

# Kompilacja jądra Linux

Łukasz Wójcik

## 1. Stara metoda localmodconfig

make bzImage

```
root@slack32:/usr/src/linux-5.12.1# time make -j2 bzImage
  SYNC      include/config/auto.conf.cmd
  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
  AS        arch/x86/entry/entry_32.o
  CC        init/main.o
  CC        arch/x86/entry/syscall_32.o
  CC        arch/x86/entry/common.o
  CC        arch/x86/entry/vdso/vma.o
  CHK       include/generated/compile.h
  CC        init/do_mounts.o
  CC        arch/x86/entry/vdso/extable.o
  CC        arch/x86/entry/vdso/vdso32-setup.o
  CC        init/do_mounts.rd.o

  CC        arch/x86/boot/compressed/early_serial_console.o
  CC        arch/x86/boot/compressed/acpi.o
  CC        arch/x86/boot/compressed/misc.o
  LZMA      arch/x86/boot/compressed/vmlinux.bin.lzma
  MKPIGGY   arch/x86/boot/compressed/piggy.S
  AS        arch/x86/boot/compressed/piggy.o
  LD        arch/x86/boot/compressed/vmlinux
  ZOFFSET   arch/x86/boot/zoffset.h
  OBJCOPY   arch/x86/boot/vmlinux.bin
  AS        arch/x86/boot/header.o
  LD        arch/x86/boot/setup.elf
  OBJCOPY   arch/x86/boot/setup.bin
  BUILD     arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready  (#2)

real    9m50.400s
user    15m41.843s
sys     2m29.001s
```

Kompilacja została wykonana przy użyciu 2 rdzeni. Operacja przebiegła pomyślnie i zajęła 9 minut i 50 sekund.

make modules

```
root@slack32:/usr/src/linux-5.12.1# time make -j2 modules
CALL scripts/atomic/check-atomics.sh
CALL scripts/checksyscalls.sh
CC [M] arch/x86/crypto/crc32-pclmul_glue.o
LD [M] arch/x86/crypto/crc32-pclmul.o
CC [M] fs/fuse/dev.o
CC [M] drivers/video/fbdev/core/sysfillrect.o
CC [M] fs/fuse/dir.o
CC [M] drivers/video/fbdev/core/syscopyarea.o
CC [M] drivers/video/fbdev/core/sysimgblt.o
CC [M] fs/fuse/file.o
CC [M] drivers/video/fbdev/core/fb_sys_fops.o
CC [M] drivers/acpi/ac.o
CC [M] drivers/acpi/button.o
CC [M] drivers/acpi/processor_core.o
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

real    1m3.560s
user    1m47.149s
sys     0m14.779s
root@slack32:/usr/src/linux-5.12.1#
```

Kompilacja została wykonana przy użyciu 2 rdzeni. Operacja przebiegła pomyślnie i zajęła 1 minutę i 3 sekundy.

make modules\_install

```
root@slack32:/usr/src/linux-5.12.1# time make -j2 modules_install
INSTALL arch/x86/crypto/crc32-pclmul.ko
INSTALL drivers/acpi/ac.ko
INSTALL drivers/acpi/button.ko
INSTALL drivers/acpi/video.ko
INSTALL drivers/block/loop.ko
INSTALL drivers/char/agp/agpgart.ko
INSTALL drivers/char/agp/intel-agp.ko
INSTALL drivers/char/agp/intel-gtt.ko
INSTALL drivers/char/tpm/tpm.ko
INSTALL drivers/char/tpm/tpm_tis.ko
INSTALL drivers/char/tpm/tpm_tis_core.ko
INSTALL drivers/gpu/drm/drm.ko
INSTALL drivers/gpu/drm/drm_kms_helper.ko
INSTALL drivers/gpu/drm/ttm/ttm.ko
INSTALL drivers/gpu/drm/vmwgfx/vmwgfx.ko
INSTALL drivers/i2c/algos/i2c-algo-bit.ko
INSTALL drivers/i2c/busses/i2c-nix4.ko
INSTALL net/ieee80211/rx11.ko
INSTALL net/wireless/cfg80211.ko
INSTALL sound/ac97_bus.ko
INSTALL sound/core/snd-pcm.ko
INSTALL sound/core/snd-timer.ko
INSTALL sound/core/snd.ko
INSTALL sound/pci/ac97/snd-ac97-codec.ko
INSTALL sound/pci/snd-intel8x0.ko
INSTALL sound/soundcore.ko
DEPMOD 5.12.1-smp

real    0m1.124s
user    0m0.527s
sys     0m0.511s
```

Operacja przebiegła pomyślnie i zajęła 1,124 sekundy, wykorzystując 2 rdzenie.

Tworzenie dysku ram

```
root@slack32:/boot# /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:
mkinitrd -c -k 5.12.1-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd.gz
root@slack32:/boot# mkinitrd -c -k 5.12.1-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd.gz
31927 bloków
/boot/initrd.gz created.
Be sure to run lilo again if you use it.
root@slack32:/boot#
```

Tworzenie wpisu w lilo

```
image = /boot/vmlinuz-custom-5.12.1-old-smp
root = /dev/sda1
label = "kernel-old"
read-only
# Linux bootable partition config ends
```

```
root@slack32:/boot# root@slack32:/boot# lilo
Warning: LBA32 addressing assumed
Added Slackware_14.2 *
Added kernel-custom +
Added kernel-old
One warning was issued.
root@slack32:/boot#
```

