

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное  
учреждение  
высшего образования  
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И.  
ВЕРНАДСКОГО»  
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ  
Кафедра компьютерной инженерии и моделирования

ОТЧЕТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ № 7  
«Ядро ОС Linux. Сборка и установка.»

Практическая работа  
по дисциплине «Операционные системы»  
студента 1 курса группы ПИ-б-о-231(2)  
Аметов Кемран Ленверович

направления подготовки 09.03.04 «Программная инженерия»

Симферополь, 2024

**Цель:** познакомиться с понятием Ядро ОС, принципами конфигурации, компиляции и установки ядра Linux из исходных кодов. Изучить методику сборки новой версии ядра Linux из исходных кодов. Получить и распаковать свежую копию исходных кодов ядра Linux. Выполнить конфигурацию исходных кодов ядра. Скомпилировать и установить ядро Linux. Проверить работоспособность созданной системы.

Ход работы:

1. Для последующей сборки ядра необходимо установить дополнительные пакеты, например для дистрибутивов семейства Debian GNU/Linux:

```
kemran@kemran:/usr/src$ sudo apt install fakeroot build-essential libncurses-dev xz-utils libssl-dev flex libelf-dev bison bzip2 wget
[sudo] пароль для kemran:
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Уже установлен пакет bison самой новой версии (2:3.8.2+dfsg-1build1).
Уже установлен пакет build-essential самой новой версии (12.9ubuntu3).
Уже установлен пакет bzip2 самой новой версии (1.0.8-5build1).
Уже установлен пакет fakeroot самой новой версии (1.28-1ubuntu1).
Уже установлен пакет flex самой новой версии (2.6.4-8build2).
Уже установлен пакет libelf-dev самой новой версии (0.186-1build1).
Уже установлен пакет wget самой новой версии (1.21.2-2ubuntu1).
Уже установлен пакет xz-utils самой новой версии (5.2.5-2ubuntu1).
Уже установлен пакет libncurses-dev самой новой версии (6.3-2ubuntu0.1).
Уже установлен пакет libssl-dev самой новой версии (3.0.2-0ubuntu1.15).
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 78 пакетов не обновлено.
kemran@kemran:/usr/src$
```

2. Скачайте с сайта <https://kernel.org/> свежую копию исходных кодов ядра

```
kemran@kemran:/usr/src$ wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.8.9.tar.xz
--2024-05-28 19:26:18-- https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.8.9.tar.xz
Распознаётся cdn.kernel.org (cdn.kernel.org)... 146.75.117.176, 2a04:4e42:8d::432
Подключение к cdn.kernel.org (cdn.kernel.org)|146.75.117.176|:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 142582332 (136M) [application/x-xz]
Сохранение в: 'linux-6.8.9.tar.xz'

linux-6.8.9.tar.xz          100%[=====] 135,98M  11,0MB/s   за 28s
2024-05-28 19:26:47 (4,78 MB/s) - 'linux-6.8.9.tar.xz' сохранён [142582332/142582332]
```

### 3. Проверьте верность скачанного архива по *PGP* подписи:

```
kemran@kemran:/usr/src$ unxz linux-6.8.9.tar.xz
kemran@kemran:/usr/src$ gpg --verify linux-6.8.9.tar.sign linux-6.8.9.tar
gpg: Подпись сделана Чт 02 мая 2024 17:36:34 MSK
gpg:          ключом RSA с идентификатором 647F28654894E3BD457199BE38DBBDC86092693E
gpg: Действительная подпись пользователя "Greg Kroah-Hartman <gregkh@kernel.org>" [неизвестно]
gpg: Внимание: Данный ключ не заверен доверенной подписью!
gpg:          Нет указаний на то, что подпись принадлежит владельцу.
Отпечаток первичного ключа: 647F 2865 4894 E3BD 4571  99BE 38DB BDC8 6092 693E
kemran@kemran:/usr/src$
```

### 4. Распакуйте архив с кодами в папку предназначенную для хранения исходных кодов ПО в GNU/Linux:

```
linux-6.8.9/virt/lib/
linux-6.8.9/virt/lib/Kconfig
linux-6.8.9/virt/lib/Makefile
linux-6.8.9/virt/lib/irqbypass.c
kemran@kemran:/usr/src$ tar -vxf linux-6.8.9.tar
```

### 5. В данной папке сделайте ссылку linux указывающую на распакованную вами директорию.

```
kemran@kemran:/usr/src$ ln -s linux-6.8.9 linux
```

