Системне програмне забезпечення кіберфізичних систем

# Лабораторна робота № 1

НАЗВА: Визначення параметрів ядра ОС Linux та конфігурації апаратних засобів комп'ютера.
МЕТА: Навчитись отримувати інформацію про ядро ОС та апаратні засоби комп'ютера за допомогою команд і системних утиліт ОС Linux.

## 1. Загальні відомості

1.1. Ядро ОС Linux має монолітну архітектуру та реалізує базову функціональність системного рівня [1,2], доступ до якої програмам користувацького рівня надається через інтерфейс системних викликів (System Call Interface, Linux kernel API), що містить близько 380 системних викликів. Вихідний код ядра ОС Linux [3] та відповідна документація [4] є відкритими. Вони доступні на веб-сайті Linux Kernel Organization [3], яка представляє інтереси об'єднання розробників ядра.

1.2. Визначити версію ядра ОС Linux можна за допомогою команди **uname** з параметром -r ($uname -r) або в розширеному вигляді з параметром -v ($uname -v). Повну інформацію про систему (в тому числі її апаратну архітектуру) можна отримати, якщо вказати параметр -a ($uname -a). Інший спосіб визначити версію ядра — подивитись значення системного запису version ($cat /proc/version). Інформацію про версію ядра ОС Linux та іншу загальну інформацію про систему у зручному вигляді можна отримати за допомогою bash-скрипту Screenfetch (пакет screenfetch в Debian Linux).

1.2. Ядро ОС Linux — це складний програмний комплекс з великою кількістю параметрів, за допомогою яких можна змінювати режими його роботи та налагоджувати його функціональність в різних аспектах. Значення параметрів ядра та контроль за їх змінами грають важливу роль у забезпеченні ефективної роботи системи, її надійності та безпеки. Існує три способи конфігурації параметрів ядра: 1) під час компіляції ядра з вихідних кодів, 2) під час завантаження ядра в момент старту системи за допомогою параметрів командного рядка ядра [5], 3) під час роботи системи за допомогою системної утиліти **sysctl**. Конфігурація (зміна) параметрів ядра вимагає прав адміністратора системи та має виконуватись з великою обережністю.

1.3. Під час роботи системи параметри ядра ОС Linux відображаються у директорії **proc/sys/** віртуальної файлової системи proc/ у вигляді окремих системних записів (файлів), які згруповані у декілька груп (директорій) [6]. Групи параметрів ядра у proc/sys/ частково відповідають основним функціональним компонентам ядра: управління обчислювальними процесами (більшість параметрів з групи /proc/sys/kernel, в тому числі параметри диспетчеризації процесів /proc/sys/kernel/sched_* та /proc/sys/kernel/sched_domain/),

управління пам’яттю (група параметрів /proc/sys/vm/), управління вводом/виводом та пристроями (група параметрів /proc/sys/dev/), файлова підсистема (група параметрів /proc/sys/fs/), мережна підсистема (група параметрів /proc/sys/net/). Пояснення щодо самих параметрів ядра та їх значень можна знайти в Документації ядра ОС Linux [6]. Значення окремих параметрів ядра у proc/sys/ можна дивитись командою cat.

Приклади:

1) kernel.threads-max - максимальна кількість програмних потоків в системі:

```
$cat /proc/sys/kernel/threads-max
29147
```

2) kernel.yama.ptrace_scope — параметр модуля безпеки Yama, що визначає схему дозволів на використання системного виклику ptrace [7]:

```
$cat /proc/sys/kernel/yama/ptrace_scope
0
```

1.4. Системна утиліта **sysctl** призначена для конфігурації (зміни значень) параметрів ядра під час роботи системи. Нею також можна скористатись для визначення встановлених значень параметрів ядра. Виконавши **sysctl** з параметром -a ($sysctl -a), отримаємо повний список усіх параметрів ядра з їх значеннями (для виконання утиліти **sysctl** потрібні права адміністратора, для чого можна скористатись утилітою sudo). Для отримання значення окремого параметра ядра використовується параметр утиліти -n:

```
$sysctl -n kernel.threads-max
29147
```

Без параметра -n утиліта **sysctl** поверне значення параметру ядра в іншому вигляді:

```
$sysctl kernel.threads-max
kernel.threads-max = 29147
```

Приклади:

1) зберегти список всіх параметрів ядра зі значеннями у файлі:

```
$sysctl -a > kparams.txt
```

2) визначити загальну кількість параметрів ядра:

```
$sysctl -a | wc -l
```

3) вивести список параметрів з групи параметрів /proc/sys/vm/:

```
$sysctl -a | grep vm
```

1.5. В архітектурі ядра ОС Linux реалізовано механізм завантажуваних модулів ядра (loadable kernel modules, LKM), які можуть бути під’єднані чи від’єднані від ядра за потребою під час роботи системи. Основне призначення модулів ядра — забезпечення підтримки нових апаратних засобів у складі системи (драйвери пристроїв, які підключаються до комп’ютера). Завантажуваний модуль — це об’єктний файл з розширенням .ko (kernel object). Для роботи

з модулями використовуються команди insmod/rmmod чи modprobe (додати та видалити модуль з ядра).

