL drivers/hid/hid.ko
  INSTALL drivers/hid/usbhid/usbhid.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/joydev.ko
  INSTALL drivers/input/mouse/psmouse.ko
  INSTALL drivers/input/serio/serio_raw.ko
  INSTALL drivers/misc/vmw_balloon.ko
  INSTALL drivers/misc/vmw_vmci/vmw_vmci.ko
  INSTALL drivers/net/ethernet/intel/e1000/e1000.ko
  INSTALL drivers/net/fjes/fjes.ko
  INSTALL drivers/usb/host/ehci-hcd.ko
  INSTALL drivers/usb/host/ehci-pci.ko
  INSTALL drivers/usb/host/uhci-hcd.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 net/ipv6/ipv6.ko
  INSTALL net/rfkill/rfkill.ko
  INSTALL net/wireless/cfg80211.ko
  DEPMOD  5.12.1-smp
bash-4.3# █
```

Przekopiowanie plików jądra do sytemu



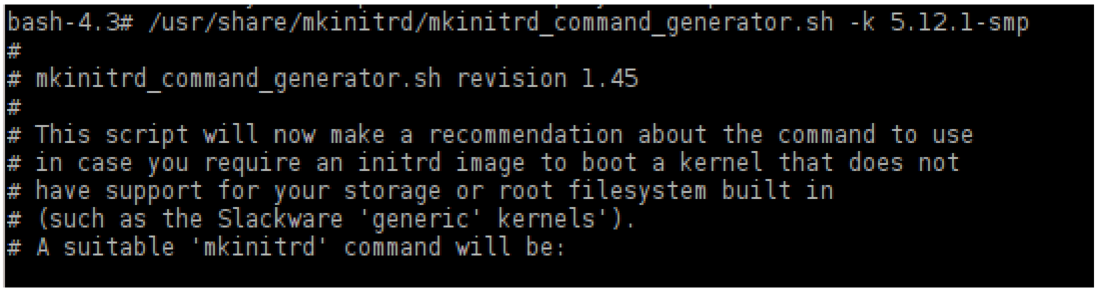
```
bash-4.3# cp arch/x86/boot/bzImage /boot/vmlinuz-new-5.12.1-smp
bash-4.3# cp System.map /boot/System.map-new-5.12.1-smp
bash-4.3# cp .config /boot/config-new-5.12.1-smp
bash-4.3# cd /boot
bash-4.3# █
```

Tworze link symboliczny do tablicy kerneli.

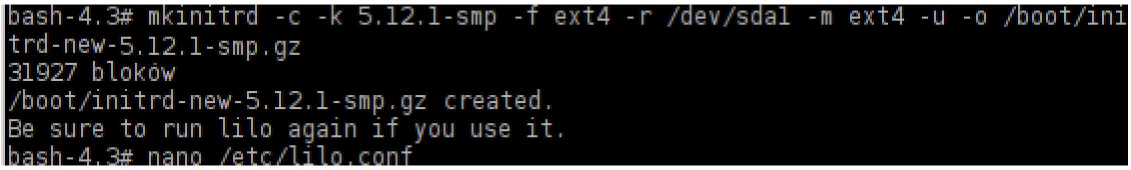


```
bash-4.3# rm System.map
bash-4.3# ln -s System.map-new-5.12.1-smp System.map
```

Po wszystkim wywołuje następujące polecenia które pozwolą przygotować dysk ram.

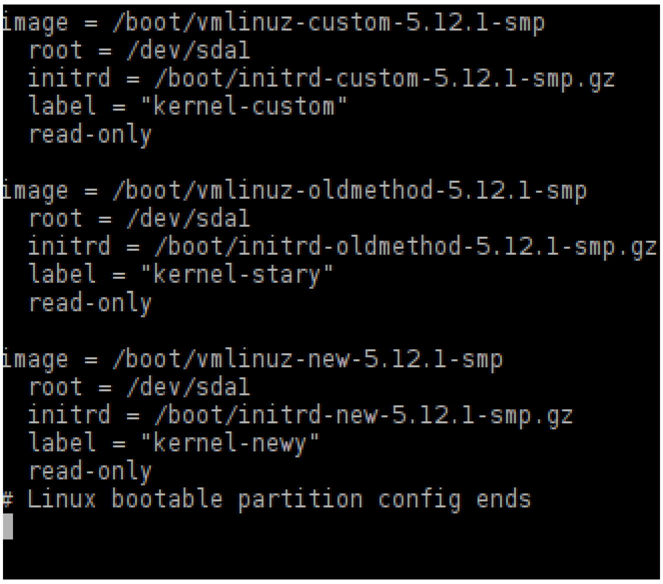


