

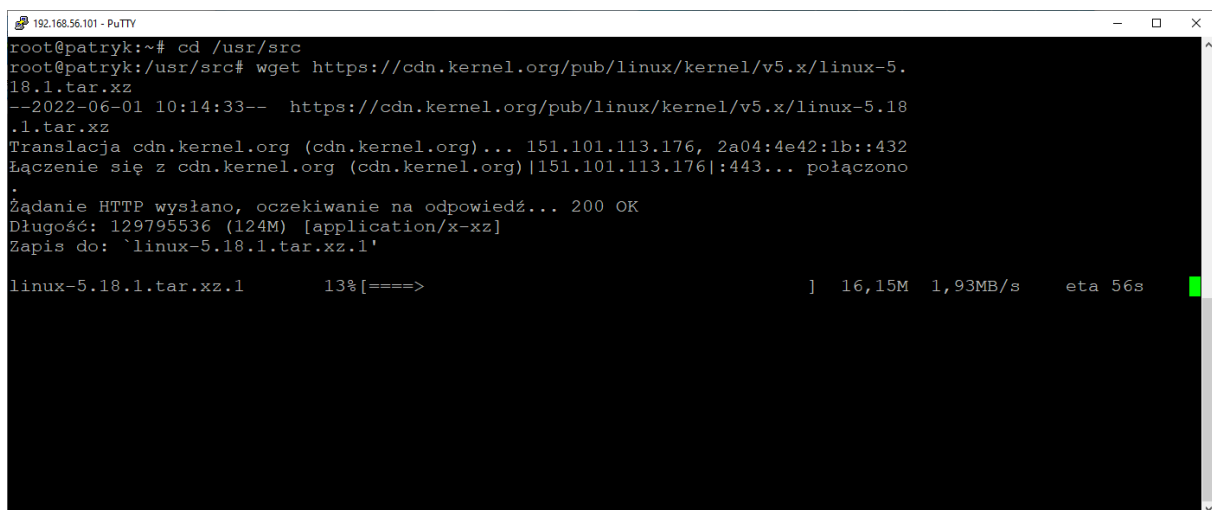
Kompilacja jądra Linux – Patryk Wojtiuk

1. Przygotowanie

Pierwszym elementem, który należy wykonać to pobranie jądra Linux ze strony <https://www.kernel.org/>. Wykorzystana wersja w niniejszej pracy to: **5.18.1**

W tym celu przechodzimy do katalogu `/usr/src` i wykonujemy polecenie `wget` oraz rozpakowujemy pobrane archiwum:

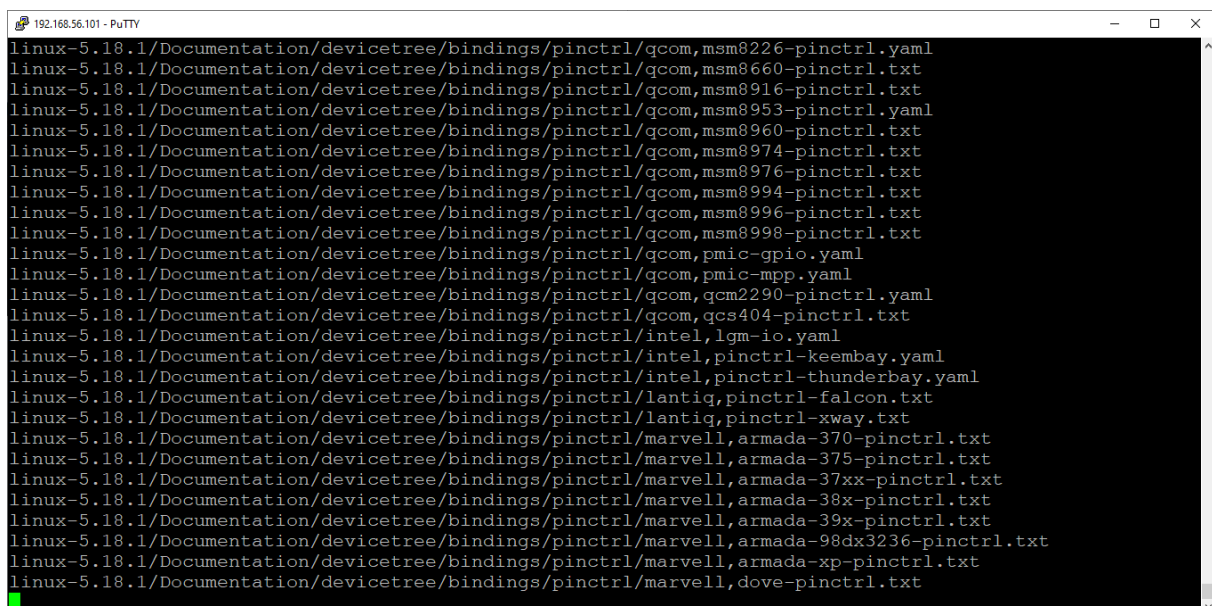
1. `cd /usr/src`
2. `wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.18.1.tar.xz`
3. `tar -xvpf linux-5.18.1.tar.xz`



```
192.168.56.101 - PuTTY
root@patryk:~# cd /usr/src
root@patryk:/usr/src# wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.18.1.tar.xz
--2022-06-01 10:14:33-- https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.18.1.tar.xz
Translacja cdn.kernel.org (cdn.kernel.org)... 151.101.113.176, 2a04:4e42:1b::432
Łączenie się z cdn.kernel.org (cdn.kernel.org)|151.101.113.176|:443... połączono
.
Żądanie HTTP wysłano, oczekiwanie na odpowiedź... 200 OK
Długość: 129795536 (124M) [application/x-xz]
Zapis do: `linux-5.18.1.tar.xz.1'

linux-5.18.1.tar.xz.1      13%[====>] 16,15M  1,93MB/s  eta 56s
```

Rysunek 1. Użycie komendy `wget`



```
192.168.56.101 - PuTTY
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8226-pinctrl.yaml
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8660-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8916-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8953-pinctrl.yaml
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8960-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8974-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8976-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8994-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8996-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,msm8998-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,pmic-gpio.yaml
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,pmic-mpp.yaml
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,qcm2290-pinctrl.yaml
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/qcom,qcs404-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/intel,lgm-io.yaml
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/intel,pinctrl-keembay.yaml
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/intel,pinctrl-thunderbay.yaml
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/lantiq,pinctrl-falcon.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/lantiq,pinctrl-xway.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/marvell,armada-370-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/marvell,armada-375-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/marvell,armada-37xx-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/marvell,armada-38x-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/marvell,armada-39x-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/marvell,armada-98dx3236-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/marvell,armada-xp-pinctrl.txt
linux-5.18.1/Documentation/devicetree/bindings/pinctrl/marvell,dove-pinctrl.txt
```

Rysunek 2. Rozpakowanie archiwum komendą `tar -xvpf linux-5.18.1.tar.xz`

2. Konfiguracja **localmodconfig** (metoda stara)

Przechodzimy do folderu z rozpakowanym kernelu:

- `cd linux-5.18.1`

Następnym elementem jest wykonanie kopii konfiguracji z starego jądra:

- `zcat /proc/config.gz > .config`

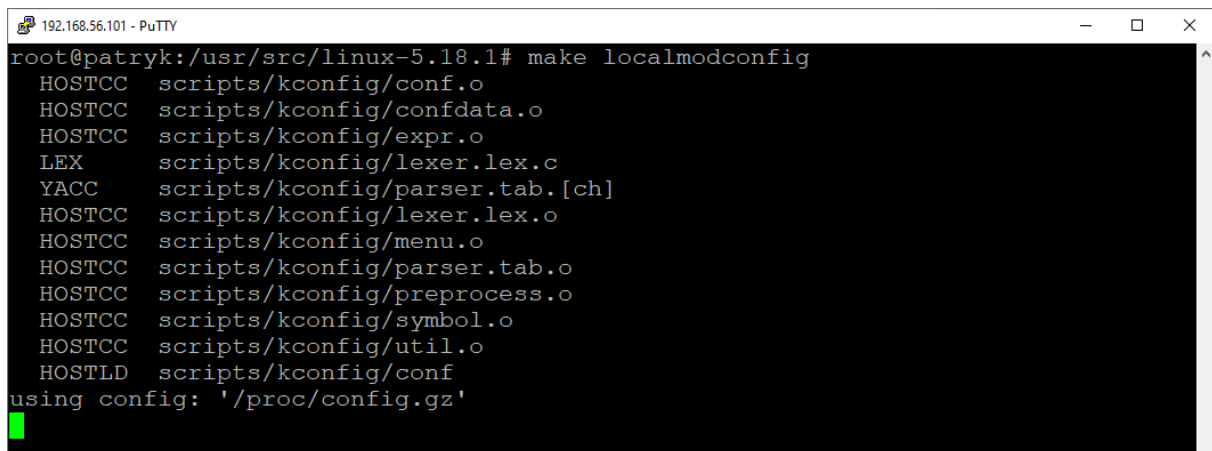


```
192.168.56.101 - PuTTY
root@patryk:/usr/src# zcat /proc/config.gz > .config
root@patryk:/usr/src#
```

Rysunek 3. Wykonywanie komendy `zcat /proc/config.gz > .config`

Następnie wykonujemy polecenie generacji pliku `.config`:

- `make localmodconfig`



```
192.168.56.101 - PuTTY
root@patryk:/usr/src/linux-5.18.1# make localmodconfig
HOSTCC scripts/kconfig/conf.o
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o
LEX scripts/kconfig/lexer.lex.c
YACC scripts/kconfig/parser.tab.[ch]
HOSTCC scripts/kconfig/lexer.lex.o
HOSTCC scripts/kconfig/menu.o
HOSTCC scripts/kconfig/parser.tab.o
HOSTCC scripts/kconfig/preprocess.o
HOSTCC scripts/kconfig/symbol.o
HOSTCC scripts/kconfig/util.o
HOSTLD scripts/kconfig/conf
using config: '/proc/config.gz'
```

Rysunek 4. Wykonanie komendy `make localmodconfig`

```
192.168.56.101 - PuTTY
LEX      scripts/kconfig/lexer.lex.c
YACC     scripts/kconfig/parser.tab.[ch]
HOSTCC   scripts/kconfig/lexer.lex.o
HOSTCC   scripts/kconfig/menu.o
HOSTCC   scripts/kconfig/parser.tab.o
HOSTCC   scripts/kconfig/preprocess.o
HOSTCC   scripts/kconfig/symbol.o
HOSTCC   scripts/kconfig/util.o
HOSTLD   scripts/kconfig/conf
using config: '/proc/config.gz'
*
* Restart config...
*
*
* Timers subsystem
*
Timer tick handling
  1. Periodic timer ticks (constant rate, no dynticks) (HZ_PERIODIC)
> 2. Idle dynticks system (tickless idle) (NO_HZ_IDLE)
choice[1-2]: 2
Old Idle dynticks config (NO HZ) [Y/n/?] y
High Resolution Timer Support (HIGH RES TIMERS) [Y/n/?] y
Clocksource watchdog maximum allowable skew (in us) (CLOCKSOURCE_WATCHDOG_MAX_SK
EW_US) [100] (NEW)
```

Rysunek 5. Localmodconfig opcje

W tym przypadku wszystkie opcje pozostawiam na statusie domyślnym potwierdzając klawiszem ENTER

```
192.168.56.101 - PuTTY
Test the XArray code at runtime (TEST_XARRAY) [N/m/y/?] n
Perform selftest on resizable hash table (TEST_RHASH_TABLE) [N/m/y/?] n
Perform selftest on siphash functions (TEST_SIPHASH) [N/m/y/?] (NEW)
Perform selftest on IDA functions (TEST_IDA) [N/m/y/?] n
Test module loading with 'hello world' module (TEST_LKM) [N/m/?] n
Test module for compilation of bitops operations (TEST_BITOPS) [N/m/?] n
Test module for stress/performance analysis of vmalloc allocator (TEST_VMLLOC) [N/m/?] n
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/?] n
Test find bit functions (FIND_BIT_BENCHMARK) [N/m/y/?] n
Test firmware loading via userspace interface (TEST_FIRMWARE) [N/m/y/?] n
sysctl test driver (TEST_SYSCTL) [N/m/y/?] n
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/?] n
Test memcat_p() helper function (TEST_MEMCAT_P) [N/m/y/?] n
Test heap/page initialization (TEST_MEMINIT) [N/m/y/?] n
Test freeing pages (TEST_FREE_PAGES) [N/m/y/?] n
Test floating point operations in kernel space (TEST_FPU) [N/m/y/?] n
Test clocksource watchdog in kernel space (TEST_CLOCKSOURCE_WATCHDOG) [N/m/y/?] n
#
# configuration written to .config
#
root@patryk:/usr/src/linux-5.18.1#
root@patryk:/usr/src/linux-5.18.1#
root@patryk:/usr/src/linux-5.18.1#
```

