

# Kompilacja jądra linux

## Historia o jądrze istniejącym w przyszłości

Filip Ręka

Czerwiec 2022

## Pobranie najnowszych źródeł kernela

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

```
root@slack:~# cd /usr/src
root@slack:/usr/src# ls
root@slack:/usr/src# wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.18.1.tar.xz
--2022-05-05 16:51:42-- https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.18.1.tar.xz
Translacja cdn.kernel.org (cdn.kernel.org)... 151.101.13.176, 2a04:4e42:3::432
Łączenie się z cdn.kernel.org (cdn.kernel.org)[151.101.13.176]:443... połączono.
Żądanie HTTP wysłano, oczekiwanie na odpowiedź... 200 OK
Długość: 129795536 (124M) [application/x-xz]
Zapis do: 'linux-5.18.1.tar.xz'

linux-5.18.1.tar.xz      100%[=====] 123,78M  4,66MB/s   w 27s

2022-05-05 16:52:10 (4,57 MB/s) - zapisano 'linux-5.18.1.tar.xz' [129795536/129795536]

root@slack:/usr/src# ls
linux-5.18.1.tar.xz
root@slack:/usr/src# |
```

## Rozpakowanie pliku archiwum z źródłami kernela

- `tar -xvpf linux-5.18.1.tar.xz`

Podczas wywołania komendy dla każdego wypakowanego pliku wyskakiwały takie powiadomienia

```
tar: linux-5.18.1/include/dt-bindings: znacznik czasowy 2022-05-30 09:24:09 jest przesunięty o 2132672.034094676 s w przyszłość
tar: linux-5.18.1/include: znacznik czasowy 2022-05-30 09:24:09 jest przesunięty o 2132672.033953596 s w przyszłość
tar: linux-5.18.1/arch/arm64/boot/dts/arm: znacznik czasowy 2022-05-30 09:24:09 jest przesunięty o 2132672.032792834 s w przyszłość
tar: linux-5.18.1/arch/arm64/boot/dts: znacznik czasowy 2022-05-30 09:24:09 jest przesunięty o 2132672.032618231 s w przyszłość
tar: linux-5.18.1/arch/arm64/boot: znacznik czasowy 2022-05-30 09:24:09 jest przesunięty o 2132672.032541685 s w przyszłość
tar: linux-5.18.1/arch/arm64: znacznik czasowy 2022-05-30 09:24:09 jest przesunięty o 2132672.032076263 s w przyszłość
tar: linux-5.18.1/arch: znacznik czasowy 2022-05-30 09:24:09 jest przesunięty o 2132672.031984911 s w przyszłość
tar: linux-5.18.1: znacznik czasowy 2022-05-30 09:24:09 jest przesunięty o 2132672.031873164 s w przyszłość
root@slack:/usr/src# ls
linux-5.18.1/ linux-5.18.1.tar.xz
root@slack:/usr/src# |
```

Mam nadzieję, że nie będzie to miało wpływu na dalszy proces

## Stara metoda

### Wywołanie polecenia do utworzenia pliku config

- `cd linux-5.18.1`
- `zcat /proc/config.gz >.config`
- `make localmodconfig`

Po wielu naciśnięciach przycisku Enter udało się dotrzeć do końca, jednak dalej pojawiają się problemy z zegarem. Na razie nic nie ruszam, ponieważ programiści nie przejmują się warningami.

```
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
#
make[1]: uwaga: Wykryto przestawienie zegara. Budowanie może być niekompletne.
make: uwaga: Wykryto przestawienie zegara. Budowanie może być niekompletne.
root@slack:/usr/src/linux-5.18.1#
```

### Zbudowanie pliku konfiguracyjnego kernela z konfiguracją z poprzedniego kernela

- `make olddefconfig`

Zegar znowu okazuje się problemem jednak ja dalej i nieustannie zamierzam go ignorować (podobno `touch *` może pomóc)

```
root@slack:/usr/src/linux-5.18.1# make olddefconfig
make: Uwaga: Plik 'arch/x86/Makefile' ma czas modyfikacji 2131548 s w przyszłości
make[1]: Uwaga: Plik 'scripts/Makefile.host' ma czas modyfikacji 2131548 s w przyszłości
HOSTCC scripts/basic/fixdep
make[1]: uwaga: Wykryto przestawienie zegara. Budowanie może być niekompletne.
make[1]: Uwaga: Plik 'scripts/Makefile.host' ma czas modyfikacji 2131547 s w przyszłości
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
#
# No change to .config
#
make[1]: uwaga: Wykryto przestawienie zegara. Budowanie może być niekompletne.
make: uwaga: Wykryto przestawienie zegara. Budowanie może być niekompletne.
```