```
bash-4.3# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.12.1-smp
#
# mkinitrd_command_generator.sh revision 1.45
#
# This script will now make a recommendation about the command to use
# in case you require an initrd image to boot a kernel that does not
# have support for your storage or root filesystem built in
# (such as the Slackware 'generic' kernels').
# A suitable 'mkinitrd' command will be:
```



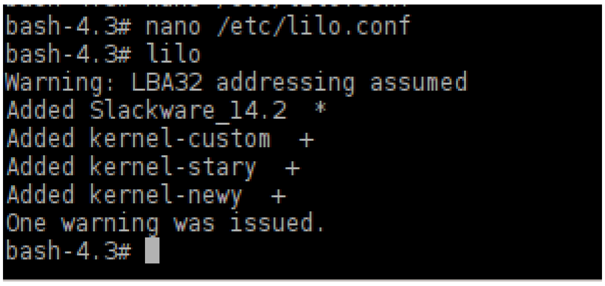
```
bash-4.3# mkinitrd -c -k 5.12.1-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/ini
trd-new-5.12.1-smp.gz
31927 bloków
/boot/initrd-new-5.12.1-smp.gz created.
Be sure to run lilo again if you use it.
bash-4.3# nano /etc/lilo.conf
```

Dodaje nowe wpisy do konfiguracji bootloadera

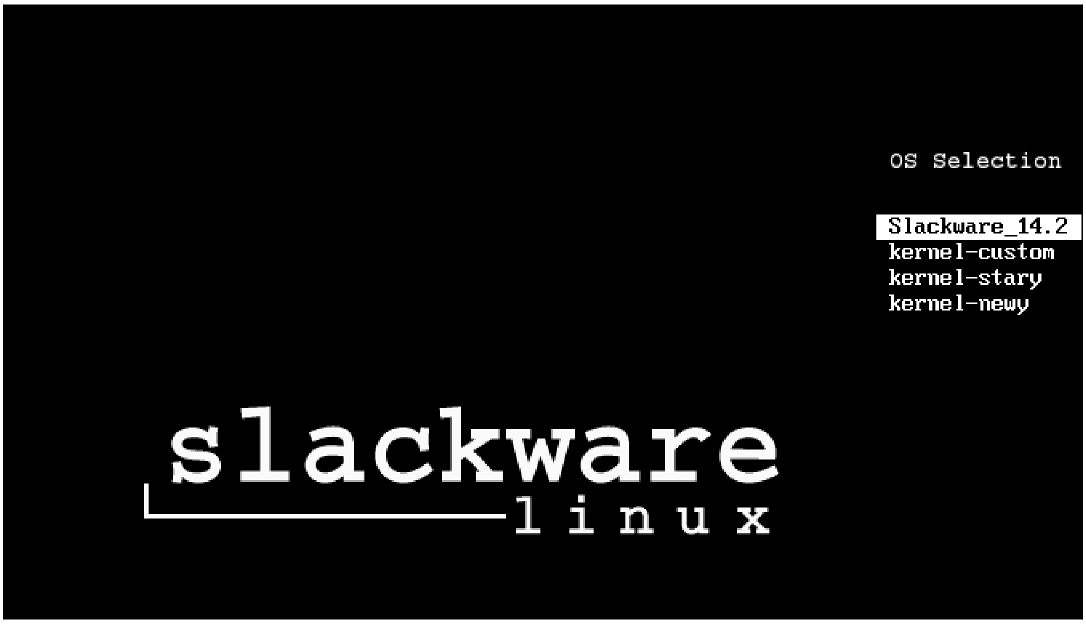


```
image = /boot/vmlinuz-custom-5.12.1-smp
  root = /dev/sda1
  initrd = /boot/initrd-custom-5.12.1-smp.gz
  label = "kernel-custom"
  read-only

image = /boot/vmlinuz-oldmethod-5.12.1-smp
  root = /dev/sda1
  initrd = /boot/initrd-oldmethod-5.12.1-smp.gz
  label = "kernel-stary"
  read-only

image = /boot/vmlinuz-new-5.12.1-smp
  root = /dev/sda1
  initrd = /boot/initrd-new-5.12.1-smp.gz
  label = "kernel-newy"
  read-only
# Linux bootable partition config ends
```

Wywołuje komendę lilo