Rysunek 6. Wynik działania localmodconfig

Krokiem opcjonalnym jest sprawdzenie zainstalowanych modułów, konfiguracji:

Sprawdzenie załadowanych modułów:

- lsmod

```
192.168.56.101 - PuTTY
root@patryk:/usr/src/linux-5.18.1# lsmod
Module                Size  Used by
vboxvideo              24576  0
drm_vram_helper        20480  1 vboxvideo
drm_ttm_helper         16384  2 drm_vram_helper,vboxvideo
cfg80211               770048  0
8021q                  28672  0
garp                   16384  1 8021q
mrp                    20480  1 8021q
stp                    16384  1 garp
llc                     16384  2 garp,stp
rfkill                 24576  1 cfg80211
ipv6                   458752  20
intel_rapl_msr         20480  0
joydev                 20480  0
vmwgfx                 270336  2
intel_rapl_common      24576  1 intel_rapl_msr
ttm                    61440  3 drm_vram_helper,drm_ttm_helper,vmwgfx
snd_intel8x0           36864  0
drm_kms_helper         237568  3 drm_vram_helper,vboxvideo,vmwgfx
rapl                   20480  0
snd_ac97_codec         122880  1 snd_intel8x0
intel_cstate           20480  0
drm                    475136  8 drm_vram_helper,drm_ttm_helper,vboxvideo,vmwgfx,ttm,drm_kms_helper
evdev                  20480  13
snd_pcm                102400  2 snd_ac97_codec,snd_intel8x0
psmouse               126976  0
snd_timer              32768  1 snd_pcm
fb_sys_fops            16384  1 drm_kms_helper
syscopyarea            16384  1 drm_kms_helper
snd                    73728  4 snd_ac97_codec,snd_timer,snd_intel8x0,snd_pcm
sysfillrect            16384  1 drm_kms_helper
soundcore              16384  1 snd
i2c_piix4              20480  0
sysimgblt              16384  1 drm_kms_helper
serio_raw              16384  0
ac97_bus               16384  1 snd_ac97_codec
```

Rysunek 7. Wykonanie komedy lsmod

Sprawdzenie konfiguracji kernela można wykonać na kilka sposób w tym przypadku użyłem sposobu dotyczącego trybu tekstowego z wykorzystaniem z oknem dialogowym:

- make menuconfig

```
192.168.56.101 - PuTTY
.config - Linux/x86 5.18.1 Kernel Configuration
Linux/x86 5.18.1 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --- (or empty submenu ---).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable
General setup ---
[ ] 64-bit kernel
Processor type and features ---
Power management and ACPI options ---
Bus options (PCI etc.) ---
Binary Emulations ---
[*] Virtualization ---
General architecture-dependent options ---
[*] Enable loadable module support ---
[*] Enable the block layer ---
Executable file formats ---
Memory Management options ---
[*] Networking support ---
Device Drivers ---
File systems ---
Security options ---
-- Cryptographic API ---
Library routines ---
Kernel hacking ---
<Select> <Exit> <Help> <Save> <Load>
```

Rysunek 8. Wykonanie komendy make menuconfig

Polecenie nie wykryło żadnych niedogodności

```
root@patryk:/usr/src/linux-5.18.1# 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

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

Rysunek 9. Wynik polecenia `make menuconfig`

Sprawdzenie pliku konfiguracyjnego z poprzedniego kernela w celu upewnienia się czy nic nie należy wykonać dodatkowego:

- `make olddefconfig`

```
192.168.56.101 - PuTTY
root@patryk:/usr/src/linux-5.18.1# make olddefconfig
#
# No change to .config
#
root@patryk:/usr/src/linux-5.18.1#
```

Rysunek 10. Wykonanie komendy `make olddefconfig`

Następnym elementem jest zbudowanie obrazu jądra w tym celu została wykorzystana komenda:

- `make -j2 bzImage`

Gdzie parametr `-j2` oznacza liczbę wykorzystanych rdzeni w tym przypadku 2.

```
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
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
SYSHDR   arch/x86/include/generated/uapi/asm/unistd_x32.h
WRAP     arch/x86/include/generated/uapi/asm/types.h
SYSTBL   arch/x86/include/generated/asm/syscalls_32.h
HOSTCC   arch/x86/tools/relocs_32.o
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/mmiowb.h
WRAP     arch/x86/include/generated/asm/module.lds.h
WRAP     arch/x86/include/generated/asm/rwonce.h
WRAP     arch/x86/include/generated/asm/unaligned.h
HOSTCC   arch/x86/tools/relocs_64.o
UPD      include/generated/uapi/linux/version.h
UPD      include/generated/utsrelease.h
```

Rysunek 11. Użycie komendy `make -j2 bzImage`

```
192.168.56.101 - PuTTY
AR arch/x86/kernel/cpu/resctrl/built-in.a
CC arch/x86/kernel/cpu/cacheinfo.o
CC kernel/rcu/sync.o
CC kernel/rcu/srcutree.o
CC arch/x86/kernel/cpu/scattered.o
CC kernel/rcu/tree.o
CC arch/x86/kernel/cpu/topology.o
CC arch/x86/kernel/cpu/common.o
CC arch/x86/kernel/cpu/rdrand.o
CC arch/x86/kernel/cpu/match.o
CC arch/x86/kernel/cpu/bugs.o
CC kernel/rcu/rcu_segcblist.o
CC arch/x86/kernel/cpu/aperfmpperf.o
AR kernel/rcu/built-in.a
CC arch/x86/kernel/cpu/cpuid-deps.o
AR kernel/livepatch/built-in.a
CC kernel/dma/mapping.o
CC arch/x86/kernel/cpu/umwait.o
CC kernel/dma/direct.o
CC arch/x86/kernel/cpu/proc.o
MKCAP arch/x86/kernel/cpu/capflags.c
CC kernel/dma/ops_helpers.o
CC kernel/dma/dummy.o
CC kernel/dma/contiguous.o
CC kernel/dma/swiotlb.o
CC kernel/dma/remap.o
AR kernel/dma/built-in.a
CC kernel/entry/common.o
CC kernel/entry/syscall_user_dispatch.o
AR kernel/entry/built-in.a
CC kernel/time/time.o
CC kernel/time/timer.o
```

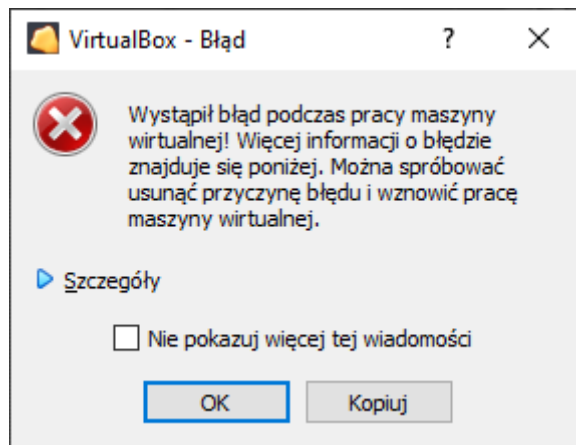
Rysunek 12. Ciąg dalszy budowania obrazu

Proces budowania obrazu jądra rozpoczął się o godzinie **11:53**

Godzina 12:34 Niestety podczas budowania obrazu zabrakło miejsca na dysku

```
192.168.56.101 - PuTTY
CC arch/x86/boot/edd.o
CC arch/x86/boot/main.o
CC arch/x86/boot/memory.o
LDS arch/x86/boot/compressed/vmlinux.lds
AS arch/x86/boot/compressed/kernel_info.o
CC arch/x86/boot/pm.o
AS arch/x86/boot/compressed/head_32.o
VOFFSET arch/x86/boot/compressed/./voffset.h
AS arch/x86/boot/pmjump.o
CC arch/x86/boot/printf.o
CC arch/x86/boot/compressed/string.o
CC arch/x86/boot/regs.o
CC arch/x86/boot/compressed/cmdline.o
CC arch/x86/boot/string.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
RELOCS arch/x86/boot/compressed/vmlinux.relocs
CC arch/x86/boot/video-bios.o
HOSTCC arch/x86/boot/compressed/mkpiggy
CC arch/x86/boot/compressed/cpuflags.o
HOSTCC arch/x86/boot/tools/build
CC arch/x86/boot/compressed/early_serial_console.o
CC arch/x86/boot/compressed/kaslr.o
CC arch/x86/boot/compressed/acpi.o
CC arch/x86/boot/compressed/misc.o
LZMA arch/x86/boot/compressed/vmlinux.bin.lzma
CPUSTR arch/x86/boot/cpustr.h
CC arch/x86/boot/cpu.o
```

Rysunek 13. Widok terminala, który zaprzestał pracować (stał w miejscu)



Rysunek 14. Błąd wynikający z braku miejsca na dysku

Ponowne uruchomienie maszyny i wykonanie komendy `make -j2 bzImage`, proces budowania obrazu rozpoczął się o godzinie **12:57**

W tym przypadku wystąpił błąd:

