

[Autor: Damian Tomczyszyn](#)

Kompilacja przy pomocy Localmodconfig

1. Przechodzę do rozpakowanego katalogu z plikami źródłowymi.
2. Kopiuję wykorzystywany plik konfiguracyjny poleceniem `zcat /proc/config.gz > .config`

```
*
* PCI GPIO expanders
*
AMD 8111 GPIO driver (GPIO_AMD8111) [N/m/y/?] n
BT8XX GPIO abuser (GPIO_BT8XX) [N/m/y/?] (NEW) ^Cmake[1]: *** [scripts/kconfig/Makefile:53: localmod
config] Interrupt
make: *** [Makefile:628: localmodconfig] Interrupt

root@slack:/usr/src/linux-5.18.2# ls
COPYING      Makefile      config_strip  kernel/
CREDITS      Module.symvers  crypto/       lib/
Documentation/ README        drivers/      mm/
Kbuild       System.map    fs/           modules-only.symvers
Kconfig      arch/         include/      modules.builtin
LICENSES/    block/        init/         modules.builtin.modinfo
MAINTAINERS  certs/        ipc/          modules.order
net/         virt/
samples/    umlinux*
scripts/   umlinux.o
security/  umlinux.symvers
sound/
tools/
usr/

root@slack:/usr/src/linux-5.18.2# ls -a
./          .tmp_System.map      Module.symvers      modules.builtin
../         .tmp_umlinux.kallsyms1*  README             modules.builtin.modinfo
..config.un~ .tmp_umlinux.kallsyms1.S  System.map          modules.order
.Module.symvers.cmd .tmp_umlinux.kallsyms1.o  arch/              net/
.clang-format .tmp_umlinux.kallsyms2*  block/             samples/
.cocciconfig .tmp_umlinux.kallsyms2.S  certs/             scripts/
.config      .tmp_umlinux.kallsyms2.o  config_strip        security/
.config.old  .version                crypto/             sound/
.config.old.1 .umlinux.cmd            drivers/            tools/
.config~     COPYING                 fs/                 usr/
.get_maintainer.ignore CREDITS                 include/            virt/
.gitattributes Documentation/           init/              umlinux*
.gitignore   Kbuild                  ipc/               umlinux.o
.mailmap     Kconfig                 kernel/            umlinux.symvers
.missing-syscalls.d LICENSES/                lib/
.modules-only.symvers.cmd MAINTAINERS             mm/
.modules.order.cmd Makefile                 modules-only.symvers

root@slack:/usr/src/linux-5.18.2# rm .config
root@slack:/usr/src/linux-5.18.2# zcat /proc/config.gz > .config
root@slack:/usr/src/linux-5.18.2#
```

3. Wydaję polecenie `make localmodconfig`

Dokonujemy zmian wybierając opcje które wpisy mają zostać zawarte. Nie chcę dokonywać wyborów, więc wszędzie daję enter, aby zastosował opcje domyślne.

4. Dostaję wygenerowaną konfigurację zapisaną do .config

```
Intel PMC Core driver (INTEL_PMC_CORE) [Y/n/m/?] y
Intel HID Event (INTEL_HID_EVENT) [N/m/?] n
Intel Virtual Button (INTEL_VBTN) [N/m/?] n
Intel ACPI INT0002 Virtual GPIO driver (INTEL_INT0002_UGPIO) [N/m/y/?] n
Intel Oaktrail Platform Extras (INTEL_OAKTRAIL) [N/m/?] n
Intel P-Unit IPC Driver (INTEL_PUNIT_IPC) [N/m/y/?] n
Intel Rapid Start Technology Driver (INTEL_RST) [N/m/y/?] n
Intel ACPI Smart Connect disabling driver (INTEL_SMARTCONNECT) [N/m/y/?] n
Intel Vendor Specific Extended Capabilities Driver (INTEL_VSEC) [N/m/y/?] n
MSI Laptop Extras (MSI_LAPTOP) [N/m/?] n
PC Engines APUv2/3 front button and LEDs driver (PCENGINES_APU2) [N/m/y/?] n
Barco P50 GPIO driver for identify LED/button (BARCO_P50_GPIO) [N/m/y/?] n
Samsung Laptop driver (SAMSUNG_LAPTOP) [N/m/?] n
Samsung Q10 Extras (SAMSUNG_Q10) [N/m/y/?] n
Toshiba Bluetooth RfKill switch support (TOSHIBA_BT_RFKILL) [N/m/?] n
Toshiba HDD Active Protection Sensor (TOSHIBA_HAPS) [N/m/y/?] n
CMPC Laptop Extras (ACPI_CMPC) [N/m/?] n
Compal (and others) Laptop Extras (COMPAL_LAPTOP) [N/m/?] n
Panasonic Laptop Extras (PANASONIC_LAPTOP) [N/m/y/?] n
Sony Laptop Extras (SONY_LAPTOP) [N/m/?] n
Topstar Laptop Extras (TOPSTAR_LAPTOP) [N/m/y/?] n
Serial bus multi instantiate pseudo device driver (SERIAL_MULTI_INSTANTIATE) [N/m/?] n
Mellanox Technologies platform support (MLX_PLATFORM) [N/m/?] n
X86 Android tablet support (X86_ANDROID_TABLETS) [N/m/?] n
Intel Intelligent Power Sharing (INTEL_IPS) [N/m/y/?] n
Intel SCU PCI driver (INTEL_SCU_PCI) [Y/n/?] y
Intel SCU platform driver (INTEL_SCU_PLATFORM) [N/m/y/?] n
Intel SCU IPC utility driver (INTEL_SCU_IPC_UTIL) [N/m/y/?] n
Siemens Simatic IPC Class driver (SIEMENS_SIMATIC_IPC) [N/m/y/?] n
#
# configuration written to .config
#
root@slack:/usr/src/linux-5.18.2#
root@slack:/usr/src/linux-5.18.2#
root@slack:/usr/src/linux-5.18.2#
root@slack:/usr/src/linux-5.18.2#
root@slack:/usr/src/linux-5.18.2#
```