```
bash-4.3# nano /etc/lilo.conf
bash-4.3# lilo
Warning: LBA32 addressing assumed
Added Slackware_14.2  *
Added kernel-custom  +
Added kernel-stary  +
Added kernel-newy  +
One warning was issued.
bash-4.3# 
```

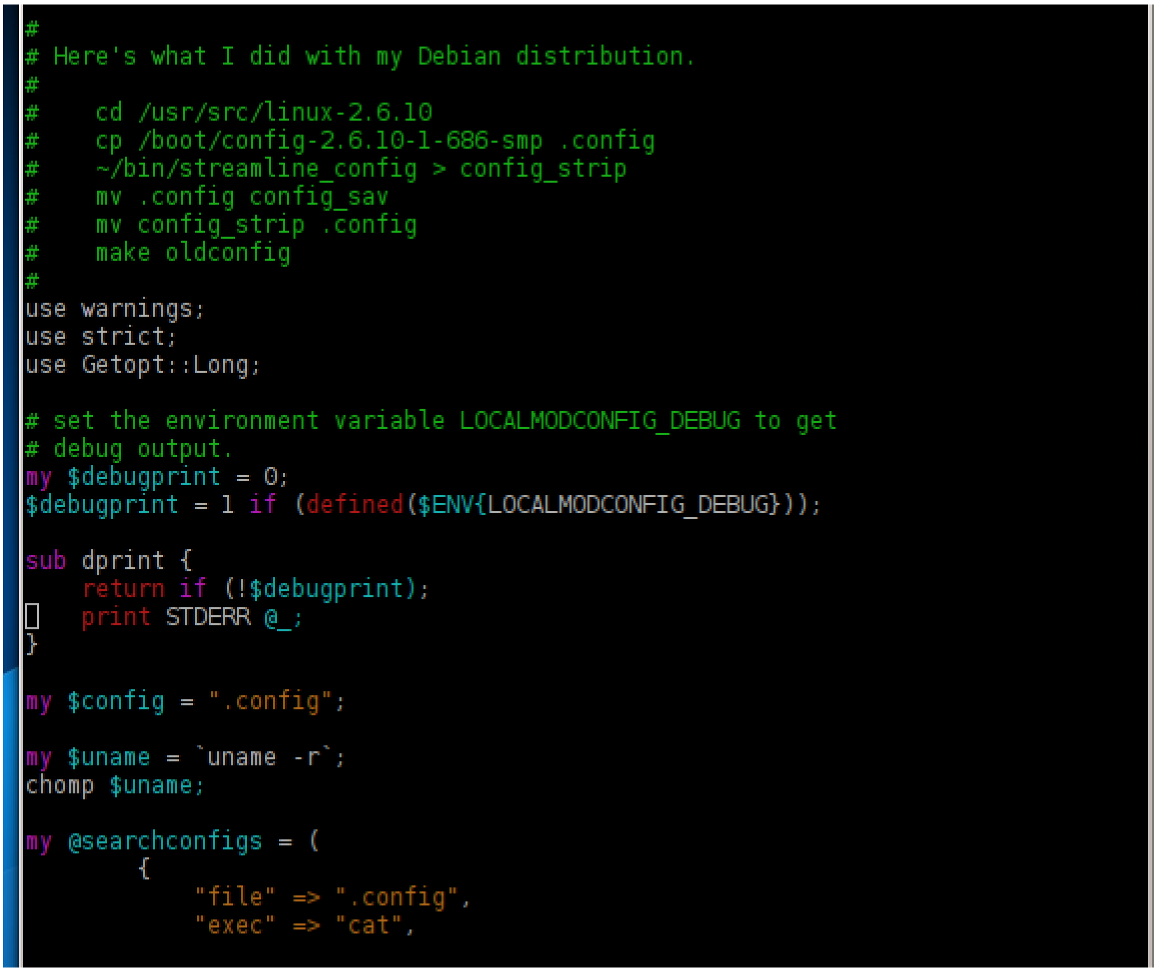


2) Nowa metoda:

Poranie i ropadkownie nastepuje tak samo zmian zaczyna się dopiero po wywoła niu polecienia

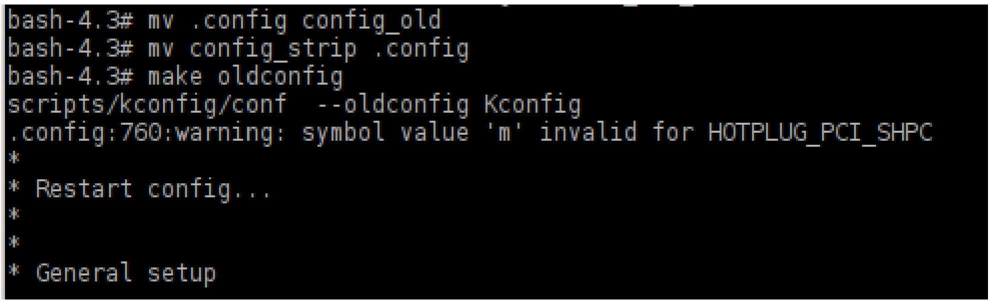
zcat /pro/confi.gz > .config

Na początku otwieram skrypt scripts/kconfig/streamline_config.pl

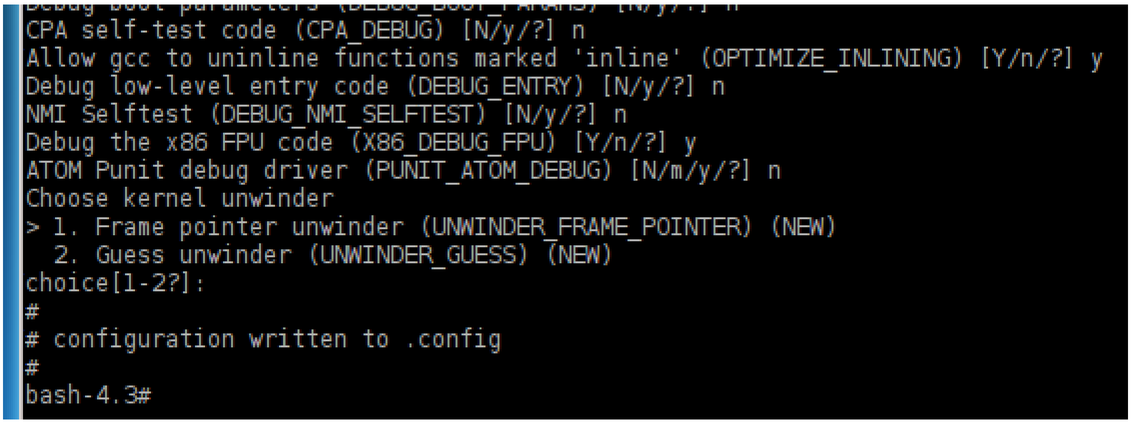