## Kompilacja obrazu jądra

- `make -j2 bzImage`

Jak jednak się okazało programista powinien zwracać uwagę na warningi, ponieważ, czego nie zauważyłem przez 4 godziny, powtarzała się ta sekwencja:

```

SYNC      include/config/auto.conf.cmd
make[1]: Uwaga: Plik 'arch/x86/Makefile' ma czas modyfikacji 2118330 s w przyszłości
make[2]: Uwaga: Plik 'scripts/Makefile.host' ma czas modyfikacji 2118330 s w przyszłości
HOSTCC    scripts/basic/fixdep
make[2]: uwaga: Wykryto przestawienie zegara. Budowanie może być niekompletne.
make[2]: Uwaga: Plik 'scripts/Makefile.host' ma czas modyfikacji 2118330 s w przyszłości
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/menu.o
HOSTCC    scripts/kconfig/parser.tab.o
HOSTCC    scripts/kconfig/preprocess.o
HOSTCC    scripts/kconfig/symbol.o
HOSTCC    scripts/kconfig/util.o
HOSTCC    scripts/kconfig/lexer.lex.o
HOSTLD    scripts/kconfig/conf
make[2]: uwaga: Wykryto przestawienie zegara. Budowanie może być niekompletne.
make[1]: uwaga: Wykryto przestawienie zegara. Budowanie może być niekompletne.
```

Błąd prawdopodobnie wynikał z tego, że maszyna została odtworzona z migawki wykonanej kilka dni wcześniej przez co czas systemowy nie został zaktualizowany na obecny. Po ponownym uruchomieniu problem rozwiązał się. Wykonałem ponownie komendy tworzące konfigurację i tym razem wykonały się bez żadnych problemów.

Tym razem się udało

```

HOSTCC    arch/x86/boot/tools/build
CC         arch/x86/boot/compressed/kaslr.o
CC         arch/x86/boot/compressed/acpi.o
CPUTSTR    arch/x86/boot/cpustr.h
CC         arch/x86/boot/compressed/misc.o
LZMA       arch/x86/boot/compressed/vmlinux.bin.lzma
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@slack:/usr/src/linux-5.18.1#
```

## Zbudowanie modułów jądra

- `make -j2 modules`

```
LD [M] net/802/psnap.ko
LD [M] net/802/stp.ko
LD [M] net/8021q/8021q.ko
LD [M] net/llc/llc.ko
LD [M] net/rfkill/rfkill.ko
LD [M] net/ipv6/ipv6.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@slack:/usr/src/linux-5.18.1#
```

Moduły budowały się podejrzanie krótko, jednak mam nadzieję, że nic się złego nie stanie...

## Instalacja modułów

- `make modules_install`

Dla sprawdzenia czy wszystko zadziało poprawnie wykonujemy

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

```
root@slack:/usr/src/linux-5.18.1# ls /lib/modules/5.1
5.15.27/ 5.15.27-smp/ 5.18.1-smp/
root@slack:/usr/src/linux-5.18.1# 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@slack:/usr/src/linux-5.18.1#
```

## Przekopiowanie plików kernela do systemu

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

```
root@slack:/usr/src/linux-5.18.1# cp arch/x86/boot/bzImage /boot/vmlinuz-custom-5.18.1-smp
root@slack:/usr/src/linux-5.18.1# cp System.map /boot/System.map-custom-5.18.1-smp
root@slack:/usr/src/linux-5.18.1# cp .config /boot/config-custom-5.18.1-smp
```

## Utworzenie linku symbolicznego w systemie dla tablicy symboli kernela

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

```
root@slack:/boot# rm System.map
root@slack:/boot# ln -s System.map-custom-5.18.1-smp System.map
root@slack:/boot#
```

## 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-custom-5.18.1-smp.gz`

```
root@slack:/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/sda1 -m ext4 -u -o /boot/initrd.gz
root@slack:/boot# mkinitrd -c -k 5.18.1-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd-custom-5.18.1-smp.gz
49031 bloków
/boot/initrd-custom-5.18.1-smp.gz created.
Be sure to run lilo again if you use it.
root@slack:/boot#
```

## Dodanie nowego wpisy do konfiguracji bootloadera lilo

- `vim /etc/lilo.conf`

```
root@slack:/boot# vim /etc/lilo.conf
root@slack:/boot# tail /etc/lilo.conf
image = /boot/vmlinuz
  root = /dev/sda1
  label = Linux
  read-only
image = /boot/vmlinuz-custom-5.18.1-smp
  root = /dev/sda1
  initrd = /boot/initrd-custom-5.18.1-smp.gz
  label = "kernel_custom_old"
  read-only
# Linux bootable partition config ends
root@slack:/boot#
```