## 2. Nowa metoda streamline\_config.pl

make bzImage

```
root@slack32:/usr/src/linux-5.12.1# time make -j2 bzImage
SYNC    include/config/auto.conf.cmd
CALL    scripts/atomic/check-atomics.sh
CALL    scripts/checksyscalls.sh
CHK     include/generated/compile.h
GZIP    kernel/config_data.gz
CC      kernel/configs.o
AR      kernel/built-in.a
GEN     .version
CHK     include/generated/compile.h
UPD     include/generated/compile.h
CC      init/version.o
AR      init/built-in.a
LD      vmlinux.o
MODPOST vmlinux.symvers
MODINFO modules.builtin.modinfo
GEN     modules.builtin
LD      .tmp_vmlinux.kallsyms1
KSYMS   .tmp_vmlinux.kallsyms1.S

VOFFSET arch/x86/boot/compressed/./voffset.h
OBJCOPY arch/x86/boot/compressed/vmlinux.bin
LZMA     arch/x86/boot/compressed/vmlinux.bin.lzma
CC       arch/x86/boot/compressed/misc.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 (#3)

real    0m29.479s
user    0m24.534s
sys     0m18.120s
```

Kompilacja z użyciem dwóch rdzeni zajęła około 30 sekund. Zkompilowane zostało o wiele mniej elementów w porównaniu do metody starej.

make modules

```
sys 0m10.120s
root@slack32:/usr/src/linux-5.12.1# time make -j2 modules
  CALL      scripts/atomic/check-atomics.sh
  CALL      scripts/checksyscalls.sh
  MODPOST   Module.symvers

real    0m5.874s
user    0m7.980s
sys     0m2.756s
root@slack32:/usr/src/linux-5.12.1#
```

Operacja przebiegła pomyślnie i zajęła 5,8 sekund przy użyciu dwóch rdzeni. Ponownie skompilowana została mniejsza ilość modułów.

make modules\_install

```
sys 0m2.750s
root@slack32:/usr/src/linux-5.12.1# time make -j2 modules_install
  INSTALL arch/x86/crypto/crc32-pclmul.ko
  INSTALL drivers/acpi/ac.ko
  INSTALL drivers/acpi/button.ko
  INSTALL drivers/acpi/video.ko
  INSTALL drivers/block/loop.ko
  INSTALL drivers/char/agp/agpgart.ko
  INSTALL drivers/char/agp/intel-agp.ko
  INSTALL drivers/char/agp/intel-gtt.ko
  INSTALL drivers/char/tpm/tpm.ko
  INSTALL drivers/char/tpm/tpm_tis.ko
  INSTALL drivers/char/tpm/tpm_tis_core.ko
  INSTALL drivers/gpu/drm/drm.ko
  INSTALL drivers/gpu/drm/drm_kms_helper.ko
  INSTALL drivers/gpu/drm/ttm/ttm.ko
  INSTALL drivers/gpu/drm/vmwgfx/vmwgfx.ko
  INSTALL drivers/i2c/algos/i2c-algo-bit.ko
  INSTALL drivers/i2c/busses/i2c-piix4.ko
  INSTALL drivers/i2c/i2c-core.ko

  INSTALL sound/core/snd-timer.ko
  INSTALL sound/core/snd.ko
  INSTALL sound/pci/ac97/snd-ac97-codec.ko
  INSTALL sound/pci/snd-intel8x0.ko
  INSTALL sound/soundcore.ko
  DEPMOD   5.12.1-smp

real    0m0.880s
user    0m0.527s
sys     0m0.499s
```

Komenda została zakończona w 0,8 sekundy używając dwóch rdzeni.

## Tworzenie dysku ram

```
root@slack32:/boot# /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:

mkinitrd -c -k 5.12.1-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd.gz
root@slack32:/boot# mkinitrd -c -k 5.12.1-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd.gz
31927 bloków
/boot/initrd.gz created.
Be sure to run lilo again if you use it.
root@slack32:/boot#
```

## Konfiguracja lilo

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

```
root@slack32:/boot# root@slack32:/boot# lilo
Warning: LBA32 addressing assumed
Added Slackware_14.2 *
Added kernel-custom +
Added kernel-old
Added kernel-new
One warning was issued.
```

### 3. Podsumowanie

Czas wykonania poszczególnych komend

	Stara metoda	Nowa metoda
make bzImage	9m 50,4s	29,48s
make modules	1m 3,56s	5,87s
make modules_install	1,12s	0,88s

Kompilacja przy użyciu nowej metody jest zdecydowanie szybsza. Wynika to głównie ze zmniejszenia ilości kompilowanych elementów.

Czas uruchamiania

Czas uruchamiania zmierzony jest od naciśnięcia klawisza enter w menu lilo do pojawienia się ekranu logowania. Przed rozruchem maszyna wirtualna jest całkowicie wyłączona. Jest to również pierwsze uruchomienie po kompilacji.



	Stara metoda	Nowa metoda
Czas	około 19 sekund	Około 18 sekund

Nowa metoda uruchamia się niewiele ponad sekundę szybciej.



## Rozmiar na dysku

```
lrwxrwxrwx 1 root root 28 Kwi 27 19:28 vmlinuz 7 vmlinuz-huge-smp 4.  
-rw-r--r-- 1 root root 8852336 maj 25 20:31 vmlinuz-custom-5.12.1-new-smp  
-rw-r--r-- 1 root root 8851760 maj 25 20:13 vmlinuz-custom-5.12.1-old-smp  
-rw-r--r-- 1 root root 8818544 maj 12 09:26 vmlinuz-custom-5.12.1-smp
```

Różnica jest bardzo niewielka, jądro skompilowane przy użyciu starej metody jest o 576 bajty mniejsze.