### 6. Перейдите в распакованную директорию и запустите процесс конфигурации:

```

kemran@kemran:/usr/src/linux$ make menuconfig
HOSTCC  scripts/basic/fixdep
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
HOSTCC  scripts/kconfig/mnconf-common.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/mconf
#
# using defaults found in /boot/config-6.5.0-35-generic
#
.config:10580:warning: symbol value 'm' invalid for ANDROID_BINDER_IPC
.config:10581:warning: symbol value 'm' invalid for ANDROID_BINDERFS
.config:10807:warning: symbol value 'm' invalid for FSCACHE

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

```

включение поддержки виртуализации KVM и Xen

```

.config - Linux/x86 6.8.9 Kernel Configuration
> Virtualization
    Virtualization
    Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
    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

    -- Virtualization
    <*> Kernel-based Virtual Machine (KVM) support
    [*]   Compile KVM with -Werror
    [ ]   Enable support for KVM software-protected VMs
    <*>   KVM for Intel (and compatible) processors support
    [ ]   Software Guard extensions (SGX) Virtualization
    < >   KVM for AMD processors support
    [ ]   System Management Mode emulation
    [ ]   Support for Microsoft Hyper-V emulation
    [*]   Support for Xen hypercall interface
    [ ]   Prove KVM MMU correctness
    (4096) Maximum number of vCPUs per KVM guest

```

## полная поддержка ядром cgroups

```
.config - Linux/x86 6.8.9 Kernel Configuration
> General setup > Control Group support
Control Group support
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
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

-- Control Group support
[*] Favor dynamic modification latency reduction by default
[*] Memory controller
[*] IO controller
-* CPU controller --->
[*] Utilization clamping per group of tasks
[*] PIDs controller
[*] RDMA controller
[*] Freezer controller
[*] HugeTLB controller
[*] Cpuset controller
[*] Include legacy /proc/<pid>/cpuset file
[*] Device controller
[*] Simple CPU accounting controller
[*] Perf controller
[*] Support for eBPF programs attached to cgroups
[*] Misc resource controller
[*] Debug controller
```

## настройка межсетевого экрана iptables

```
.config - Linux/x86 6.8.9 Kernel Configuration
[...] kernel options > Network packet filtering framework (Netfilter) > Core Netfilter Configuration
Core Netfilter Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
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

^(-)
-* Network Address Translation support
<*> Netfilter nf_tables support
[*] Netfilter nf_tables mixed IPv4/IPv6 tables support
[*] Netfilter nf_tables netdev tables support
<*> Netfilter nf_tables number generator module
<*> Netfilter nf_tables conntrack module
<*> Netfilter nf_tables hardware flow offload module
<*> Netfilter nf_tables connlimit module
<*> Netfilter nf_tables log module
<*> Netfilter nf_tables limit module
<*> Netfilter nf_tables masquerade support
<*> Netfilter nf_tables redirect support
<*> Netfilter nf_tables nat module
<*> Netfilter nf_tables tunnel module
<*> Netfilter nf_tables queue module
<*> Netfilter nf_tables quota module
<*> Netfilter nf_tables reject support
<*> Netfilter x_tables over nf_tables module
<*> Netfilter nf_tables hash module
<*> Netfilter nf_tables fib inet support
<*> Netfilter nf_tables xfrm/IPSec security association matching
<*> Netfilter nf_tables socket match support
<*> Netfilter nf_tables passive OS fingerprint support
<*> Netfilter nf_tables tproxy support
<*> Netfilter nf_tables SYNPROXY expression support
-* Netfilter packet duplication support
<*> Netfilter nf_tables netdev packet duplication support
<*> Netfilter nf_tables netdev packet forwarding support
<*> Netfilter nf_tables netdev fib lookups support
<*> Netfilter nf_tables netdev REJECT support
<*> Netfilter flow table mixed IPv4/IPv6 module
<*> Netfilter flow table module
[ ] Supply flow table statistics in procfs
-* Netfilter Xtables support (required for ip_tables)
[*] Netfilter Xtables 32bit support
*** Xtables combined modules ***
{M} nfmark target and match support
{M} ctmark target and match support
<M> set target and match support
<M> *** Xtables targets ***
<M> AUDIT target support
v(+)

<Select> < Exit > < Help > < Save > < Load >
```