5. Uruchamiam `olddefconfig` do sprawdzenia błędów

```
root@slack:/usr/src/linux-5.18.2# make olddefconfig
#
# No change to .config
#
root@slack:/usr/src/linux-5.18.2#
```

6. Nie pokazało żadnych błędów więc uruchamiam budowanie jądra poleceniem `make -j8 bzImage`. Jądro budowało się około 5 min i 10s. Był to czas krótszy niż jaki otrzymałem kompilując jądro nową metodą.

```
GEN      modules.builtin
LD        .tmp_umlinux.kallsyms1
KSYMS     .tmp_umlinux.kallsyms1.S
AS        .tmp_umlinux.kallsyms1.S
LD        .tmp_umlinux.kallsyms2
KSYMS     .tmp_umlinux.kallsyms2.S
AS        .tmp_umlinux.kallsyms2.S
LD        umlinux
SYSMAP    System.map
SORTTAB   umlinux
CC        arch/x86/boot/edd.o
CC        arch/x86/boot/main.o
CC        arch/x86/boot/version.o
VOFFSET   arch/x86/boot/compressed/./voffset.h
AS        arch/x86/boot/compressed/head_32.o
OBJCOPY   arch/x86/boot/compressed/umlinux.bin
RELOCS    arch/x86/boot/compressed/umlinux.relocs
CC        arch/x86/boot/compressed/cmdline.o
CC        arch/x86/boot/compressed/error.o
CC        arch/x86/boot/compressed/early_serial_console.o
CC        arch/x86/boot/compressed/kaslr.o
CC        arch/x86/boot/compressed/acpi.o
LZMA      arch/x86/boot/compressed/umlinux.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/umlinux
OBJCOPY   arch/x86/boot/umlinux.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
Kernel: arch/x86/boot/bzImage is ready (#2)
root@slack:/usr/src/linux-5.18.2#
root@slack:/usr/src/linux-5.18.2#
```

Kompilacja przy pomocy streamline_config.pl

1. Skopiowanie naszego pliku konfiguracyjnego poleceniem `zcat /proc/config.gz > .config`

```
linux-5.18.2/usr/default_cpio_list
linux-5.18.2/usr/gen_init_cpio.c
linux-5.18.2/usr/gen_initramfs.sh
linux-5.18.2/usr/include/
linux-5.18.2/usr/include/.gitignore
linux-5.18.2/usr/include/Makefile
linux-5.18.2/usr/include/headers_check.pl
linux-5.18.2/usr/initramfs_data.S
linux-5.18.2/virt/
linux-5.18.2/virt/Makefile
linux-5.18.2/virt/kvm/
linux-5.18.2/virt/kvm/Kconfig
linux-5.18.2/virt/kvm/Makefile.kvm
linux-5.18.2/virt/kvm/async_pf.c
linux-5.18.2/virt/kvm/async_pf.h
linux-5.18.2/virt/kvm/binary_stats.c
linux-5.18.2/virt/kvm/coalesced_mmio.c
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/ufio.c
linux-5.18.2/virt/kvm/ufio.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# ls
linux-5.18.2/  linux-5.18.2.tar.xz
root@slack:/usr/src# cd linux-5.18.2
root@slack:/usr/src/linux-5.18.2# pwd
/usr/src/linux-5.18.2
root@slack:/usr/src/linux-5.18.2# zcat /proc/config.gz > .config
root@slack:/usr/src/linux-5.18.2#
```