```

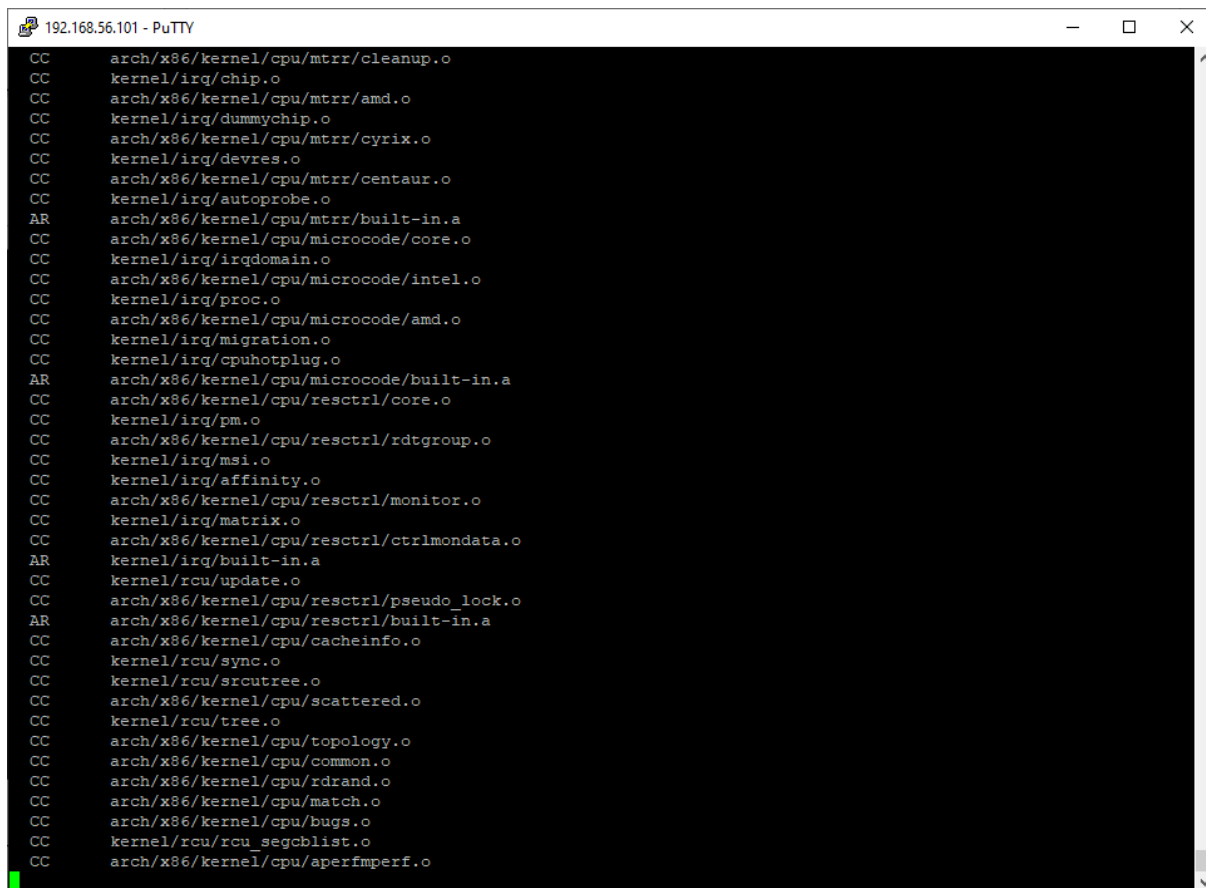
192.168.56.101 - PuTTY
login as: root
Keyboard-interactive authentication prompts from server:
| Password:
| End of keyboard-interactive prompts from server
Last login: Wed Jun  1 10:34:11 2022 from 192.168.56.1
Linux 5.15.27-smp.
root@patryk:~# cd /usr/src
root@patryk:/usr/src# ls
linux-5.17.8/  linux-5.17.8.tar.xz  linux-5.18.1/  linux-5.18.1.tar.xz
root@patryk:/usr/src# cd linux-5.18.1
root@patryk:/usr/src/linux-5.18.1# ls
COPYING      Makefile      drivers/      mm/           sound/
CREDITS      README       fs/           modules.builtin  tools/
Documentation/ System.map    include/      modules.builtin.modinfo  usr/
Kbuild       arch/        init/        net/          virt/
Kconfig      block/      ipc/         samples/      vmlinux*
LICENSES/    certs/      kernel/      scripts/      vmlinux.o
MAINTAINERS  crypto/     lib/         security/     vmlinux.symvers
root@patryk:/usr/src/linux-5.18.1# make -j2 bzImage
CALL      scripts/atomic/check-atomics.sh
CALL      scripts/checksyscalls.sh
CHK       include/generated/compile.h
HOSTCC    arch/x86/boot/tools/build
CPUSTR    arch/x86/boot/cpustr.h
CC        arch/x86/boot/compressed/misc.o
CC        arch/x86/boot/video-vesa.o
CC        arch/x86/boot/video-bios.o
CC        arch/x86/boot/compressed/cpuflags.o
CC        arch/x86/boot/compressed/early_serial_console.o
CC        arch/x86/boot/compressed/kaslr.o
CC        arch/x86/boot/compressed/acpi.o
OBJCOPY   arch/x86/boot/compressed/vmlinux.bin
objcopy: vmlinux: file format not recognized
make[2]: *** [arch/x86/boot/compressed/Makefile:113: arch/x86/boot/compressed/vmlinux.bin] Błąd 1
make[2]: *** Oczekiwanie na niezakończone zadania....
CC        arch/x86/boot/cpu.o
make[1]: *** [arch/x86/boot/Makefile:115: arch/x86/boot/compressed/vmlinux] Błąd 2
make: *** [arch/x86/Makefile:265: bzImage] Błąd 2
root@patryk:/usr/src/linux-5.18.1#

```

Rysunek 15. Komunikat o błędzie

Niestety migawkę wykonałem na początku prac, także muszę powtórzyć całą konfigurację, zatem w celu rozwiązania problemu postanowiłem wykonać powyższe kroki od nowa.

Rozpoczęcie budowy obrazu godzina **13:17**



```
192.168.56.101 - PuTTY
CC      arch/x86/kernel/cpu/mtrr/cleanup.o
CC      kernel/irq/chip.o
CC      arch/x86/kernel/cpu/mtrr/amd.o
CC      kernel/irq/dummychip.o
CC      arch/x86/kernel/cpu/mtrr/cyrix.o
CC      kernel/irq/devres.o
CC      arch/x86/kernel/cpu/mtrr/centaur.o
CC      kernel/irq/autoprobe.o
AR      arch/x86/kernel/cpu/mtrr/built-in.a
CC      arch/x86/kernel/cpu/microcode/core.o
CC      kernel/irq/irqdomain.o
CC      arch/x86/kernel/cpu/microcode/intel.o
CC      kernel/irq/proc.o
CC      arch/x86/kernel/cpu/microcode/amd.o
CC      kernel/irq/migration.o
CC      kernel/irq/cpuhotplug.o
AR      arch/x86/kernel/cpu/microcode/built-in.a
CC      arch/x86/kernel/cpu/resctrl/core.o
CC      kernel/irq/pm.o
CC      arch/x86/kernel/cpu/resctrl/rdtgroup.o
CC      kernel/irq/msi.o
CC      kernel/irq/affinity.o
CC      arch/x86/kernel/cpu/resctrl/monitor.o
CC      kernel/irq/matrix.o
CC      arch/x86/kernel/cpu/resctrl/ctrlmondata.o
AR      kernel/irq/built-in.a
CC      kernel/rcu/update.o
CC      arch/x86/kernel/cpu/resctrl/pseudo_lock.o
AR      arch/x86/kernel/cpu/resctrl/built-in.a
CC      arch/x86/kernel/cpu/cacheinfo.o
CC      kernel/rcu/sync.o
CC      kernel/rcu/srcutree.o
CC      arch/x86/kernel/cpu/scattered.o
CC      kernel/rcu/tree.o
CC      arch/x86/kernel/cpu/topology.o
CC      arch/x86/kernel/cpu/common.o
CC      arch/x86/kernel/cpu/rdand.o
CC      arch/x86/kernel/cpu/match.o
CC      arch/x86/kernel/cpu/bugs.o
CC      kernel/rcu/rcu_segcblist.o
CC      arch/x86/kernel/cpu/aperfmperf.o
```

Rysunek 16. Ponowne budowanie obrazu jądra

Zakończenie budowy obrazu jądra godzina **13:50**. Zatem sama budowa jądra trwała około 33 minuty. Jest to czas dużo szybszy w porównaniu niż jaki miał miejsce poprzednio (na zajęciach) – być może zależne od obciążenia komputera.

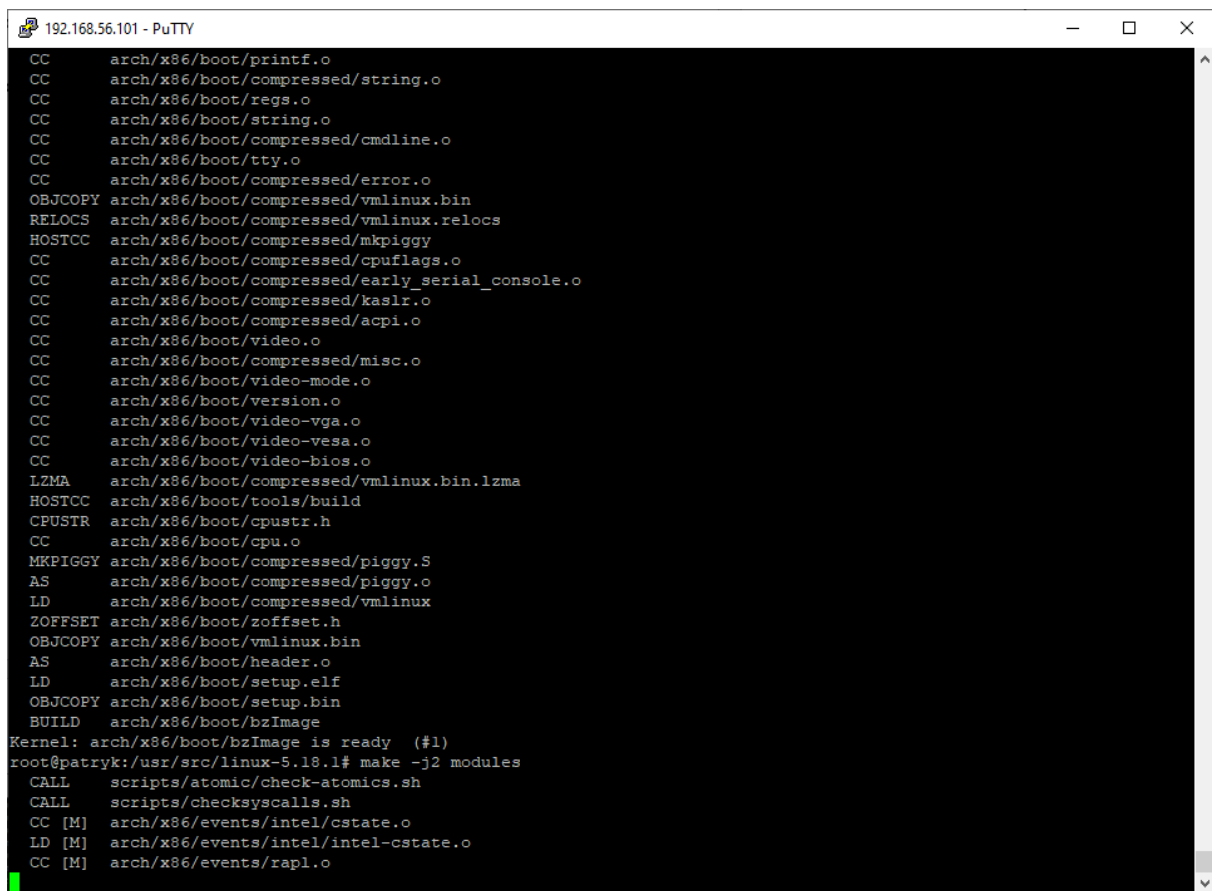