## поддержка initrd

```
.config - Linux/x86 6.8.9 Kernel Configuration
> General setup

General setup
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
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

^(-)
[*] Auditing support
    IRQ subsystem --->
    Timers subsystem --->
    BPF subsystem --->
    Preemption Model (Voluntary Kernel Preemption (Desktop)) --->
[*] Preemption behaviour defined on boot
[*] Core Scheduling for SMT
    CPU/Task time and stats accounting --->
[*] CPU isolation
    RCU Subsystem --->
< > Kernel .config support
<M> Enable kernel headers through /sys/kernel/kheaders.tar.xz
(18) Kernel log buffer size (16 => 64KB, 17 => 128KB)
(12) CPU kernel log buffer size contribution (13 => 8 KB, 17 => 128KB)
[ ] Printk indexing debugfs interface
    Scheduler features --->
[*] Memory placement aware NUMA scheduler
[*] Automatically enable NUMA aware memory/task placement
-* Control Group support --->
[*] Namespaces support --->
[*] Checkpoint/restore support
[*] Automatic process group scheduling
-* Kernel->user space relay support (formerly relayfs)
-* Initial RAM filesystem and RAM disk (initramfs/initrd) support
    () Initramfs source file(s)
    [*] Support initial ramdisk/ramfs compressed using gzip
    [*] Support initial ramdisk/ramfs compressed using bzip2
    [*] Support initial ramdisk/ramfs compressed using LZMA
    [*] Support initial ramdisk/ramfs compressed using XZ
    [*] Support initial ramdisk/ramfs compressed using LZ0
    [*] Support initial ramdisk/ramfs compressed using LZ4
    [*] Support initial ramdisk/ramfs compressed using ZSTD
    [*] Boot config support
    [ ] Force unconditional bootconfig processing
    [ ] Embed bootconfig file in the kernel
    [*] Preserve cpio archive mtimes in initramfs
    Compiler optimization level (Optimize for performance (-O2)) --->
    [*] Configure standard kernel features (expert users) --->
    Kernel Performance Events And Counters --->
    [*] Profiling support
    Kexec and crash features --->

<Select> < Exit > < Help > < Save > < Load >
```

Выключение хранения конфигурации в ядре

```
config - Linux/x86 6.8.9 Kernel Configuration
General setup

General setup
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

^(-)
  Preemption Model (Voluntary Kernel Preemption (Desktop)) ---
  [*] Preemption behaviour defined on boot
  [*] Core Scheduling for SMT
  CPU/Task time and stats accounting --->
  [*] CPU isolation
  RCU Subsystem --->
  <> Kernel .config support
  <M> Enable kernel headers through /sys/kernel/kheaders.tar.xz
  (18) Kernel log buffer size (16 => 64KB, 17 => 128KB)
  (12) CPU kernel log buffer size contribution (13 => 8 KB, 17 => 1
  [ ] Printk indexing debugfs interface
  Scheduler features --->
  *- Control Group support --->
  [*] Namespaces support --->
  [ ] Checkpoint/restore support
  [*] Automatic process group scheduling
  *- Kernel->user space relay support (formerly relayfs)
  *- Initial RAM filesystem and RAM disk (initramfs/initrd) suppor
  ( )
  Initramfs source file(s)
  [*] Support initial ramdisk/ramfs compressed using gzip
  [*] Support initial ramdisk/ramfs compressed using bzip2
  [*] Support initial ramdisk/ramfs compressed using LZMA
  [*] Support initial ramdisk/ramfs compressed using XZ
  [*] Support initial ramdisk/ramfs compressed using LZ0
  [*] Support initial ramdisk/ramfs compressed using LZ4
  [*] Support initial ramdisk/ramfs compressed using ZSTD
  [*] Boot config support
  [ ] Force unconditional bootconfig processing
  v(+)

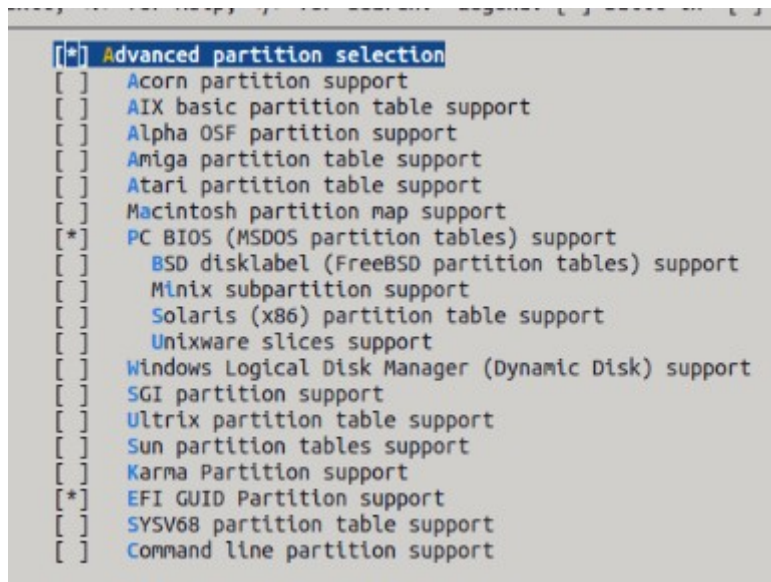
<Select> < Exit > < Help > < Save > < Load >
```