2. Otwieramy plik `streamline_config.pl` i czytamy instrukcję instalacji

```
#!/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.
```

1,19

[Top](#)

3. Uruchamiamy skrypt do utworzenia pliku konfiguracyjnego

```

root@slack:/usr/src/linux-5.18.2# ls
COPYING      Kconfig      README      crypto/     init/       mm/          security/   virt/
CREDITS      LICENSES/    arch/       drivers/    ipc/        net/         sound/
Documentation/ MAINTAINERS  block/      fs/         kernel/     samples/     tools/
Kbuild       Makefile     certs/      include/    lib/        scripts/     usr/
root@slack:/usr/src/linux-5.18.2# ls
COPYING      Kconfig      README      crypto/     init/       mm/          security/   virt/
CREDITS      LICENSES/    arch/       drivers/    ipc/        net/         sound/
Documentation/ MAINTAINERS  block/      fs/         kernel/     samples/     tools/
Kbuild       Makefile     certs/      include/    lib/        scripts/     usr/
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#

```

4. Można sprawdzić zainstalowane moduły poleceniem lsmod - wszystkie moduły które aktualnie załadowane są w systemie
5. Możemy moduły sami modyfikować. A aby nie edytować pliku tekstowego samodzielnie używamy polecenia make menuconfig , ale w moim przypadku nic nie zmieniałem.
6. Wykonuję make olddefconfig.

```
root@slack:/usr/src/linux-5.18.2# make olddefconfig
HOSTCC scripts/kconfig/conf.o
HOSTLD scripts/kconfig/conf
.config:440:warning: symbol value 'm' invalid for IBK
.config:8109:warning: symbol value 'm' invalid for VIDEO_ZORAN_DC30
.config:8110:warning: symbol value 'm' invalid for VIDEO_ZORAN_ZR36060
.config:8111:warning: symbol value 'm' invalid for VIDEO_ZORAN_BUZ
.config:8112:warning: symbol value 'm' invalid for VIDEO_ZORAN_DC10
.config:8113:warning: symbol value 'm' invalid for VIDEO_ZORAN_LML33
.config:8114:warning: symbol value 'm' invalid for VIDEO_ZORAN_LML33R10
.config:8115:warning: symbol value 'm' invalid for VIDEO_ZORAN_AVS6EYES
.config:9705:warning: symbol value 'm' invalid for CRYPTO_LIB_BLAKE2S_GENERIC
#
# configuration written to .config
#
root@slack:/usr/src/linux-5.18.2# _
```

Polecenie wykryło niepoprawne ustawienia w konfiguracji.

7. Poprawiam je ręcznie w pliku konfiguracyjnym zmieniając wartość m na y dla CONFIG_VIDEO_ZORAN i wydaje polecenie make olddefconfig jeszcze raz, tym razem nie wyświetliło ostrzeżeń.


```
CONFIG_STAGING_MEDIA=y
CONFIG_INTEL_ATOMISP=y
CONFIG_VIDEO_ATOMISP=m
# CONFIG_VIDEO_ATOMISP_ISP2401 is not set
# CONFIG_VIDEO_ATOMISP_OV2722 is not set
# CONFIG_VIDEO_ATOMISP_GC2235 is not set
# CONFIG_VIDEO_ATOMISP_MSRLIST_HELPER is not set
# CONFIG_VIDEO_ATOMISP_MT9M114 is not set
# CONFIG_VIDEO_ATOMISP_GC0310 is not set
# CONFIG_VIDEO_ATOMISP_OV2680 is not set
# CONFIG_VIDEO_ATOMISP_OV5693 is not set
# CONFIG_VIDEO_ATOMISP_LM3554 is not set
CONFIG_VIDEO_ZORAN=y
# CONFIG_VIDEO_ZORAN_DC30 is not set
# CONFIG_VIDEO_ZORAN_ZR36060 is not set
# CONFIG_VIDEO_IPU3_IMGU is not set
CONFIG_DVB_AU7110_IR=y
CONFIG_DVB_AU7110=m
CONFIG_DVB_AU7110_OSD=y
CONFIG_DVB_BUDGET_PATCH=m
CONFIG_DVB_SP8870=m
CONFIG_LTE_GDM724X=m
CONFIG_FIREWIRE_SERIAL=m
CONFIG_FWTTY_MAX_TOTAL_PORTS=64
CONFIG_FWTTY_MAX_CARD_PORTS=32
# CONFIG_UNISYSSPAR is not set
# CONFIG_FB_TFT is not set
# CONFIG_KS7010 is not set
# CONFIG_PI433 is not set
# CONFIG_FIELDBUS_DEV is not set
CONFIG_QLGE=m
".config" 10485L, 244202B written
root@slack:/usr/src/linux-5.18.2# make olddefconfig
#
# configuration written to .config
#
root@slack:/usr/src/linux-5.18.2# _
```