```
#
# Here's what I did with my Debian distribution.
#
#    cd /usr/src/linux-2.6.10
#    cp /boot/config-2.6.10-1-686-smp .config
#    ~/bin/streamline_config > config_strip
#    mv .config config_sav
#    mv config_strip .config
#    make oldconfig
#
use warnings;
use strict;
use Getopt::Long;

# set the environment variable LOCALMODCONFIG_DEBUG to get
# debug output.
my $debugprint = 0;
$debugprint = 1 if (defined($ENV{LOCALMODCONFIG_DEBUG}));

sub dprint {
    return if (!$debugprint);
□   print STDERR @_;
}

my $config = ".config";

my $uname = `uname -r`;
chomp $uname;

my @searchconfigs = (
        {
            "file" => ".config",
            "exec" => "cat",
```

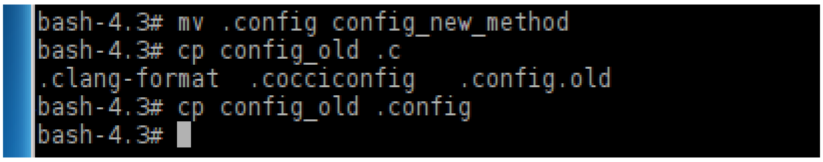
Następnie postępuje według jego instrukcji i wywołuje kluczowe polecenie make oldconfig.



```
bash-4.3# mv .config config_old
bash-4.3# mv config_strip .config
bash-4.3# make oldconfig
scripts/kconfig/conf  --oldconfig Kconfig
.config:760:warning: symbol value 'm' invalid for HOTPLUG_PCI_SHPC
*
* Restart config...
*
*
* General setup
```

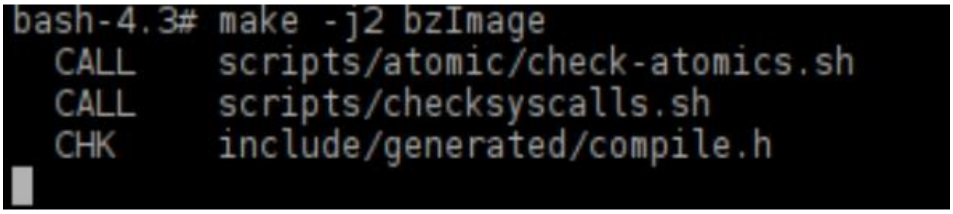


```
Debug boot parameters (DEBUG_BOOT_PARAMS) [N/y/?] n
CPA self-test code (CPA_DEBUG) [N/y/?] n
Allow gcc to uninline functions marked 'inline' (OPTIMIZE_INLINING) [Y/n/?] y
Debug low-level entry code (DEBUG_ENTRY) [N/y/?] n
NMI Selftest (DEBUG_NMI_SELFTEST) [N/y/?] n
Debug the x86 FPU code (X86_DEBUG_FPU) [Y/n/?] y
ATOM Punit debug driver (PUNIT_ATOM_DEBUG) [N/m/y/?] n
Choose kernel unwinder
> 1. Frame pointer unwinder (UNWINDER_FRAME_POINTER) (NEW)
  2. Guess unwinder (UNWINDER_GUESS) (NEW)
choice[1-2?]:
#
# configuration written to .config
#
bash-4.3#
```

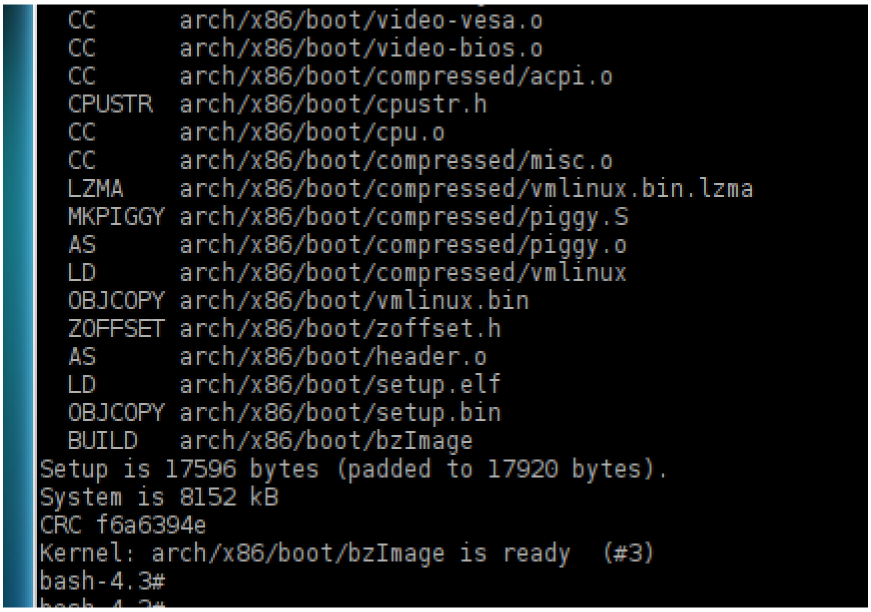