```
192.168.56.101 - PuTTY
LDS      arch/x86/boot/compressed/vmlinux.lds
AS        arch/x86/boot/compressed/kernel_info.o
AS        arch/x86/boot/compressed/head_32.o
CC        arch/x86/boot/pm.o
VOFFSET  arch/x86/boot/compressed/./voffset.h
AS        arch/x86/boot/pmjump.o
CC        arch/x86/boot/printf.o
CC        arch/x86/boot/compressed/string.o
CC        arch/x86/boot/regs.o
CC        arch/x86/boot/string.o
CC        arch/x86/boot/compressed/cmdline.o
CC        arch/x86/boot/tty.o
CC        arch/x86/boot/compressed/error.o
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
CC        arch/x86/boot/compressed/kaslr.o
CC        arch/x86/boot/compressed/acpi.o
CC        arch/x86/boot/video.o
CC        arch/x86/boot/compressed/misc.o
CC        arch/x86/boot/video-mode.o
CC        arch/x86/boot/version.o
CC        arch/x86/boot/video-vga.o
CC        arch/x86/boot/video-vesa.o
CC        arch/x86/boot/video-bios.o
LZMA     arch/x86/boot/compressed/vmlinux.bin.lzma
HOSTCC   arch/x86/boot/tools/build
CPUSTR   arch/x86/boot/cpustr.h
CC        arch/x86/boot/cpu.o
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@patryk:/usr/src/linux-5.18.1#
```

Rysunek 17. Zakończenie budowania obrazu jądra

Następnie odbyło się budowanie modułów jądra poprzez wykonanie komendy:

- `make -j2 modules`

Budowanie modułów rozpoczęło się o godzinie **13:56**



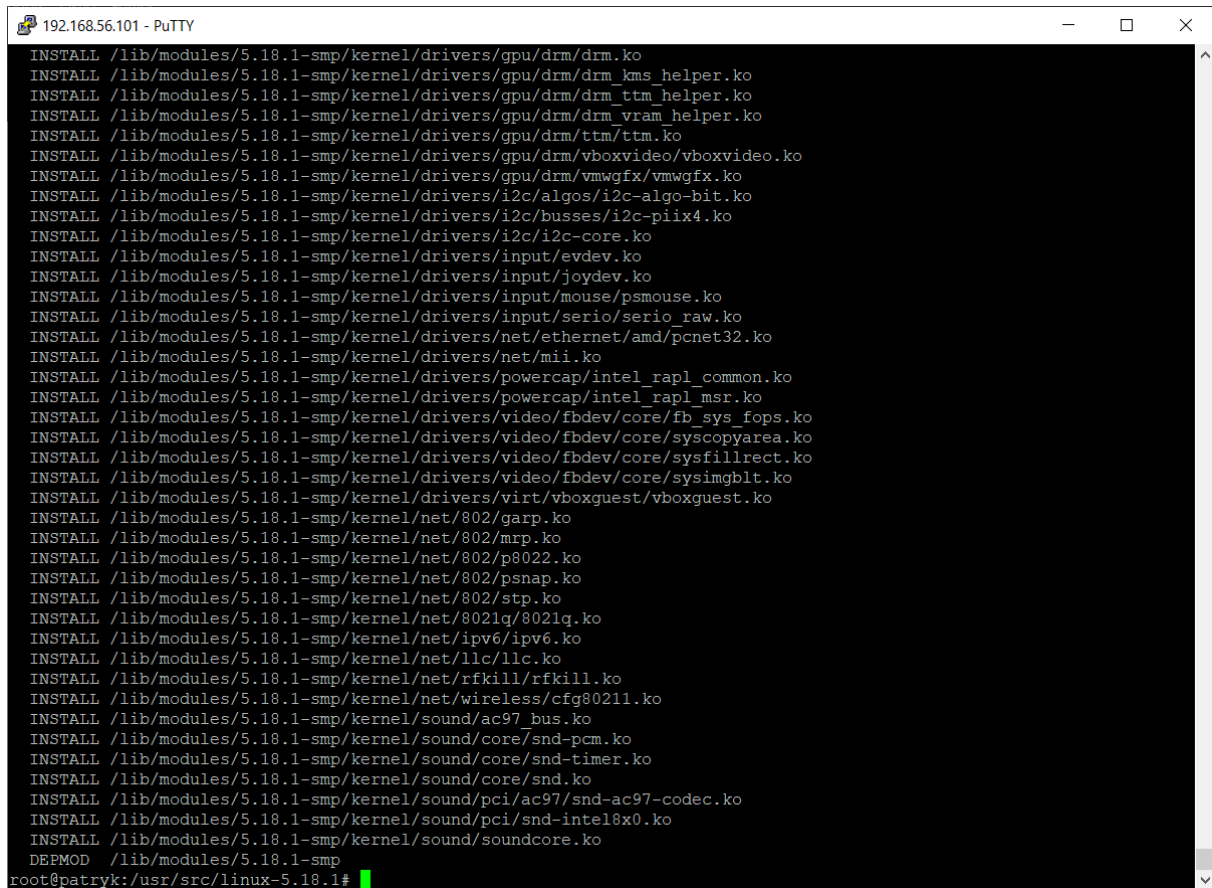
```
CC      arch/x86/boot/printf.o
CC      arch/x86/boot/compressed/string.o
CC      arch/x86/boot/regs.o
CC      arch/x86/boot/string.o
CC      arch/x86/boot/compressed/cmdline.o
CC      arch/x86/boot/tty.o
CC      arch/x86/boot/compressed/error.o
OBJCOPY arch/x86/boot/compressed/vmlinux.bin
RELOCS  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
CC      arch/x86/boot/compressed/kaslr.o
CC      arch/x86/boot/compressed/acpi.o
CC      arch/x86/boot/video.o
CC      arch/x86/boot/compressed/misc.o
CC      arch/x86/boot/video-mode.o
CC      arch/x86/boot/version.o
CC      arch/x86/boot/video-vga.o
CC      arch/x86/boot/video-vesa.o
CC      arch/x86/boot/video-bios.o
LZMA    arch/x86/boot/compressed/vmlinux.bin.lzma
HOSTCC  arch/x86/boot/tools/build
CPUSTR  arch/x86/boot/cpustr.h
CC      arch/x86/boot/cpu.o
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@patryk:/usr/src/linux-5.18.1# make -j2 modules
CALL    scripts/atomic/check-atomics.sh
CALL    scripts/checksyscalls.sh
CC [M]  arch/x86/events/intel/cstate.o
LD [M]  arch/x86/events/intel/intel-cstate.o
CC [M]  arch/x86/events/rapl.o
```

Rysunek 18. Budowanie modułów jądra komendą `make -j2 modules`

Zakończenie budowania modułów nastąpiło o godzinie **14:10**

Następnym etapem jest zainstalowanie modułów za pomocą komendy:

- `make modules_install`



```
192.168.56.101 - PuTTY
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/drm.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/drm_kms_helper.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/drm_ttm_helper.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/drm_vram_helper.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/ttm/ttm.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/vboxvideo/vboxvideo.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/vmwgfx/vmwgfx.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/i2c/algo/i2c-algo-bit.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/i2c/busses/i2c-piix4.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/i2c/i2c-core.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/input/evdev.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/input/joydev.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/input/mouse/psmouse.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/input/serio/serio_raw.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/net/ethernet/amd/pcnet32.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/net/mii.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/powercap/intel_rapl_common.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/powercap/intel_rapl_msr.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/video/fbdev/core/fb_sys_fops.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/video/fbdev/core/syscopyarea.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/video/fbdev/core/sysfillrect.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/video/fbdev/core/sysimgblt.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/virt/vboxguest/vboxguest.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/garp.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/mrp.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/p8022.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/psnap.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/stp.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/8021q/8021q.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/ipv6/ipv6.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/llc/llc.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/rfkill/rfkill.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/wireless/cfg80211.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/ac97_bus.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/core/snd-pcm.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/core/snd-timer.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/core/snd.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/pci/ac97/snd-ac97-codec.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/pci/snd-intel8x0.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/soundcore.ko
DEPMOD /lib/modules/5.18.1-smp
root@patryk:/usr/src/linux-5.18.1#
```

Rysunek 19. Zakończenie instalacji modułów

Aby sprawdzić zainstalowane moduły wykonano komendę:

- `ls /lib/modules/5.18.1-smp`

```
192.168.56.101 - PuTTY
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/powercap/intel_rapl_msr.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/video/fbdev/core/fb_sys_fops.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/video/fbdev/core/syscopyarea.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/video/fbdev/core/sysfillrect.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/video/fbdev/core/sysimgblt.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/virt/vboxguest/vboxguest.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/garp.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/mrp.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/p8022.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/psnap.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/802/stp.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/8021q/8021q.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/ipv6/ipv6.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/llc/llc.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/rfkill/rfkill.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/wireless/cfg80211.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/ac97_bus.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/core/snd-pcm.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/core/snd-timer.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/core/snd.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/pci/ac97/snd-ac97-codec.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/pci/snd-intel8x0.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/soundcore.ko
DEPMOD /lib/modules/5.18.1-smp
root@patryk:/usr/src/linux-5.18.1# ls
COPYING      MAINTAINERS  block/      init/        modules.builtin      security/    vmlinux.o
CREDITS      Makefile     certs/      ipc/         modules.builtin.modinfo  sound/      vmlinux.symvers
Documentation/ Module.symvers crypto/      kernel/      modules.order        tools/
Kbuild       README       drivers/    lib/         net/            usr/
Kconfig      System.map   fs/         mm/          samples/        virt/
LICENSES/    arch/        include/    modules-only.symvers  scripts/        vmlinux*
root@patryk:/usr/src/linux-5.18.1# cd /lib/
root@patryk:/lib# cd
root@patryk:~# ls /lib/mod
modprobe.d/ modules/
root@patryk:~# ls /lib/mod
modprobe.d/ modules/
root@patryk:~# ls /lib/modules/5.18.1-smp/
build@      modules.alias.bin      modules.builtin.bin      modules.dep.bin      modules.softdep      source@
kernel/     modules.builtin        modules.builtin.modinfo  modules.devname      modules.symbols
modules.alias      modules.builtin.alias.bin  modules.dep            modules.order        modules.symbols.bin
root@patryk:~#
```

Rysunek 20. Sprawdzenie zainstalowanych modułów

Następnym etapem jest przekopiowanie jądra do katalogu boot, tablicy symboli używanych przez kernela oraz pliku konfiguracyjnego

- `cp arch/x86/boot/bzImage /boot/vmlinuz-oldmethod-5.18.1-smp`
- `cp System.map /boot/System.map-oldmethod-5.18.1-smp`
- `cp .config /boot/config-oldmethod-5.18.1-smp`

```
192.168.56.101 - PuTTY
INSTALL /lib/modules/5.18.1-smp/kernel/net/llc/llc.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/rfkill/rfkill.ko
INSTALL /lib/modules/5.18.1-smp/kernel/net/wireless/cfg80211.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/ac97_bus.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/core/snd-pcm.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/core/snd-timer.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/core/snd.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/pci/ac97/snd-ac97-codec.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/pci/snd-intel8x0.ko
INSTALL /lib/modules/5.18.1-smp/kernel/sound/soundcore.ko
DEPMOD /lib/modules/5.18.1-smp
root@patryk:/usr/src/linux-5.18.1# ls
COPYING      MAINTAINERS  block/      init/        modules.builtin      security/  vmlin
CREDITS      Makefile     certs/      ipc/         modules.builtin.modinfo  sound/    vmlin
Documentation/ Module.symvers crypto/      kernel/        modules.order        tools/
Kbuild       README       drivers/    lib/          net/                usr/
Kconfig      System.map   fs/         mm/           samples/            virt/
LICENSES/    arch/        include/    modules-only.symvers  scripts/            vmlinux*
root@patryk:/usr/src/linux-5.18.1# cd /lib/
root@patryk:/lib# cd
root@patryk:~# ls /lib/mod
modprobe.d/ modules/
root@patryk:~# ls /lib/mod
modprobe.d/ modules/
root@patryk:~# ls /lib/modules/5.18.1-smp/
build@      modules.alias.bin      modules.builtin.bin      modules.dep.bin      modules.softdep
kernel/     modules.builtin         modules.builtin.modinfo  modules.devname      modules.symbols
modules.alias modules.builtin.alias.bin modules.dep              modules.order         modules.symbols.bin
root@patryk:~# ls
Dokumenty/ Muzyka/ Obrazy/ Pobrane/ Publiczny/ Pulpit/ Szablony/ Wideo/ testowyKernel/
root@patryk:~# cd /usr/src/linux-5.18.1/
root@patryk:/usr/src/linux-5.18.1# ls
COPYING      MAINTAINERS  block/      init/        modules.builtin      security/  vmlin
CREDITS      Makefile     certs/      ipc/         modules.builtin.modinfo  sound/    vmlin
Documentation/ Module.symvers crypto/      kernel/        modules.order        tools/
Kbuild       README       drivers/    lib/          net/                usr/
Kconfig      System.map   fs/         mm/           samples/            virt/
LICENSES/    arch/        include/    modules-only.symvers  scripts/            vmlinux*
root@patryk:/usr/src/linux-5.18.1# cp arch/x86/boot/bzImage /boot/vmlinuz-oldmethod-5.18.1-smp
root@patryk:/usr/src/linux-5.18.1# cp System.map /boot/System.map-oldmethod-5.18.1-smp
root@patryk:/usr/src/linux-5.18.1# cp .config /boot/config-oldmethod-5.18.1-smp
root@patryk:/usr/src/linux-5.18.1#
```

Rysunek 21. Wykonanie powyższych poleceń

Następnym etapem jest utworzenie linku symbolicznego dla tablicy symboli kernela, w tym przypadku należy przejść do katalogu *boot*, usunąć starą tablicę symboli oraz utworzyć link symboliczny. Wykonanie powyższego polecenia zostało wykonane poprzez polecenia:

- `cd /boot`
- `rm System.map`
- `ln -s System.map-oldmethod-5.18.1-smp System.map`

```
192.168.56.101 - PuTTY
root@patryk:/usr/src/linux-5.18.1# cd /boot
root@patryk:/boot# rm System.map
root@patryk:/boot# ln -s System.map-oldmethod-5.18.1-smp System.map
root@patryk:/boot#
```

Rysunek 22. Wykonanie powyższych poleceń

Następnym etapem jest utworzenie dysku RAM, w tym celu wykonano skrypt generujący komendę do wykonania za pośrednictwem następującego polecenia:

- `/usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.18.1-smp`

```
192.168.56.101 - PuTTY
root@patryk:~# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.18.1-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:

mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sdal -m ext4 -u -o /boot/initrd.gz
root@patryk:~#
```

Rysunek 23. Użycie powyższej komendy

Wygenerowane polecenie należy zmienić do następującej postaci:

- `mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd-oldmethod-5.18.1-smp.gz`

```
192.168.56.101 - PuTTY
root@patryk:~# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.18.1-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:

mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sdal -m ext4 -u -o /boot/initrd.gz
root@patryk:~# mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sdal -m ext4 -u -o /boot/initrd-oldmethod-5.18.1-smp.gz
49039 bloków
/boot/initrd-oldmethod-5.18.1-smp.gz created.
Be sure to run lilo again if you use it.
root@patryk:~#
```

Rysunek 24. Wykonanie powyższego polecenia

Następnym punktem odniesienia jest dodanie nowego wpisu do konfiguracji bootloadera lilo, w tym celu należy podać edycji plik *lilo.conf*

- `nano /etc/lilo.conf`