поддержка загрузки/выгрузки модулей

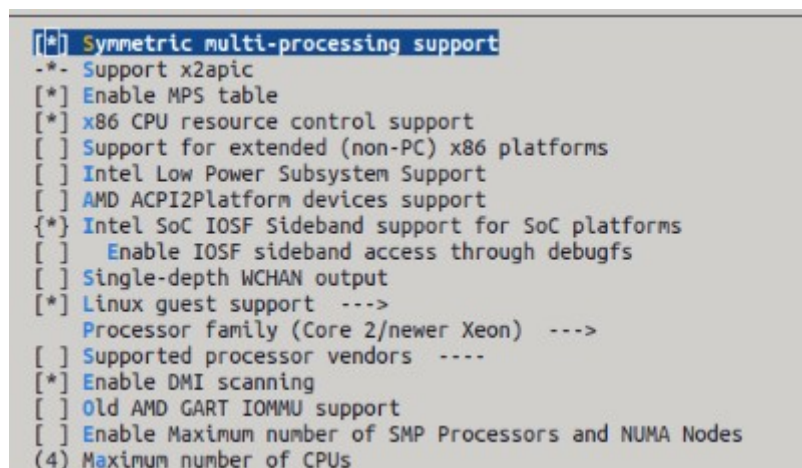
```
Arrow keys navigate the menu. <Enter> selects submenus ---> (or emp
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

-- Enable loadable module support
[ ] Module debugging
[ ] Forced module loading
[*] Module unloading
[ ] Forced module unloading
[ ] Tainted module unload tracking
[ ] Module versioning support
[ ] Source checksum for all modules
[ ] Module signature verification
Module compression mode (ZSTD) --->
[*] Support in-kernel module decompression
[ ] Allow loading of modules with missing namespace imports
(/sbin/modprobe) Path to modprobe binary
[ ] Trim unused exported kernel symbols
```

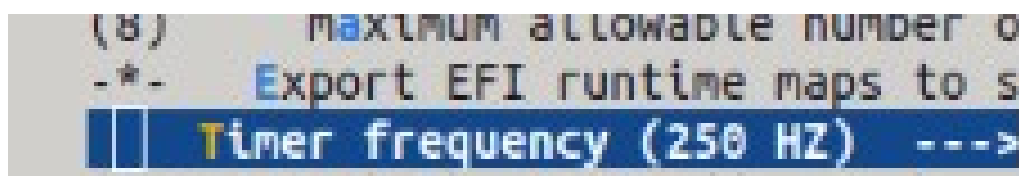
поддержка таблиц разделов только PC-BIOS и EFI GUID



поддержка SMP и максимальное количество CPU — 4



выставленная частота таймера (250 Hz)



поддержка PCI



```
.config - Linux/x86 6.8.9 Kernel Configuration
> Device Drivers

                                Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

[*] EISA support --->
[*] PCI support --->
<M> PCCard (PCMCIA/CardBus) support --->
<*> RapidIO support --->
    Generic Driver Options --->
    Bus devices --->
    Cache Drivers ----
[*] Connector - unified userspace <-> kernelspace linker ---->
    Firmware Drivers --->
< > GNSS receiver support ----
< > Memory Technology Device (MTD) support ----
[ ] Device Tree and Open Firmware support ----
<M> Parallel port support --->
-* Plug and Play support --->
[*] Block devices --->
    NVME Support --->
    Misc devices --->
    SCSI device support --->
<*> Serial ATA and Parallel ATA drivers (libata) --->
[*] Multiple devices driver support (RAID and LVM) --->
< > Generic Target Core Mod (TCM) and ConfigFS Infrastructure --
[*] Fusion MPT device support --->
    IEEE 1394 (FireWire) support --->
[*] Macintosh device drivers --->
-* Network device support --->
    Input device support --->
    Character devices --->
    I2C support --->
<M> I3C support --->
v(+)

<Select>  < Exit >  < Help >  < Save >  < Load >
```