```
bash-4.3# mv .config config_new_method
bash-4.3# cp config_old .c
.clang-format  .cocciconfig   .config.old
bash-4.3# cp config_old .config
bash-4.3# 
```

Potem przechodzę do kompilacji obrazu jądra

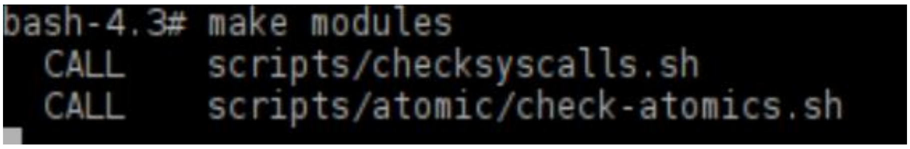


```
bash-4.3# make -j2 bzImage
  CALL    scripts/atomic/check-atomics.sh
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
```

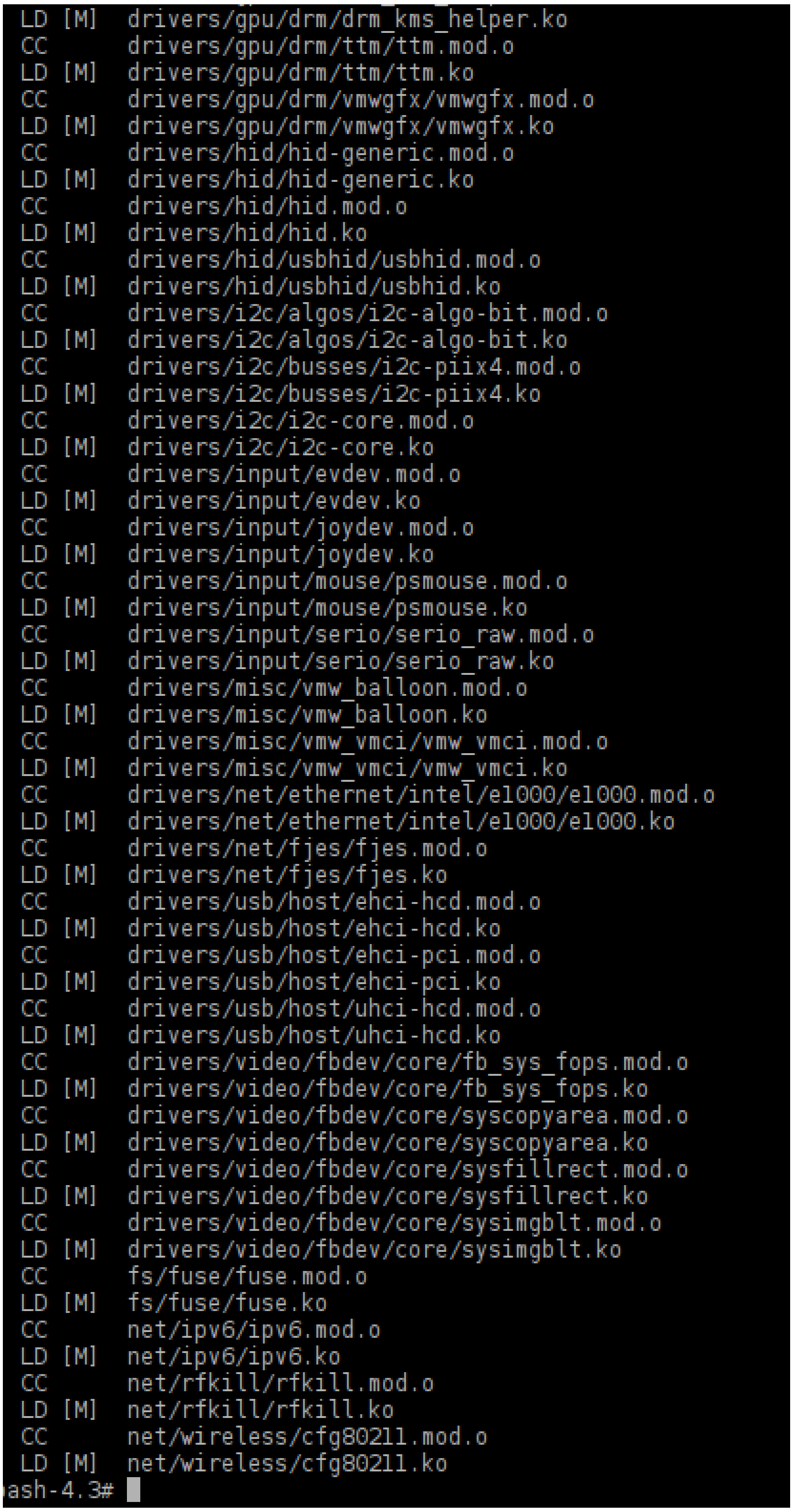