```
192.168.56.101 - PuTTY
GNU nano 6.0 /etc/lilo.conf
# LILO configuration file
# generated by 'liloconfig'
#
# Start LILO global section
# Append any additional kernel parameters:
append=" "

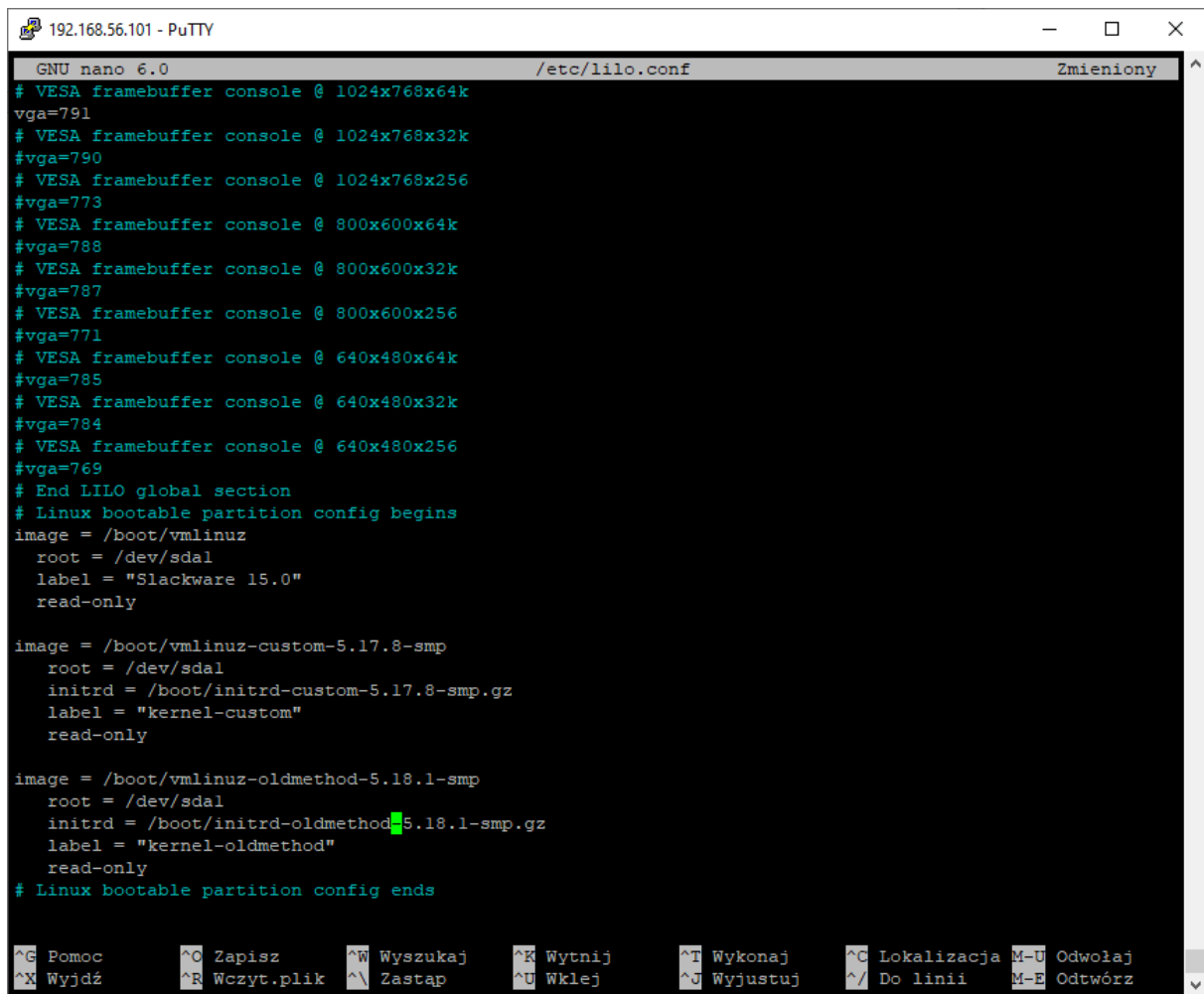
boot = /dev/sda
# This option loads the kernel and initrd much faster:
compact

# Boot BMP Image.
# Bitmap in BMP format: 640x480x8
bitmap = /boot/slack.bmp
# Menu colors (foreground, background, shadow, highlighted
# foreground, highlighted background, highlighted shadow):
bmp-colors = 255,0,255,0,255,0
# Location of the option table: location x, location y, number of
# columns, lines per column (max 15), "spill" (this is how many
# entries must be in the first column before the next begins to
# be used. We don't specify it here, as there's just one column.
bmp-table = 60,6,1,16
# Timer location x, timer location y, foreground color,
# background color, shadow color.
bmp-timer = 65,27,0,255

# Standard menu.
# Or, you can comment out the bitmap menu above and
# use a boot message with the standard menu:
#message = /boot/boot_message.txt

# Wait until the timeout to boot (if commented out, boot the
# first entry immediately):
prompt
# Timeout before the first entry boots.
# This is given in tenths of a second, so 600 for every minute:
timeout = 300
# Override dangerous defaults that rewrite the partition table:
[ Wczytano 75 linii ]
^G Pomoc      ^O Zapisz      ^W Wyszukaj    ^K Wytnij     ^T Wykonaj    ^C Lokalizacja M-U Odwołaj
^X Wyjdź      ^R Wczyt.plik ^\ Zastąp     ^U Wklej      ^J Wyjustuj   ^/ Do linii    M-E Odtwórz
```

Rysunek 25. Plik konfiguracyjny bootloadera lilo



```
GNU nano 6.0 /etc/lilo.conf Zmieniony
# VESA framebuffer console @ 1024x768x64k
vga=791
# VESA framebuffer console @ 1024x768x32k
#vga=790
# VESA framebuffer console @ 1024x768x256
#vga=773
# VESA framebuffer console @ 800x600x64k
#vga=788
# VESA framebuffer console @ 800x600x32k
#vga=787
# VESA framebuffer console @ 800x600x256
#vga=771
# VESA framebuffer console @ 640x480x64k
#vga=785
# VESA framebuffer console @ 640x480x32k
#vga=784
# VESA framebuffer console @ 640x480x256
#vga=769
# End LILO global section
# Linux bootable partition config begins
image = /boot/vmlinuz
root = /dev/sdal
label = "Slackware 15.0"
read-only

image = /boot/vmlinuz-custom-5.17.8-smp
root = /dev/sdal
initrd = /boot/initrd-custom-5.17.8-smp.gz
label = "kernel-custom"
read-only

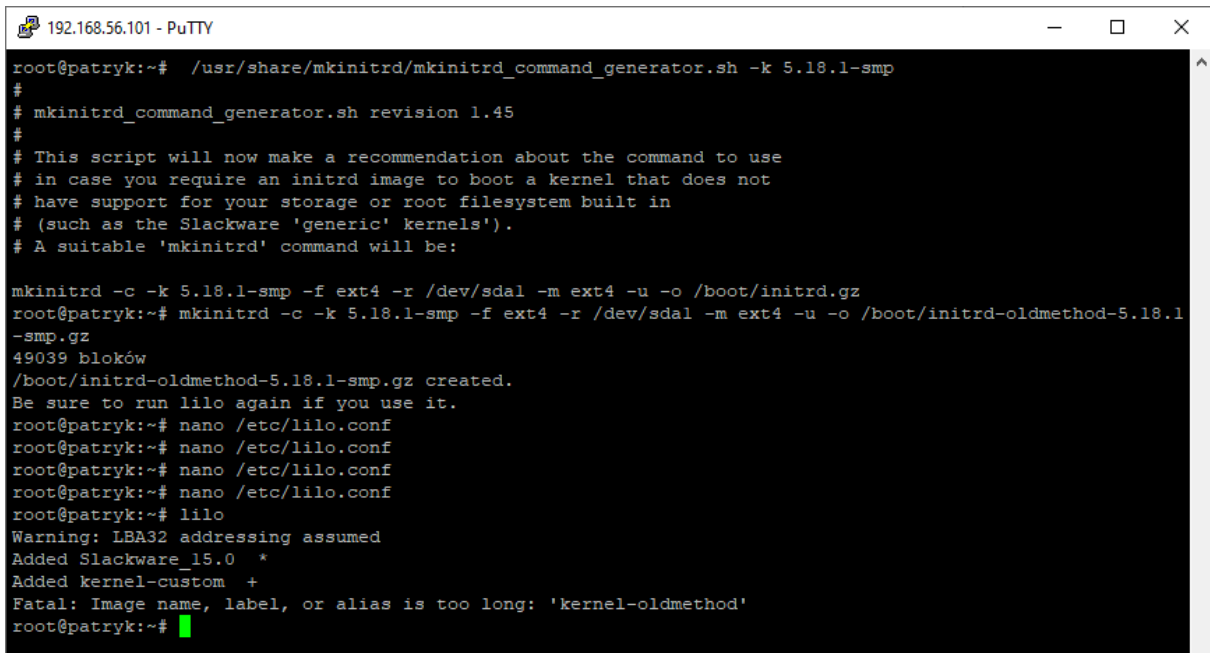
image = /boot/vmlinuz-oldmethod-5.18.1-smp
root = /dev/sdal
initrd = /boot/initrd-oldmethod-5.18.1-smp.gz
label = "kernel-oldmethod"
read-only
# Linux bootable partition config ends

^G Pomoc      ^O Zapisz    ^W Wyszukaj  ^K Wytnij   ^T Wykonaj  ^C Lokalizacja M-U Odwołaj
^X Wyjdź     ^R Wczyt.plik ^\ Zastąp   ^U Wklej    ^J Wyjustuj ^/ Do linii  M-E Odtwórz
```

Rysunek 26. Dodanie wpisu do bootloadera

Wywołanie komendy lilo:

- lilo

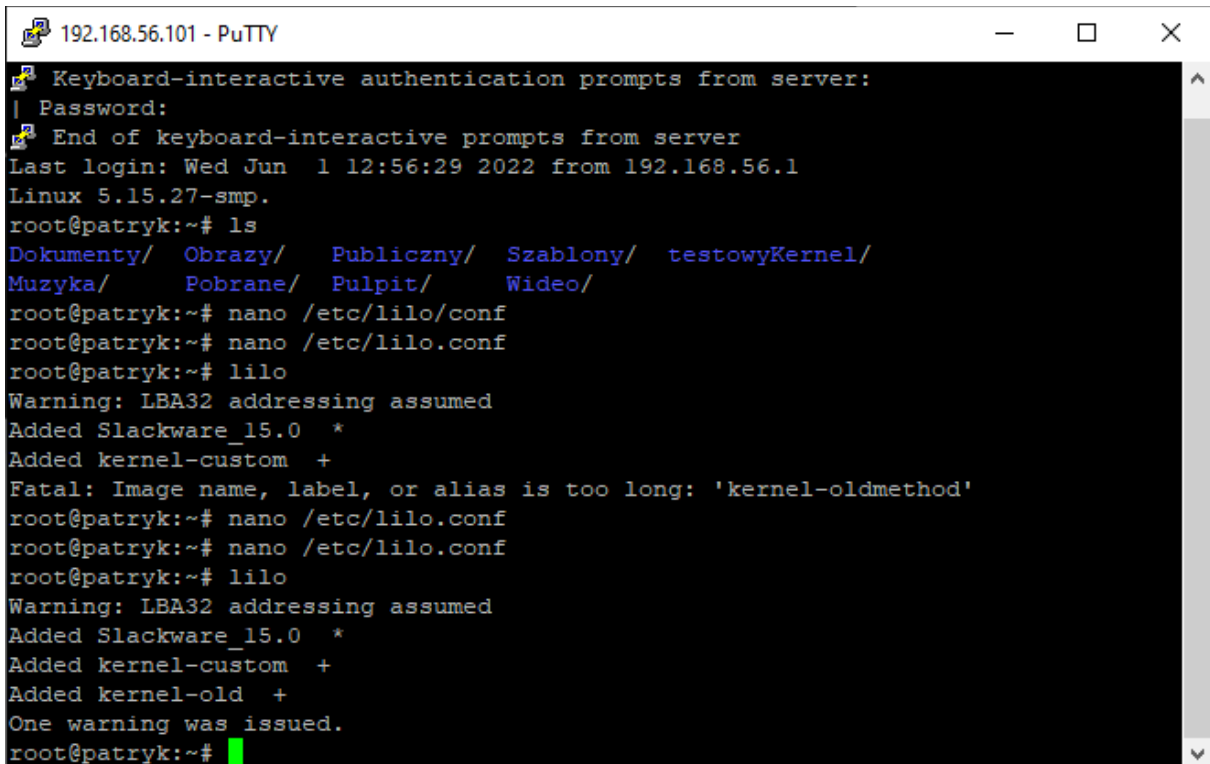


```
192.168.56.101 - PuTTY
root@patryk:~# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.18.1-smp
#
# mkinitrd_command_generator.sh revision 1.45
#
# This script will now make a recommendation about the command to use
# in case you require an initrd image to boot a kernel that does not
# have support for your storage or root filesystem built in
# (such as the Slackware 'generic' kernels').
# A suitable 'mkinitrd' command will be:

mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sdal -m ext4 -u -o /boot/initrd.gz
root@patryk:~# mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sdal -m ext4 -u -o /boot/initrd-oldmethod-5.18.1-smp.gz
49039 bloków
/boot/initrd-oldmethod-5.18.1-smp.gz created.
Be sure to run lilo again if you use it.
root@patryk:~# nano /etc/lilo.conf
root@patryk:~# nano /etc/lilo.conf
root@patryk:~# nano /etc/lilo.conf
root@patryk:~# nano /etc/lilo.conf
root@patryk:~# lilo
Warning: LBA32 addressing assumed
Added Slackware_15.0 *
Added kernel-custom +
Fatal: Image name, label, or alias is too long: 'kernel-oldmethod'
root@patryk:~#
```

Rysunek 27. Wywołanie komendy lilo

Jak widać na powyższym obrazku, nazwa jest zbyt długa więc należy ją zmienić.