- lilo

```
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 Linux *
Fatal: Image name, label, or alias is too long: 'kernel_custom_old'
root@slack:/boot#
```

Error jest oczywisty do rozwiązania

```
root@slack:/boot# vim /etc/lilo.conf
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 Linux *
Added custom_old +
3 warnings were issued.
root@slack:/boot#
```

### CHWILA PRAWDY....

```
Welcome to Linux 5.18.1-smp i686 (tty1)

slack login: root
Password:
Last login: Sat Jun  4 16:59:06 from 192.168.56.1
Linux 5.18.1-smp.
root@slack:~# S_
```

Uruchomienie systemu zostało wykonane poprawnie, system działa jak trzeba, nic dodać nic ująć.

## Nowa metoda

W między czasie pojawiła się nowa wersja kernela (5.18.2), więc będziemy teraz pracowali na niej.

- `cd /usr/src`
- `wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.18.2.tar.xz`

```
root@slack:/usr/src# wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.18.2.tar.xz
--2022-06-08 20:19:59-- https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.18.2.tar.xz
Translacja cdn.kernel.org (cdn.kernel.org)... 151.101.13.176, 2a04:4e42:3::432
Łączenie się z cdn.kernel.org (cdn.kernel.org)[151.101.13.176]:443... połączono.
Żądanie HTTP wysłano, oczekiwanie na odpowiedź... 200 OK
Długość: 129796020 (124M) [application/x-xz]
Zapis do: 'linux-5.18.2.tar.xz'

linux-5.18.2.tar.xz      100%[=====] 123,78M  4,07MB/s   w 31s
2022-06-08 20:20:31 (3,95 MB/s) - zapisano 'linux-5.18.2.tar.xz' [129796020/129796020]
root@slack:/usr/src# |
```

Rozpakowywanie plików przeszło pomyślnie i nie przysporzyło żadnych problemów.

- `tar -xvpf linux-5.18.2.tar.xz`

```
linux-5.18.2/virt/kvm/coalesced_mmio.h
linux-5.18.2/virt/kvm/dirty_ring.c
linux-5.18.2/virt/kvm/eventfd.c
linux-5.18.2/virt/kvm/irqchip.c
linux-5.18.2/virt/kvm/kvm_main.c
linux-5.18.2/virt/kvm/kvm_mm.h
linux-5.18.2/virt/kvm/pfncache.c
linux-5.18.2/virt/kvm/vfio.c
linux-5.18.2/virt/kvm/vfio.h
linux-5.18.2/virt/lib/
linux-5.18.2/virt/lib/Kconfig
linux-5.18.2/virt/lib/Makefile
linux-5.18.2/virt/lib/irqbypass.c
root@slack:/usr/src# |
```

Po wypakowaniu wchodzimy do folderu, kopiujemy config, a następnie uruchamiamy skrypt o nazwie `streamline_config.pl`, którego wyjście przekazujemy do pliku `config_strip`

- `cd linux-5.18.2`
- `cp /boot/config .config`
- `./scripts/kconfig/streamline_config.pl > config_strip`

```
root@slack:/usr/src/linux-5.18.2# ./scripts/kconfig/streamline_config.pl > config_strip
using config: '.config'
root@slack:/usr/src/linux-5.18.2#
```



Następnie dalej postępujemy zgodnie z instrukcją i wykonujemy następujące polecenia:

- `mv .config config_sav`
- `mv config_strip .config`
- `make oldconfig`

Znowu klikamy sporo razy Enter.

```
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_VMALLOC) [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@slack:/usr/src/linux-5.18.2#
```

Następnie wykonujemy komendę

- `make`

```
CC [M] sound/core/snd-timer.mod.o
LD [M] sound/core/snd-timer.ko
CC [M] sound/core/snd.mod.o
LD [M] sound/core/snd.ko
CC [M] sound/pci/ac97/snd-ac97-codec.mod.o
LD [M] sound/pci/ac97/snd-ac97-codec.ko
CC [M] sound/pci/snd-intel8x0.mod.o
LD [M] sound/pci/snd-intel8x0.ko
CC [M] sound/soundcore.mod.o
LD [M] sound/soundcore.ko
root@slack:/usr/src/linux-5.18.2# |
```

potem

- `make modules`

```
root@slack:/usr/src/linux-5.18.2# make -j2 modules
CALL scripts/atomic/check-atomics.sh
CALL scripts/checksyscalls.sh
root@slack:/usr/src/linux-5.18.2# |
```

podejrzenie mało się wyświetliło ale chyba jest ok...