```
  CC      arch/x86/boot/video-vesa.o
  CC      arch/x86/boot/video-bios.o
  CC      arch/x86/boot/compressed/acpi.o
  CPUSTR  arch/x86/boot/cpustr.h
  CC      arch/x86/boot/cpu.o
  CC      arch/x86/boot/compressed/misc.o
  LZMA    arch/x86/boot/compressed/vmlinux.bin.lzma
  MKPIGGY arch/x86/boot/compressed/piggy.S
  AS      arch/x86/boot/compressed/piggy.o
  LD      arch/x86/boot/compressed/vmlinux
  OBJCOPY arch/x86/boot/vmlinux.bin
  ZOFFSET arch/x86/boot/zoffset.h
  AS      arch/x86/boot/header.o
  LD      arch/x86/boot/setup.elf
  OBJCOPY arch/x86/boot/setup.bin
  BUILD   arch/x86/boot/bzImage
Setup is 17596 bytes (padded to 17920 bytes).
System is 8152 kB
CRC f6a6394e
Kernel: arch/x86/boot/bzImage is ready  (#3)
bash-4.3#
```

Buduje moduły niezbędne do działania.

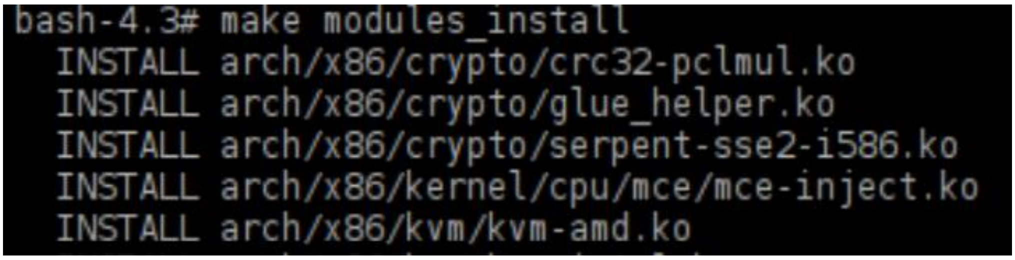


```
bash-4.3# make modules
  CALL    scripts/checksyscalls.sh
  CALL    scripts/atomic/check-atomics.sh
```

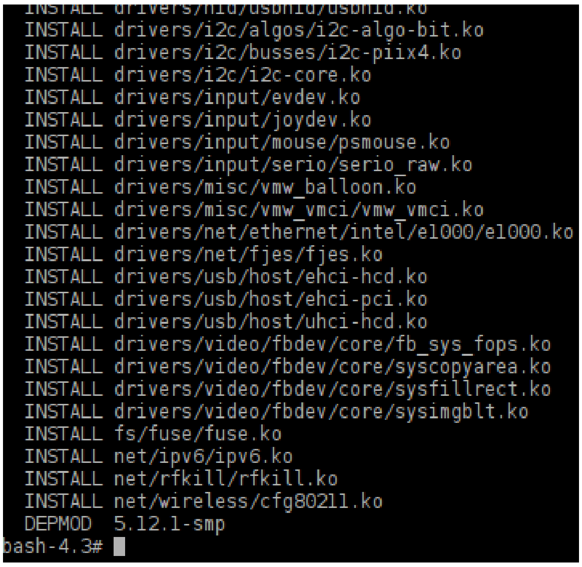


```
  LD [M]  drivers/gpu/drm/drm_kms_helper.ko
  CC      drivers/gpu/drm/ttm/ttm.mod.o
  LD [M]  drivers/gpu/drm/ttm/ttm.ko
  CC      drivers/gpu/drm/vmwgfx/vmwgfx.mod.o
  LD [M]  drivers/gpu/drm/vmwgfx/vmwgfx.ko
  CC      drivers/hid/hid-generic.mod.o
  LD [M]  drivers/hid/hid-generic.ko
  CC      drivers/hid/hid.mod.o
  LD [M]  drivers/hid/hid.ko
  CC      drivers/hid/usbhid/usbhid.mod.o
  LD [M]  drivers/hid/usbhid/usbhid.ko
  CC      drivers/i2c/algos/i2c-algo-bit.mod.o
  LD [M]  drivers/i2c/algos/i2c-algo-bit.ko
  CC      drivers/i2c/busses/i2c-piix4.mod.o
  LD [M]  drivers/i2c/busses/i2c-piix4.ko
  CC      drivers/i2c/i2c-core.mod.o
  LD [M]  drivers/i2c/i2c-core.ko
  CC      drivers/input/evdev.mod.o
  LD [M]  drivers/input/evdev.ko
  CC      drivers/input/joydev.mod.o
  LD [M]  drivers/input/joydev.ko
  CC      drivers/input/mouse/psmouse.mod.o
  LD [M]  drivers/input/mouse/psmouse.ko
  CC      drivers/input/serio/serio_raw.mod.o
  LD [M]  drivers/input/serio/serio_raw.ko
  CC      drivers/misc/vmw_balloon.mod.o
  LD [M]  drivers/misc/vmw_balloon.ko
  CC      drivers/misc/vmw_vmci/vmw_vmci.mod.o
  LD [M]  drivers/misc/vmw_vmci/vmw_vmci.ko
  CC      drivers/net/ethernet/intel/e1000/e1000.mod.o
  LD [M]  drivers/net/ethernet/intel/e1000/e1000.ko
  CC      drivers/net/fjes/fjes.mod.o
  LD [M]  drivers/net/fjes/fjes.ko
  CC      drivers/usb/host/ehci-hcd.mod.o
  LD [M]  drivers/usb/host/ehci-hcd.ko
  CC      drivers/usb/host/ehci-pci.mod.o
  LD [M]  drivers/usb/host/ehci-pci.ko
  CC      drivers/usb/host/uhci-hcd.mod.o
  LD [M]  drivers/usb/host/uhci-hcd.ko
  CC      drivers/video/fbdev/core/fb_sys_fops.mod.o
  LD [M]  drivers/video/fbdev/core/fb_sys_fops.ko
  CC      drivers/video/fbdev/core/syscopyarea.mod.o
  LD [M]  drivers/video/fbdev/core/syscopyarea.ko
  CC      drivers/video/fbdev/core/sysfillrect.mod.o
  LD [M]  drivers/video/fbdev/core/sysfillrect.ko
  CC      drivers/video/fbdev/core/sysimgblt.mod.o
  LD [M]  drivers/video/fbdev/core/sysimgblt.ko
  CC      fs/fuse/fuse.mod.o
  LD [M]  fs/fuse/fuse.ko
  CC      net/ipv6/ipv6.mod.o
  LD [M]  net/ipv6/ipv6.ko
  CC      net/rfkill/rfkill.mod.o
  LD [M]  net/rfkill/rfkill.ko
  CC      net/wireless/cfg80211.mod.o
  LD [M]  net/wireless/cfg80211.ko
ash-4.3#
```

Następnie instalacja modułów

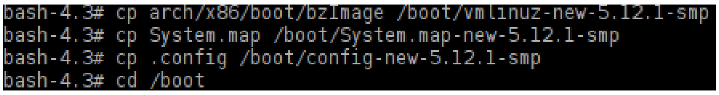