```
192.168.56.101 - PuTTY
Keyboard-interactive authentication prompts from server:
| Password:
End of keyboard-interactive prompts from server
Last login: Wed Jun  1 12:56:29 2022 from 192.168.56.1
Linux 5.15.27-smp.
root@patryk:~# ls
Dokumenty/  Obrazy/    Publiczny/ Szablony/  testowyKernel/
Muzyka/     Pobrane/  Pulpit/    Wideo/
root@patryk:~# nano /etc/lilo.conf
root@patryk:~# nano /etc/lilo.conf
root@patryk:~# lilo
Warning: LBA32 addressing assumed
Added Slackware_15.0 *
Added kernel-custom +
Fatal: Image name, label, or alias is too long: 'kernel-oldmethod'
root@patryk:~# nano /etc/lilo.conf
root@patryk:~# nano /etc/lilo.conf
root@patryk:~# lilo
Warning: LBA32 addressing assumed
Added Slackware_15.0 *
Added kernel-custom +
Added kernel-old +
One warning was issued.
root@patryk:~#
```

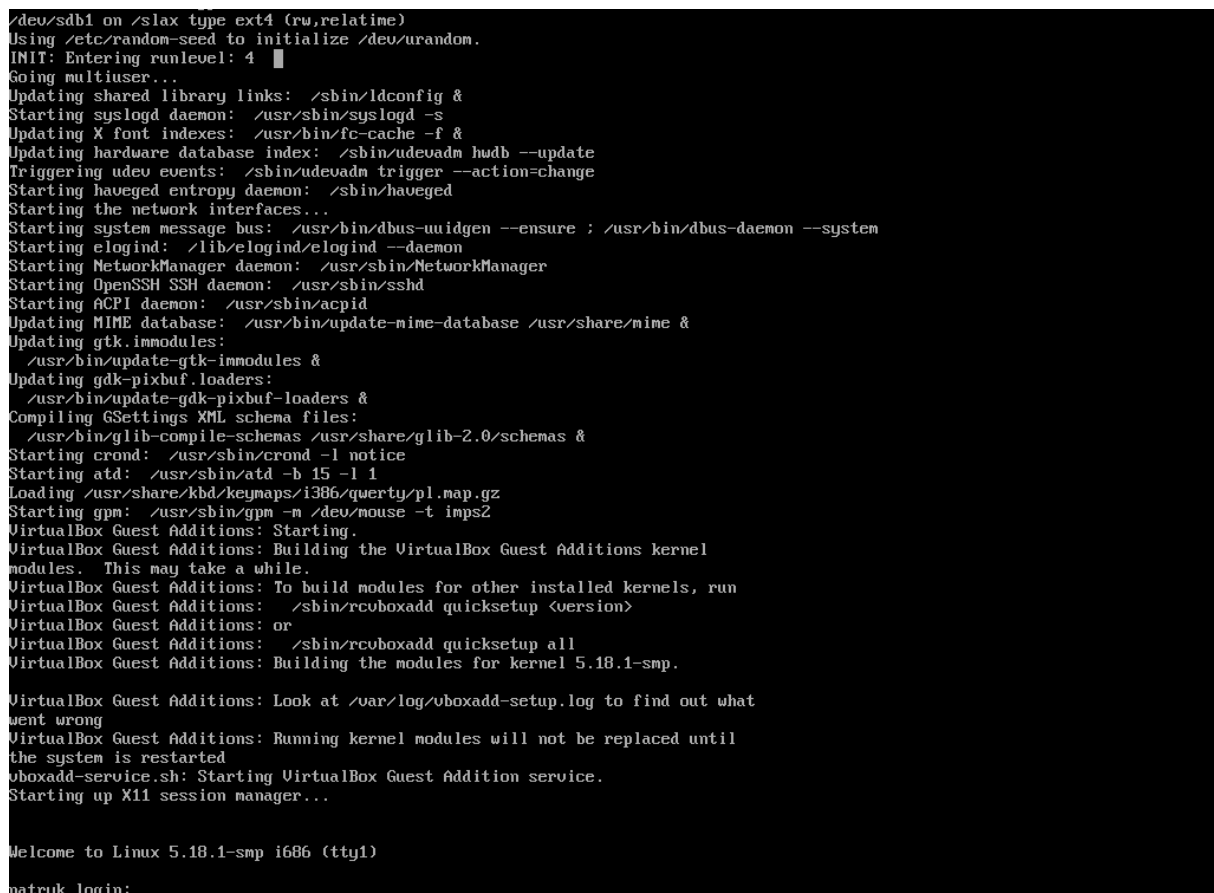
Rysunek 28. Rezultat zmiany nazwy

Następnie trzeba wykonać restart za pomocą komendy:

- reboot



Rysunek 29. Widok LiLo z widoczną opcją kernel-old



Rysunek 30. Uruchomienie kernel-old

3. Konfiguracja **scripts/kconfig/streamline_config.pl** (metoda nowa)

W celu przygotowania rozpakuję ponownie archiwium z kernelem i zmieniam mu nazwę.

```
root@patryk:/usr/src# mv linux-5.18.1 linux-5.18.1-new
root@patryk:/usr/src# ls
linux-5.17.8/  linux-5.17.8.tar.xz  linux-5.18.1-new/  linux-5.18.1-old/  linux-5.18.1.tar.xz
root@patryk:/usr/src#
```

Rysunek 31. Zmiana nazwy oraz pliki w katalogu /usr/src

Następnie przechodzę do katalogu oraz kopiuję konfigurację z starego jądra:

- Cd linux-5-18-1-new
- Zcat /proc/config.gz > .config

W przypadku nowej metody początkowym etapem będzie użycie polecenia

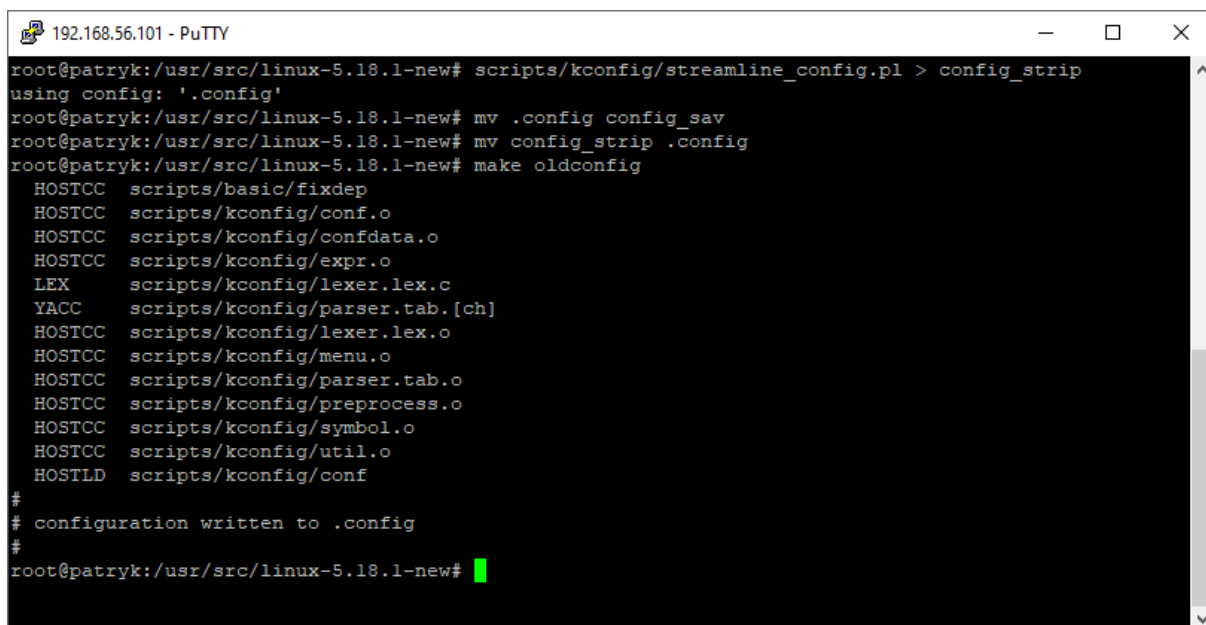
- nano scripts/kconfig/streamline_config.pl

```
GNU nano 6.0          scripts/kconfig/streamline_config.pl
#!/usr/bin/env perl
# SPDX-License-Identifier: GPL-2.0
#
# Copyright 2005-2009 - Steven Rostedt
#
# It's simple enough to figure out how this works.
# If not, then you can ask me at stripconfig@goodmis.org
#
# What it does?
#
# If you have installed a Linux kernel from a distribution
# that turns on way too many modules than you need, and
# you only want the modules you use, then this program
# is perfect for you.
#
# It gives you the ability to turn off all the modules that are
# not loaded on your system.
#
# Howto:
#
# 1. Boot up the kernel that you want to stream line the config on.
# 2. Change directory to the directory holding the source of the
#    kernel that you just booted.
# 3. Copy the configuration file to this directory as .config
# 4. Have all your devices that you need modules for connected and
#    operational (make sure that their corresponding modules are loaded)
# 5. Run this script redirecting the output to some other file
#    like config_strip.
# 6. Back up your old config (if you want too).
# 7. copy the config_strip file to .config
# 8. Run "make oldconfig"
#
# Now your kernel is ready to be built with only the modules that
# are loaded.
#
# 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
```

Rysunek 32. Wykonanie powyższej komendy

Następnym krokiem, który został wykonany zgodnie z instrukcją to:

- `scripts/kconfig/streamline_config.pl > config_strip`
- `mv .config config_sav`
- `mv config_strip .config`
- `make oldconfig`

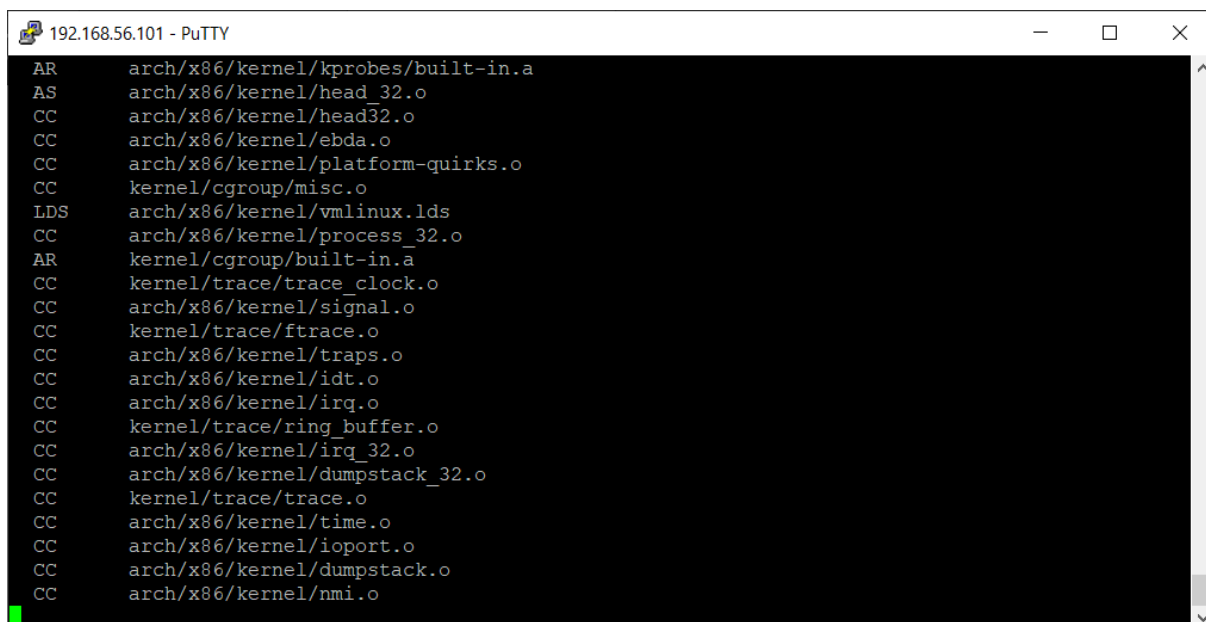