поддержка драйверов VIRTIO

```
-- Virtio drivers
<*> PCI driver for virtio devices
[*] Support for legacy virtio draft 0.9.X and older devices
<*> vDPA driver for virtio devices
<*> Support for virtio pmem driver
<*> Virtio balloon driver
<*> Virtio input driver
<*> Platform bus driver for memory mapped virtio devices
[*] Memory mapped virtio devices parameter parsing
```

## Отключение звуковой карты

```
config - Linux/x86 0.8.9 Kernel Configuration
Device Drivers

Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus
submenus ----). Highlighted letters are hotkeys. Press
includes, <N> excludes, <M> modularizes features. Press
exit, <?> for Help, </> for Search. Legend: [*] built-i

^(-)
-* GPIO Support --->
{M} Dallas's 1-wire support --->
[*] Board level reset or power off --->
-* Power supply class support --->
[*] Hardware Monitoring support --->
-* Thermal drivers --->
[*] Watchdog Timer Support --->
{M} Sonics Silicon Backplane support --->
{M} Broadcom specific AMBA --->
    Multifunction device drivers --->
-* Voltage and Current Regulator Support --->
< > Remote Controller support ----
    CEC support --->
<M> Multimedia support --->
    Graphics support --->
[*] Compute Acceleration Framework --->
< > Sound card support ----
[*] HID bus support --->
[*] USB support --->
<*> MMC/SD/SDIO card support --->
<M> Universal Flash Storage Controller --->
<M> Sony MemoryStick card support --->
-* LED Support --->
[*] Accessibility support --->
<M> InfiniBand support --->
<*> EDAC (Error Detection And Correction) reporting
[*] Real Time Clock --->
[*] DMA Engine support --->
    DMABUF options --->
```