```
bash-4.3# make modules_install
  INSTALL arch/x86/crypto/crc32-pclmul.ko
  INSTALL arch/x86/crypto/glue_helper.ko
  INSTALL arch/x86/crypto/serpent-sse2-i586.ko
  INSTALL arch/x86/kernel/cpu/mce/mce-inject.ko
  INSTALL arch/x86/kvm/kvm-amd.ko
```



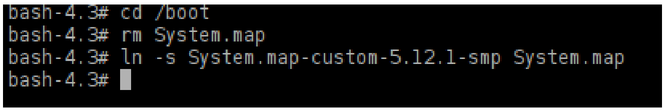
```
  INSTALL drivers/hid/usbhid/usbhid.ko
  INSTALL drivers/i2c/algos/i2c-algo-bit.ko
  INSTALL drivers/i2c/busses/i2c-piix4.ko
  INSTALL drivers/i2c/i2c-core.ko
  INSTALL drivers/input/evdev.ko
  INSTALL drivers/input/joydev.ko
  INSTALL drivers/input/mouse/psmouse.ko
  INSTALL drivers/input/serio/serio_raw.ko
  INSTALL drivers/misc/vmw_balloon.ko
  INSTALL drivers/misc/vmw_vmci/vmw_vmci.ko
  INSTALL drivers/net/ethernet/intel/e1000/e1000.ko
  INSTALL drivers/net/fjes/fjes.ko
  INSTALL drivers/usb/host/ehci-hcd.ko
  INSTALL drivers/usb/host/ehci-pci.ko
  INSTALL drivers/usb/host/uhci-hcd.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 net/ipv6/ipv6.ko
  INSTALL net/rfkill/rfkill.ko
  INSTALL net/wireless/cfg80211.ko
  DEPMOD  5.12.1-smp
bash-4.3# █
```

Potem przekopiowanie kernela do systemu



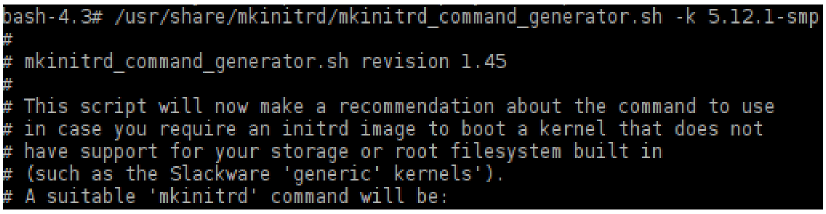
```
bash-4.3# cp arch/x86/boot/bzImage /boot/vmlinuz-new-5.12.1-smp
bash-4.3# cp System.map /boot/System.map-new-5.12.1-smp
bash-4.3# cp .config /boot/config-new-5.12.1-smp
bash-4.3# cd /boot
```

Tworze link w systemie dla tablicy symboli.

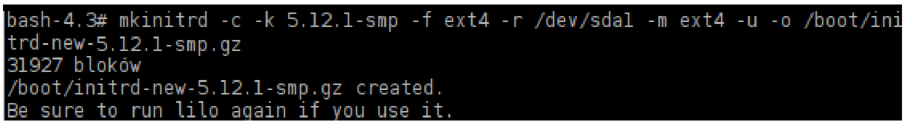