```
192.168.56.101 - PuTTY
root@patryk:/usr/src/linux-5.18.1-new# scripts/kconfig/streamline_config.pl > config_strip
using config: '.config'
root@patryk:/usr/src/linux-5.18.1-new# mv .config config_sav
root@patryk:/usr/src/linux-5.18.1-new# mv config_strip .config
root@patryk:/usr/src/linux-5.18.1-new# make oldconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o
LEX scripts/kconfig/lexer.lex.c
YACC scripts/kconfig/parser.tab.[ch]
HOSTCC scripts/kconfig/lexer.lex.o
HOSTCC scripts/kconfig/menu.o
HOSTCC scripts/kconfig/parser.tab.o
HOSTCC scripts/kconfig/preprocess.o
HOSTCC scripts/kconfig/symbol.o
HOSTCC scripts/kconfig/util.o
HOSTLD scripts/kconfig/conf
#
# configuration written to .config
#
root@patryk:/usr/src/linux-5.18.1-new#
```

Rysunek 33. Wykonanie powyższych instrukcji

Następnym elementem jest budowa obrazu jądra jak i reszta etapów analogicznych jak w przypadku poprzednim.

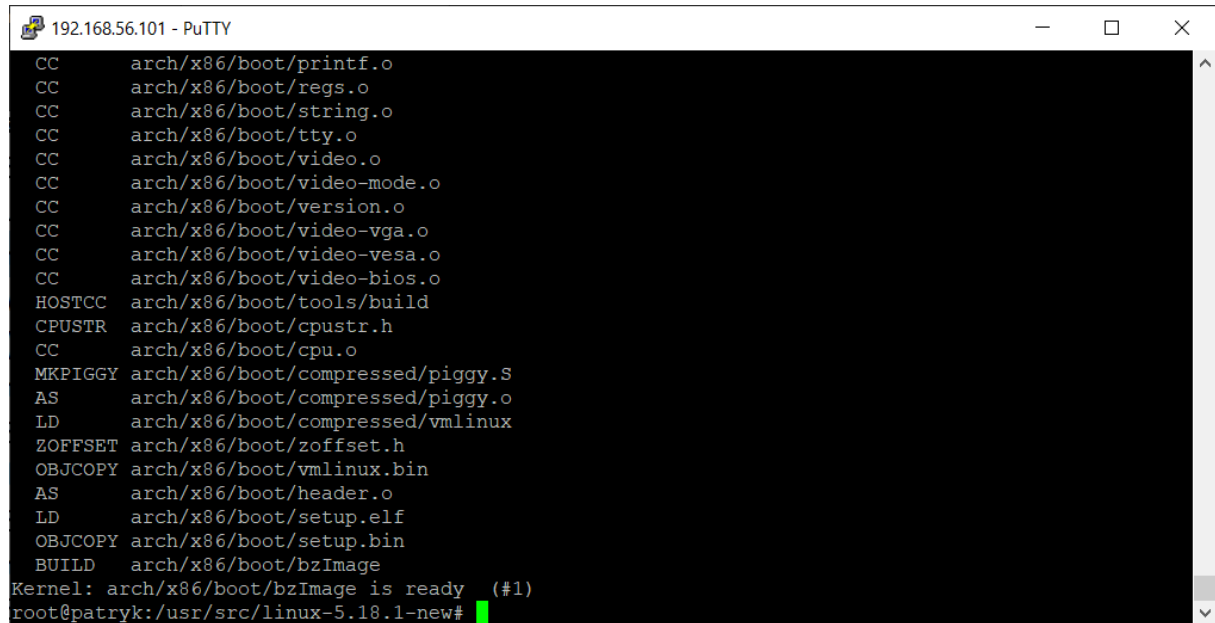
Wykonanie budowy obrazu jądra rozpoczęło się o godzinie **16:45**



```
192.168.56.101 - PuTTY
AR      arch/x86/kernel/kprobes/built-in.a
AS      arch/x86/kernel/head_32.o
CC      arch/x86/kernel/head32.o
CC      arch/x86/kernel/ebda.o
CC      arch/x86/kernel/platform-quirks.o
CC      kernel/cgroup/misc.o
LDS      arch/x86/kernel/vmlinux.lds
CC      arch/x86/kernel/process_32.o
AR      kernel/cgroup/built-in.a
CC      kernel/trace/trace_clock.o
CC      arch/x86/kernel/signal.o
CC      kernel/trace/ftrace.o
CC      arch/x86/kernel/traps.o
CC      arch/x86/kernel/idt.o
CC      arch/x86/kernel/irq.o
CC      kernel/trace/ring_buffer.o
CC      arch/x86/kernel/irq_32.o
CC      arch/x86/kernel/dumpstack_32.o
CC      kernel/trace/trace.o
CC      arch/x86/kernel/time.o
CC      arch/x86/kernel/ioport.o
CC      arch/x86/kernel/dumpstack.o
CC      arch/x86/kernel/nmi.o
```

Rysunek 34. Proces przebiegu budowy obrazu jądra

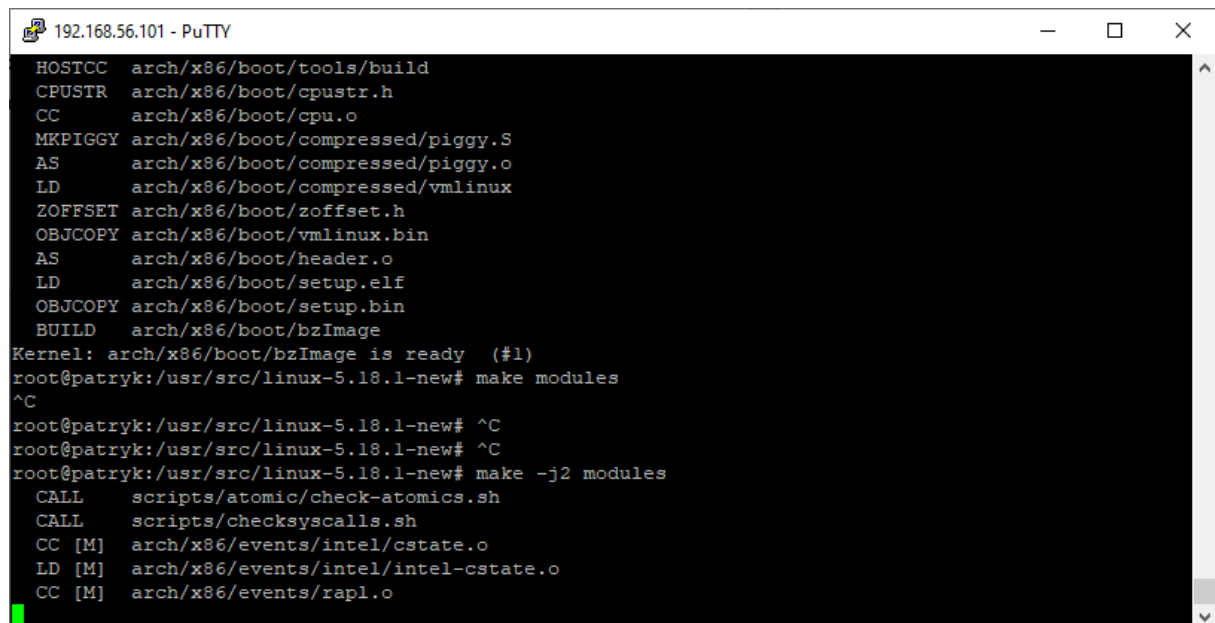
Budowa obrazu jądra zakończyła się o godzinie **17:40**, w związku z tym trwała 65 minut.



```
192.168.56.101 - PuTTY
CC      arch/x86/boot/printf.o
CC      arch/x86/boot/regs.o
CC      arch/x86/boot/string.o
CC      arch/x86/boot/tty.o
CC      arch/x86/boot/video.o
CC      arch/x86/boot/video-mode.o
CC      arch/x86/boot/version.o
CC      arch/x86/boot/video-vga.o
CC      arch/x86/boot/video-vesa.o
CC      arch/x86/boot/video-bios.o
HOSTCC  arch/x86/boot/tools/build
CPUSTR  arch/x86/boot/cpustr.h
CC      arch/x86/boot/cpu.o
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@patryk:/usr/src/linux-5.18.1-new#
```

Rysunek 35. Zakończenie budowy obrazu jądra

Zbudowanie modułów jądra rozpoczęło o godzinie **17:53**.



```
192.168.56.101 - PuTTY
HOSTCC  arch/x86/boot/tools/build
CPUSTR  arch/x86/boot/cpustr.h
CC      arch/x86/boot/cpu.o
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@patryk:/usr/src/linux-5.18.1-new# make modules
^C
root@patryk:/usr/src/linux-5.18.1-new# ^C
root@patryk:/usr/src/linux-5.18.1-new# ^C
root@patryk:/usr/src/linux-5.18.1-new# make -j2 modules
CALL    scripts/atomic/check-atomics.sh
CALL    scripts/checksyscalls.sh
CC [M]  arch/x86/events/intel/cstate.o
LD [M]  arch/x86/events/intel/intel-cstate.o
CC [M]  arch/x86/events/rapl.o
```

Rysunek 36. Budowa modułów jądra za pomocą komendy `make -j2 modules`

Zbudowanie modułów jądra zakończyło się o godzinie **17:58**.

```
192.168.56.101 - PuTTY
LD [M] drivers/powercap/intel_rapl_msr.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] drivers/virt/vboxguest/vboxguest.ko
LD [M] net/802/garp.ko
LD [M] net/802/mrp.ko
LD [M] net/802/p8022.ko
LD [M] net/802/psnap.ko
LD [M] net/802/stp.ko
LD [M] net/8021q/8021q.ko
LD [M] net/ipv6/ipv6.ko
LD [M] net/llc/llc.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/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@patryk:/usr/src/linux-5.18.1-new# y
```

Rysunek 37. Zakończenie budowy modułów

Następnym etapem jest instalacja modułów.