8. Teraz buduję jądro poleceniem `make -j8 bzImage`

```
CC      init/calibrate.o
CC      init/initramfs.o
HOSTCC  usr/gen_init_cpio
UPD      include/generated/compile.h
CC      init/init_task.o
GEN      usr/initramfs_data.cpio
CC      init/version.o
COPY    usr/initramfs_inc_data
AS       usr/initramfs_data.o
AR       usr/built-in.a
CC      arch/x86/entry/vdso/vma.o
CC      arch/x86/entry/vdso/extable.o
AR       arch/x86/entry/vsyscall/built-in.a
AS       arch/x86/entry/entry_32.o
CC      arch/x86/entry/vdso/vdso32-setup.o
LDS      arch/x86/entry/vdso/vdso32/vdso32.lds
AS       arch/x86/entry/vdso/vdso32/note.o
AS       arch/x86/entry/thunk_32.o
CC      arch/x86/entry/syscall_32.o
AS       arch/x86/entry/vdso/vdso32/system_call.o
CC      arch/x86/entry/common.o
AS       arch/x86/entry/vdso/vdso32/sigreturn.o
CC      arch/x86/entry/vdso/vdso32/vclock_gettime.o
HOSTCC  arch/x86/entry/vdso/vdso2c
CC      arch/x86/events/amd/core.o
CC      arch/x86/events/intel/core.o
CC      arch/x86/events/amd/ibs.o
CC      arch/x86/events/amd/uncore.o
CC      arch/x86/events/intel/bts.o
CC      arch/x86/events/intel/ds.o
AR       init/built-in.a
VDSO    arch/x86/entry/vdso/vdso32.so.dbg
OBJCOPY arch/x86/entry/vdso/vdso32.so
CC      kernel/sched/core.o
VDSO2C  arch/x86/entry/vdso/vdso-image-32.c
CC      arch/x86/entry/vdso/vdso-image-32.o
```

Budowa jądra na 8 procesorach zajęła około 5min i 30s.

Wnioski

1. Proces kompilacji jądra przy użyciu starej metody był nieznacznie szybszy.
2. Jeśli nie chce się zmieniać żadnych ustawień lepiej wybrać nową metodę, ponieważ nie wymaga przeklikania ustawień jak localmodconfig.
3. Localmodconfig z konfiguracji usuwa wszystkie zbędne wpisy, aby nie dokładać ciężaru do jądra.(nie wiem czy nowa metoda też tak robi ,ale wyczytałem to na [stack overflow](#))
4. Po wykonaniu olddefconfig mimo iż pokazało wiele ostrzeżeń podczas zmieniania tylko jednego wpisu i uruchomienia polecenia ponownie już więcej ich nie pokazało.
5. Wykonałem obydwie kompilacje jądra na jednej maszynie. Na początku obawiałem się że jedna kompilacja nadpisze ważne pliki z pierwszej kompilacji, które miały świadczyć o autentyczności pracy, ale skoro to nie ma znaczenia, to nie ma.

Podczas kopiowania do udostępnionego folderu w celu przeniesienia archiwum z wirtualnej maszyny do systemu sypie błędami, więc nie wiem czy wszystkie pliki jakie powinienem wrzucić do repozytorium zostały poprawnie dodane.

```
_64.S': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/copyloops/copyuser
_power7.S': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/copyloops/mem_64.S
': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/copyloops/memcpy_6
4.S': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/copyloops/memcpy_p
ower7.S': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/mce/uas-api.h': Op
eration not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/nx-gzip/include/ua
s-api.h': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/primitives/asm/asm
_compat.h': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/primitives/asm/asm
_const.h': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/primitives/asm/ext
able.h': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/primitives/asm/fea
ture-fixups.h': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/primitives/asm/ppc
_asm.h': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/primitives/word-at
-a-time.h': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/stringloops/memcmp
_32.S': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/stringloops/memcmp
_64.S': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/stringloops/strlen
_32.S': Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/uphn/asm/lppaca.h'
: Operation not permitted
cp: cannot create symbolic link '/mnt/share/linux/tools/testing/selftests/powerpc/uphn/uphn.c': Oper
ation not permitted
cp: cannot create symbolic link '/mnt/share/linux/Documentation/Changes': Operation not permitted
```