i na koniec

- `make modules_install`

```
INSTALL /lib/modules/5.18.2-smp/kernel/net/ipv6/ipv6.ko
INSTALL /lib/modules/5.18.2-smp/kernel/net/llc/llc.ko
INSTALL /lib/modules/5.18.2-smp/kernel/net/rfkill/rfkill.ko
INSTALL /lib/modules/5.18.2-smp/kernel/net/wireless/cfg80211.ko
INSTALL /lib/modules/5.18.2-smp/kernel/sound/ac97_bus.ko
INSTALL /lib/modules/5.18.2-smp/kernel/sound/core/snd-pcm.ko
INSTALL /lib/modules/5.18.2-smp/kernel/sound/core/snd-timer.ko
INSTALL /lib/modules/5.18.2-smp/kernel/sound/core/snd.ko
INSTALL /lib/modules/5.18.2-smp/kernel/sound/pci/ac97/snd-ac97-codec.ko
INSTALL /lib/modules/5.18.2-smp/kernel/sound/pci/snd-intel8x0.ko
INSTALL /lib/modules/5.18.2-smp/kernel/sound/soundcore.ko
DEPMOD /lib/modules/5.18.2-smp
root@slack:/usr/src/linux-5.18.2#
```

dla sprawdzenia można wykonać

- `ls /lib/modules/5.18.2-smp/`

```
root@slack:/usr/src/linux-5.18.2# ls /lib/modules/5.18.2-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@slack:/usr/src/linux-5.18.2#
```

Przekopiowanie plików kernela do systemu wygląda tak samo jak za pierwszym razem:

- `cp arch/x86/boot/bzImage /boot/vmlinuz-custom-5.18.2-smp`
- `cp System.map /boot/System.map-custom-5.18.2-smp`
- `cp .config /boot/config-custom-5.18.2-smp`

Tworzymy link symboliczny:

- `cd /boot/`
- `rm System.map`
- `ln -s System.map-custom-5.18.2-smp System.map`

```
root@slack:/usr/src/linux-5.18.2# cd /boot/
root@slack:/boot# rm System.map
root@slack:/boot# ln -s System.map-custom-5.18.2-smp System.map
root@slack:/boot#
```

Tworzymy ram disk tak jak w pierwszej metodzie przy pomocy tych poleceń:

- `/usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.18.2-smp`
- `mkinitrd -c -k 5.18.2-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd-custom-5.18.2-smp.gz`

```
root@slack:/boot# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.18.2-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 5.18.2-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd.gz
root@slack:/boot# mkinitrd -c -k 5.18.2-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd-custom-5.18.2-smp.gz
49031 bloków
/boot/initrd-custom-5.18.2-smp.gz created.
Be sure to run lilo again if you use it.
root@slack:/boot#
```

Dodajemy nowy wpis do lilo:

- `vim /etc/lilo.conf`

```
root@slack:/boot# vim /etc/lilo.conf
root@slack:/boot# tail /etc/lilo.conf
root = /dev/sda1
initrd = /boot/initrd-custom-5.18.1-smp.gz
label = "custom_old"
read-only
image = /boot/vmlinuz-custom-5.18.2-smp
root = /dev/sda1
initrd = /boot/initrd-custom-5.18.2-smp.gz
label = "custom_new"
read-only
# Linux bootable partition config ends
root@slack:/boot#
```

Wykonujemy polecenie:

- `lilo`

```
root@slack:/boot# lilo
Warning: LBA32 addressing assumed
Added Linux *
Added custom_old +
Added custom_new +
One warning was issued.
root@slack:/boot#
```

teraz **reboot** i liczymy na to że się uda

Na szczęście wszystko przebiegło zgodnie z planem i mamy poprawnie zainstalowane jądro linuxa

```
Welcome to Linux 5.18.2-smp i686 (tty1)

slack login: root
Password:
Last login: Wed Jun  8 22:32:01 on tty1
Linux 5.18.2-smp.
root@slack:~#
```

## Wnioski

Poza małymi problemami z zegarem systemowym nie miałem większych problemów z kompilacją jądra przy pomocy obu metod. Mimo wszystko, odrobinę bardziej intuicyjna wydaje się mi starsza, jednak prawdopodobnie jest to związane z tym, że była to metoda, którą robiliśmy to zadanie na zajęciach. Wydaje mi się, ponieważ nie sprawdzałem tego z zegarkiem w ręku, że nowsza metoda zajęła odrobinę więcej czasu.

Jeśli miałbym kompilować jądro na własnym systemie, mimo wszystko użyłbym metody pierwszej.