```
bash-4.3# cd /boot
bash-4.3# rm System.map
bash-4.3# ln -s System.map-custom-5.12.1-smp System.map
bash-4.3# █
```

Tworze dysk ram

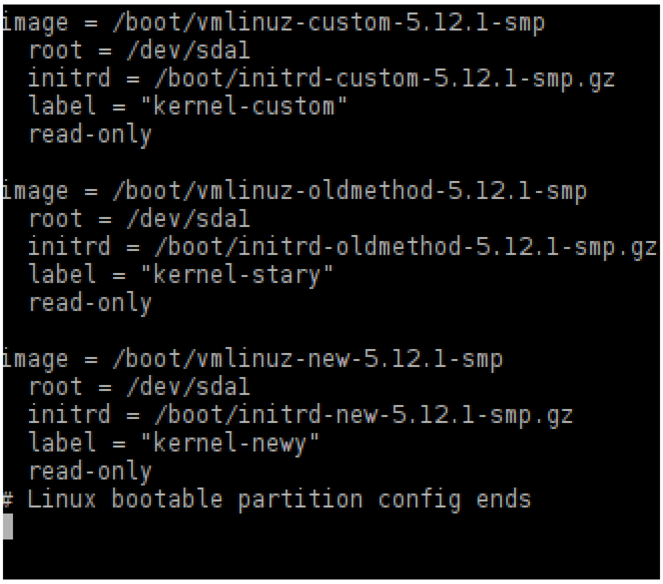


```
bash-4.3# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.12.1-smp
#
# mkinitrd_command_generator.sh revision 1.45
#
# This script will now make a recommendation about the command to use
# in case you require an initrd image to boot a kernel that does not
# have support for your storage or root filesystem built in
# (such as the Slackware 'generic' kernels').
# A suitable 'mkinitrd' command will be:
```



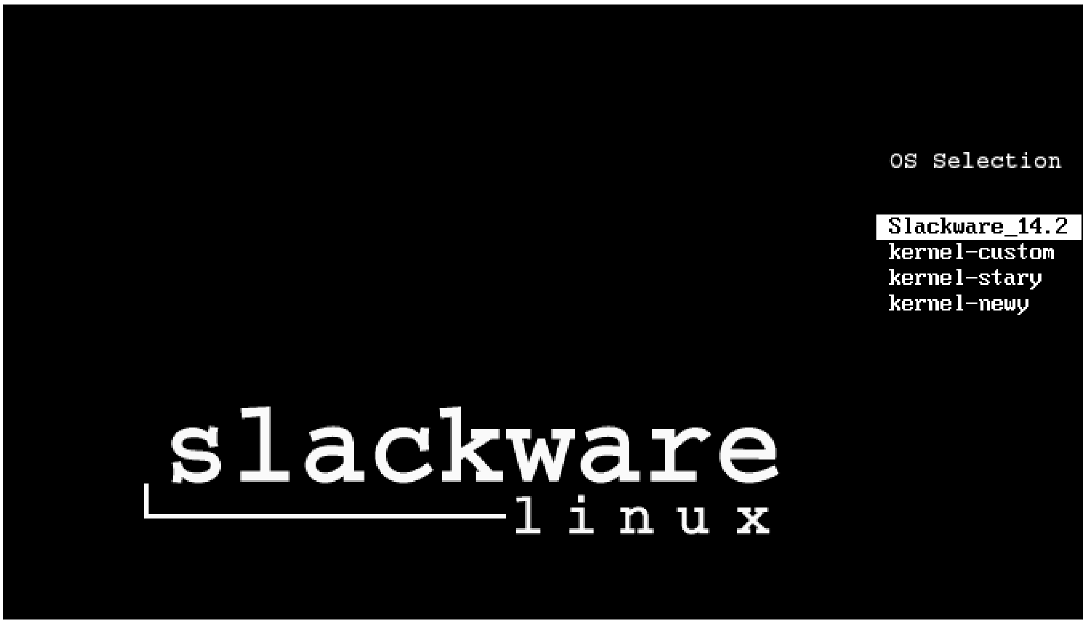
```
bash-4.3# mkinitrd -c -k 5.12.1-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/ini
trd-new-5.12.1-smp.gz
31927 bloków
/boot/initrd-new-5.12.1-smp.gz created.
Be sure to run lilo again if you use it.
```

Dodaje nowy wpis do konfiguracji bootloadera lilo



```
image = /boot/vmlinuz-custom-5.12.1-smp
  root = /dev/sda1
  initrd = /boot/initrd-custom-5.12.1-smp.gz
  label = "kernel-custom"
  read-only

image = /boot/vmlinuz-oldmethod-5.12.1-smp
  root = /dev/sda1
  initrd = /boot/initrd-oldmethod-5.12.1-smp.gz
  label = "kernel-stary"
  read-only

image = /boot/vmlinuz-new-5.12.1-smp
  root = /dev/sda1
  initrd = /boot/initrd-new-5.12.1-smp.gz
  label = "kernel-newy"
  read-only
# Linux bootable partition config ends
```

Wywołuje komedię lilo



3) Wnioski:

Przy drugiej próbie podejścia do kompilacji jądra nie maił już żadnych problemów i proces przebiła dużo szybciej bo wiedzom już jak się to robi. Nadal stara metoda okazała się szybsza od nowej metody. Nie widzę się w roli osoby która zajmuje się tym na co dzień.