включенная поддержка файловых систем NFS, FUSE,  
Quota, Inotify/Dnotify

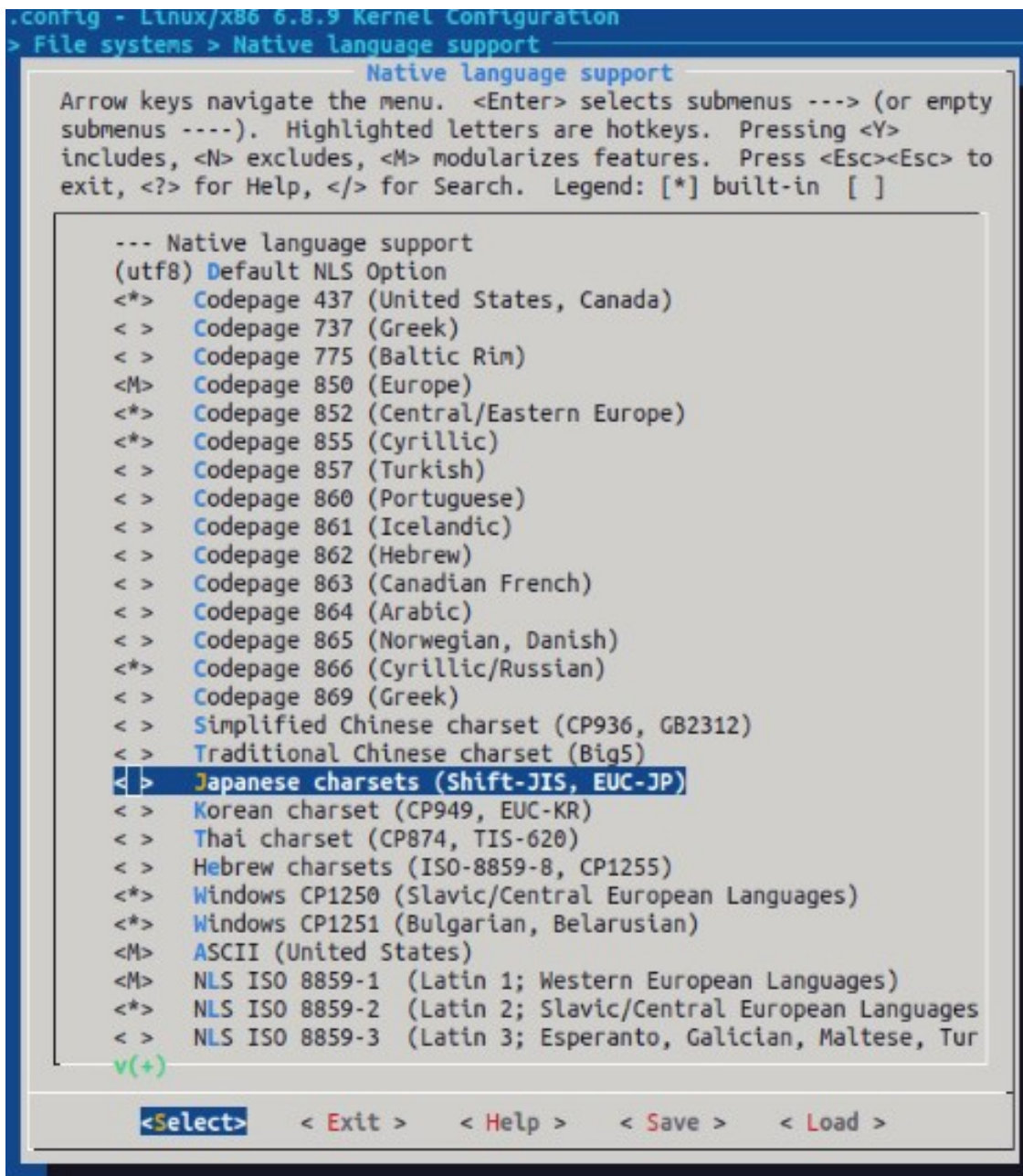
```
.config - Linux/x86 6.8.9 Kernel Configuration
> File systems
File systems
Arrow keys navigate the menu. <Enter> selects submenus ---> (or
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><E
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
^(-)
[*] FS Verity builtin signature support
[*] Dnotify support
[*] Inotify support for userspace
[*] Filesystem wide access notification
[*] Quota support
[*] Report quota messages through netlink interface
[ ] Additional quota sanity checks
< > Old quota format support
< > Quota format vfstv0 and vfstv1 support
<M> Kernel automounter support (supports v3, v4 and v5)
<*> FUSE (Filesystem in Userspace) support
<M> Character device in Userspace support
<*> Virtio Filesystem
[*] Virtio Filesystem Direct Host Memory Access support
<M> Overlay filesystem support
[ ] Overlayfs: turn on redirect directory feature by default
[*] Overlayfs: follow redirects even if redirects are turn
[ ] Overlayfs: turn on inodes index feature by default
[*] Overlayfs: auto enable inode number mapping
[ ] Overlayfs: turn on metadata only copy up feature by de
[ ] Overlayfs: turn on extra debugging checks
Caches --->
CD-ROM/DVD Filesystems --->
DOS/FAT/EXFAT/NT Filesystems --->
Pseudo filesystems --->
-*- Miscellaneous filesystems --->
[*] Network File Systems --->
-*- Native language support --->
<M> Distributed Lock Manager (DLM) --->
v(+)
```

включенная поддержка файловых систем MS-  
DOS и VFAT

```
.config - Linux/x86 6.8.9 Kernel Configuration
File systems > DOS/FAT/EXFAT/NT Filesystems
DOS/FAT/EXFAT/NT Filesystems
Arrow keys navigate the menu. <Enter> selects submenus ---> (or
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
[*] MSDOS fs support
<*> VFAT (Windows-95) fs support
(437) Default codepage for FAT
(iso8859-1) Default iocharset for FAT
[ ] Enable FAT UTF-8 option by default
<M> exFAT filesystem support
(utf8) Default iocharset for exFAT
<M> NTFS file system support
[ ] NTFS debugging support
[ ] NTFS write support
<M> NTFS Read-Write file system support
[ ] 64 bits per NTFS clusters
[*] activate support of external compressions lzx/xpress
[*] NTFS POSIX Access Control Lists
```



Кодировки UTF-8 и все кириллические  
кодировки русского языка



команда для начала сборки ядра

```
kemran@kemran:/usr/src/linux$ nproc  
12  
kemran@kemran:/usr/src/linux$ make -j12
```

команда для запуска сборки модулей

```
kemran@kemran:/usr/src/linux$ make modules
```