```
192.168.56.101 - PuTTY
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@patryk:/usr/src/linux-5.18.1-new# make modules_install
INSTALL /lib/modules/5.18.1-smp/kernel/arch/x86/events/intel-cstate.ko
INSTALL /lib/modules/5.18.1-smp/kernel/arch/x86/events/rapl.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/acpi/ac.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/acpi/battery.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/acpi/button.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/acpi/video.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/block/loop.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/char/agp/agpgart.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/char/agp/intel-agp.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/char/agp/intel-gtt.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/drm.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/drm_kms_helper.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/drm_ttm_helper.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/drm_vram_helper.ko
INSTALL /lib/modules/5.18.1-smp/kernel/drivers/gpu/drm/ttm/ttm.ko
```

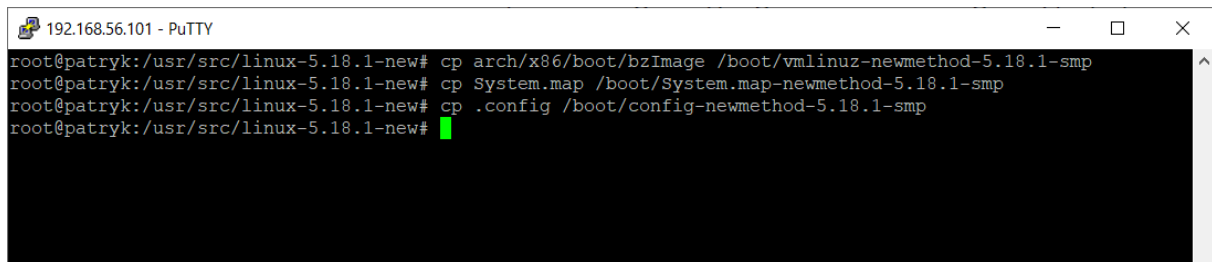
Rysunek 38. Wykonanie komendy `make modules_install`

```
192.168.56.101 - PuTTY
root@patryk:/usr/src/linux-5.18.1-new# ls /lib/modules/5.18.1-smp/
build@      modules.builtin      modules.dep      modules.softdep
kernel/     modules.builtin.alias.bin  modules.dep.bin  modules.symbols
modules.alias  modules.builtin.bin  modules.devname  modules.symbols.bin
modules.alias.bin  modules.builtin.modinfo  modules.order    source@
root@patryk:/usr/src/linux-5.18.1-new#
```

Rysunek 39. Lista zainstalowanych modułów

Analogicznie jak w poprzedniej konfiguracji następnym etapem jest przekopiowanie jądra do katalogu boot, tablicy symboli używanych przez kernela oraz pliku konfiguracyjnego

- `cp arch/x86/boot/bzImage /boot/vmlinuz-newmethod-5.18.1-smp`
- `cp System.map /boot/System.map-newmethod-5.18.1-smp`
- `cp .config /boot/config-newmethod-5.18.1-smp`



```
192.168.56.101 - PuTTY
root@patryk:/usr/src/linux-5.18.1-new# cp arch/x86/boot/bzImage /boot/vmlinuz-newmethod-5.18.1-smp
root@patryk:/usr/src/linux-5.18.1-new# cp System.map /boot/System.map-newmethod-5.18.1-smp
root@patryk:/usr/src/linux-5.18.1-new# cp .config /boot/config-newmethod-5.18.1-smp
root@patryk:/usr/src/linux-5.18.1-new#
```

Rysunek 40. Wykonanie powyższych komend

Tak samo jak w przypadku poprzednim należy utworzyć link symboliczny dla tablicy symboli kernego w tym celu należy przejść do katalogu *boot*:

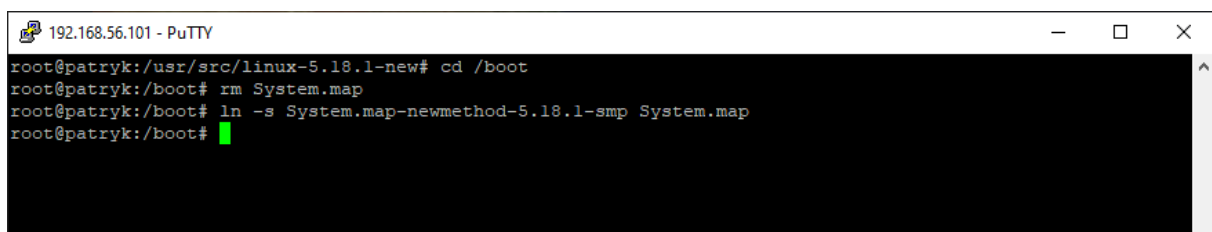
- `cd /boot`

Usunąć starą tablicę symboli:

- `rm System.map`

Utworzyć link symboliczny:

- `ln -s System.map-newmethod-5.18.1-smp System.map`



```
192.168.56.101 - PuTTY
root@patryk:/usr/src/linux-5.18.1-new# cd /boot
root@patryk:/boot# rm System.map
root@patryk:/boot# ln -s System.map-newmethod-5.18.1-smp System.map
root@patryk:/boot#
```

Rysunek 41. Wykonanie powyższych komend

Utworzenie dysku RAM

- `/usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.18.1-smp`
- `mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd-newmethod-5.18.1-smp.gz`

```
192.168.56.101 - PuTTY
root@patryk:/boot# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.18.1-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:
mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sdal -m ext4 -u -o /boot/initrd.gz
root@patryk:/boot# ^C
root@patryk:/boot# mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sdal -m ext4 -u -o /boot/initrd-newmethod-5.18.1-smp.gz
49039 bloków
/boot/initrd-newmethod-5.18.1-smp.gz created.
Be sure to run lilo again if you use it.
root@patryk:/boot#
```

Rysunek 42. Wykonanie powyższych komend

Następnym etapem jest dodanie konfiguracji do bootloadera lilo

- nano /etc/lilo.conf

```
192.168.56.101 - PuTTY
GNU nano 6.0 /etc/lilo.conf
label = "Slackware 15.0"
read-only

image = /boot/vmlinuz-custom-5.17.8-smp
root = /dev/sdal
initrd = /boot/initrd-custom-5.17.8-smp.gz
label = "kernel-custom"
read-only

image = /boot/vmlinuz-oldmethod-5.18.1-smp
root = /dev/sdal
initrd = /boot/initrd-oldmethod-5.18.1-smp.gz
label = "kernel-old"
read-only

image = /boot/vmlinuz-newmethod-5.18.1-smp
root = /dev/sdal
initrd = /boot/initrd-newmethod-5.18.1-smp.gz
label = "kernel-new"
read-only

[ Zapisano 87 linii ]
^G Pomoc      ^O Zapisz     ^W Wyszukaj   ^K Wytnij    ^T Wykonaj   ^C Lokalizacja M-U Odwołaj
^X Wyjdź      ^R Wczyt.plik ^\ Zastap    ^U Wklej     ^J Wyjustuj  ^/ Do linii  M-E Odtwórz
```

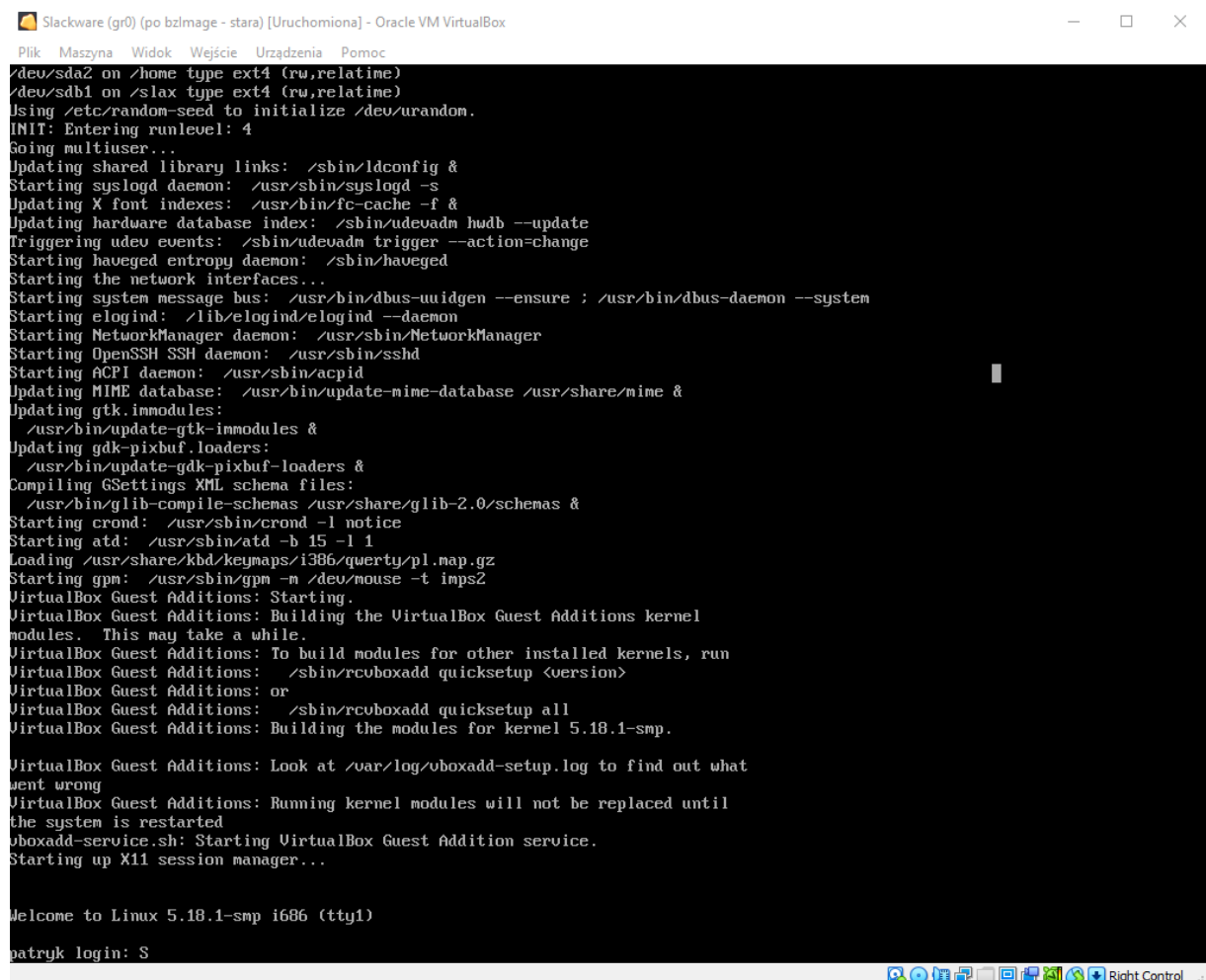
Rysunek 43. Dodanie wpisu do bootloadera

```
192.168.56.101 - PuTTY
root@patryk:/boot# lilo
Warning: LBA32 addressing assumed
Added Slackware_15.0 *
Added kernel-custom +
Added kernel-old +
Added kernel-new +
One warning was issued.
root@patryk:/boot#
```

Rysunek 44. Wykonanie komendy lilo



Rysunek 45. Ponowne uruchomienie obrazu maszyny wirtualnej skutkuje dodaniem nowego wpisu `kernel-new`



```
Slackware (gr0) (po bzImage - stara) [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
/dev/sda2 on /home type ext4 (rw,relatime)
/dev/sdb1 on /slax type ext4 (rw,relatime)
Using /etc/random-seed to initialize /dev/urandom.
INIT: Entering runlevel: 4
Going multiuser...
Updating shared library links: /sbin/ldconfig &
Starting syslogd daemon: /usr/sbin/syslogd -s
Updating X font indexes: /usr/bin/fc-cache -f &
Updating hardware database index: /sbin/udevadm hwdb --update
Triggering udev events: /sbin/udevadm trigger --action=change
Starting haveged entropy daemon: /sbin/haveged
Starting the network interfaces...
Starting system message bus: /usr/bin/dbus-uuidgen --ensure ; /usr/bin/dbus-daemon --system
Starting elogind: /lib/elogind/elogind --daemon
Starting NetworkManager daemon: /usr/sbin/NetworkManager
Starting OpenSSH SSH daemon: /usr/sbin/sshd
Starting ACPI daemon: /usr/sbin/acpid
Updating MIME database: /usr/bin/update-mime-database /usr/share/mime &
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 &
Starting crond: /usr/sbin/crond -l notice
Starting atd: /usr/sbin/atd -b 15 -l 1
Loading /usr/share/kbd/keymaps/i386/qwerty/pl.map.gz
Starting gpm: /usr/sbin/gpm -m /dev/mouse -t imps2
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 5.18.1-smp.
VirtualBox Guest Additions: Look at /var/log/vboxadd-setup.log to find out what
went wrong
VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted
vboxadd-service.sh: Starting VirtualBox Guest Addition service.
Starting up X11 session manager...

Welcome to Linux 5.18.1-smp i686 (tty1)
patryk login: S
```

Rysunek 46. Uruchomienie kernel-new

4. Wnioski

Odczucia z przebiegu kompilacji kernela mam pozytywne pomimo błędów które napotkałem – brak dostępnego miejsca itd. Metoda nowsza była możliwa do zrealizowania w sposób bardziej przyjazny, ponieważ przedstawione kroki w pliku scripts/kconfig/streamline_config.pl zostały dobrze opisane wraz z podanym przykładem. Nowsza metoda jednak wykonywała się trochę dłużej niż metoda stara, a dokładniej budowanie obrazu jądra w przypadku konfiguracji z metody nowej trwało 65 minut, natomiast w przypadku konfiguracji starej 33 minuty.