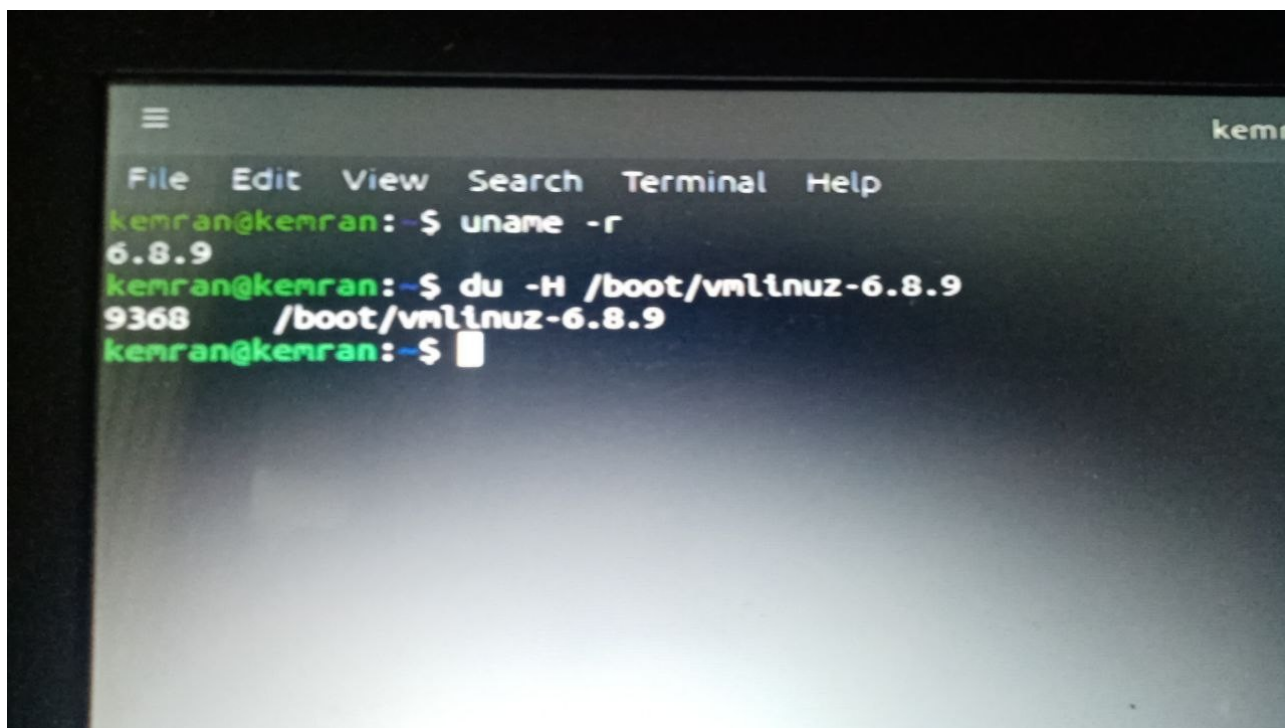
команда для запуска установки модулей ядра

```
kemran@kemran:/usr/src/linux$ make modules_install
```

команда для запуска установки ядра и автоматической  
настройки GRUB для загрузки ядра

```
kemran@kemran:/usr/src/linux$ make install
```

ЯДРО УСТАНОВЛЕНО



P.S Это лаба была адом. На моей основной системе ядро так и не поставилось, я всегда получал ошибку нехватки памяти. Её можно обойти изменив nano /etc/initramfs-tools/initramfs.conf после этого прописав sudo update-initramfs -v -u -k (версия ядра), а потом прописать апдейт-граб, в таком случае ядро запускалось и... УХОДИЛО В ВЕЧНУЮ ЗАГРУЗКУ(как оказалось позже решением после этого был запуск через граб с функцией nomodeset, но когда я это узнал ядро уже было ликвидировано). В общем мне пришлось ставить ядро на другую систему и там все запустилось без ошибок, но

с вечной загрузкой, решением оказался запуск с функцией `nomodeset`,  
как то так :(

1. Ядро операционной системы (ОС) — это основной компонент, который управляет аппаратными ресурсами компьютера и предоставляет основные функции для других программ. Ядро выполняет такие задачи, как управление процессами, памятью, устройствами ввода-вывода и сетевыми функциями. Оно действует как посредник между аппаратным обеспечением и программным обеспечением, обеспечивая стабильную и эффективную работу системы.
2. Linux — это свободная и открытая операционная система, основанная на ядре Linux, разработанном Линусом Торвальдсом в 1991 году. Сегодня термин "Linux" часто используется для обозначения целых операционных систем, которые включают в себя ядро Linux и различные утилиты и приложения из проекта GNU и других источников, образуя так называемые дистрибутивы Linux, такие как Ubuntu, Fedora, Debian и другие.
3. Cgroups (control groups) — это функциональность ядра Linux, позволяющая ограничивать, учитывать и изолировать ресурсы (CPU, память, диск, сеть) для набора процессов. Cgroups используются для управления ресурсами в контейнерах и виртуальных машинах, обеспечивая контроль над использованием системных ресурсов и повышение безопасности и стабильности системы.
4. Virtio — это стандарт для виртуализированных устройств, который позволяет более эффективно использовать виртуальные машины (VM). Устройства virtio, такие как сетевые интерфейсы (virtio-net) или дисковые устройства (virtio-blk), предоставляют высокопроизводительные и совместимые методы взаимодействия между виртуальными машинами и гипервизором, минимизируя накладные расходы на виртуализацию.
5. Особенности ядра

**Многозадачность:** Способность выполнять несколько процессов одновременно.

- **Многопользовательская система:** Поддержка множества пользователей с разделением прав доступа.
- **Модульность:** Возможность загружать и выгружать модули ядра по мере необходимости.
- **Поддержка различных файловых систем:** Включая ext4, XFS, Btrfs, FAT, NTFS и другие.
- **Высокая степень настройки:** Возможность компилировать ядро с опциями, соответствующими специфическим потребностям пользователя или устройства.
- **Поддержка различных архитектур:** Широкий спектр процессорных архитектур, включая x86, ARM, MIPS и другие.
- **Безопасность:** Механизмы управления доступом, такие как SELinux и AppArmor.
- **Поддержка сетевых технологий:** Включая IPv4, IPv6, различные протоколы и маршрутизацию.

## 6. Основные файлы, связанные с конфигурацией ядра Linux:

**.config:** Основной файл конфигурации, который создается после настройки параметров ядра с помощью команд `make menuconfig`, `make xconfig` и т.д.

- **/boot/config-\$(uname -r):** Файл конфигурации для текущей загруженной версии ядра.
- **/proc/config.gz:** Сжатый файл конфигурации ядра, доступный на некоторых системах через файловую систему /proc.

## 7. Этапы сборки

### Загрузка исходных кодов ядра:

- Скачайте архив с исходными кодами ядра с официального сайта [kernel.org](https://www.kernel.org).
- Разархивируйте его в нужный каталог.
- **Установка необходимых инструментов и зависимостей:**
- Убедитесь, что у вас установлены необходимые пакеты для сборки ядра. Обычно это пакеты типа `gcc`, `make`, `libncurses-dev`, и другие зависимости.
- **Очистка каталога сборки (опционально):**
- Перед началом сборки рекомендуется выполнить команду `make mrproper`, чтобы очистить каталог от старых артефактов и настроек. Это гарантирует чистую сборку.
- **Настройка конфигурации ядра:**
- `make menuconfig` или `make nconfig` - Запуск текстового интерфейса для настройки конфигурации ядра.
- `make xconfig` или `make gconfig` - Запуск графического интерфейса для настройки конфигурации ядра.
- `make defconfig` - Использование конфигурации по умолчанию.
- `make oldconfig` - Обновление текущей конфигурации до новой версии ядра.
- **Сборка ядра и модулей:**

- `make` - Сборка ядра.
- `make modules` - Сборка модулей ядра.
- **Установка модулей:**
- `make modules_install` - Установка собранных модулей в системный каталог (`/lib/modules/`).
- **Установка ядра:**
- `make install` - Установка ядра и связанных файлов (например, `System.map`, `config`) в системный каталог (`/boot/`).
- **Обновление загрузчика:**
- Обновите конфигурацию загрузчика (например, GRUB) для включения нового ядра.
- В Debian/Ubuntu можно использовать `update-grub`.
- **Перезагрузка:**
- Перезагрузите систему, чтобы использовать новое ядро.

8.

Команда `make mrproper` используется для глубокой очистки каталога исходных кодов ядра. Она выполняет следующие действия:

- Удаляет временные файлы и артефакты, созданные предыдущими сборками.
- Удаляет файлы конфигурации ядра, такие как `.config` и `Module.symvers`.
- Удаляет каталоги и файлы, созданные в процессе сборки, включая каталоги с объектными файлами и промежуточные файлы.

Это полезно, если вы хотите начать сборку с чистого листа, особенно если предыдущие сборки были прерваны или прошли с ошибками, чтобы избежать конфликта между старым и новым состоянием проекта.