1.6. Файли завантажуваних модулів зберігаються у директорії /lib/modules/. Список доступних для завантаження модулів можна подивитись в наступний спосіб:
$ls -R /lib/modules/4.9.0-11-amd64/kernel/ | grep .ko
Загальну кількість завантажуваних модулів можна визначити так:
$ls -R /lib/modules/4.9.0-11-amd64/kernel/ | grep -c .ko
Модулі, які завантажені і виконуються у складі ядра, відображаються у системні записи (файли) директорії /proc/modules віртуальної файлової системи proc/. Їх список можна подивитись, або виконавши: $less /proc/modules, або у більш зручному вигляді за допомогою команди **lsmod** ($lsmod). Інформацію про параметри окремого завантажуваного модуля можна отримати або командою **modinfo**, наприклад: $modinfo usbcore (потрібні права адміністратора), або виконавши: $/sbin/modinfo usbcore.

1.7. Використання завантажуваних модулів (LKM) може призвести до «забруднення» ядра. Ядро позначає себе «забрудненим» (tainted), коли з ним відбувається щось, що впливає на можливість ефективного усунення помилки або зависання ядра шляхом аналізу його вихідного коду [8]. Наприклад, у разі завантаження до ядра пропрієтарного модуля (стан «забруднення» ядра #0 (G/P, 1, proprietary module was loaded)) і виникнення помилки у роботі ядра, буде важко або взагалі неможливо встановити причину помилки, оскільки код такого модуля є закритим. З точки зору безпеки стан «забруднення» ядра #13 (E, 8192, unsigned module was loaded) може бути ознакою завантаження в ядро руткіту (rootkit) або виконання експлойту ядра (kernel exploit). Встановити чи є ядро «забрудненим» можна за допомогою параметру ядра kernel.tainted. Якщо kernel.tainted=0, то ядро «чисте», інакше (kernel.tainted>0) ядро є «забрудненим» і значення kernel.tainted визначає стан «забруднення» (див. таблицю станів «забруднення» у [8]).

Приклади:

1) визначення стану «забруднення» ядра:
```
$cat /proc/sys/kernel/tainted
4096
```

2) декодування стану «забруднення» ядра за допомогою скрипту kernel-chktaint [9]:
```
$ sh kernel-chktaint
Kernel is "tainted" for the following reasons:
 * externally-built ('out-of-tree') module was loaded  (#12)
For a more detailed explanation of the various taint flags see
 Documentation/admin-guide/tainted-kernels.rst in the the Linux kernel sources
 or https://kernel.org/doc/html/latest/admin-guide/tainted-kernels.html
Raw taint value as int/string: 4096/'G          O     '
```

1.8. Для визначення конфігурації апаратних засобів комп’ютера в ОС Linux можна використати наступні команди та системні утиліти: **lscpu** - відображення інформації про архітектуру процесора; **lspci** – виводить детальну інформацію про всі PCI-шини та підключені до них

пристрої у вигляді списку; **lsusb** — виводить інформацію про USB-шини, та підключені USB-пристрої; **lshw** — утиліта для визначення детальної інформації про апаратні засоби комп'ютера і їх конфігурацію (приклад використання: $sudo lshw -short); **hwinfo** — використовується для пошуку (опитування) всього наявного в системі апаратного забезпечення і формування відповідного звіту, який може бути використаний в роботах з технічної підтримки.

## 2. Послідовність виконання роботи

2.1. Ознайомитись з відомостями про ядро ОС Linux та описом команд і системних утиліт ОС Linux для отримання інформації про ядро та апаратні засоби комп'ютера.
2.2. Визначити версію ядра ОС Linux за допомогою команди uname.
2.3. Дослідити роботу системної утиліти sysctl в режимі отримання інформації про параметри ядра ОС Linux. Отримати інформацію про окрему групу параметрів ядра згідно варіанту.
2.4. Отримати інформацію про доступні для завантаження модулі ядра та встановити їх загальну кількість.
2.5. Отримати інформацію про завантажені модулі ядра та встановити їх кількість.
2.6. Визначити стан «забруднення» ядра ОС Linux. В разі «забрудненого» ядра декодувати стан «забруднення» за допомогою скрипту kernel-chktaint.
2.7. Дослідити роботу команд та системних утиліт для визначення конфігурації апаратних засобів комп'ютера (lscpu, lspci, lsusb та lshw/hwinfo — згідно варіанту). Дослідити можливість визначення типу комп'ютера, виходячи з конфігурації його апаратних засобів (ноутбук, десктоп, сервер).
2.8. Визначити конфігурацію апаратних засобів комп'ютера, на якому виконується лабораторна робота.
2.9. Скласти та захистити звіт з лабораторної роботи.

## 3. Варіанти завдань

| N | Група параметрів ядра | Системна утиліта для визначення конфігурації апаратних засобів |
|---|---|---|
| 1 | /proc/sys/kernel/sched_* | lshw |
| 2 | /proc/sys/vm/ | hwinfo |
| 3 | /proc/sys/dev/ | lshw |
| 4 | /proc/sys/fs/ | hwinfo |
| 5 | /proc/sys/net/ | lshw |
| 6 | /proc/sys/kernel/sched_* | hwinfo |
| 7 | /proc/sys/vm/ | lshw |
| 8 | /proc/sys/dev/ | hwinfo |
| 9 | /proc/sys/fs/ | lshw |
| 10 | /proc/sys/net/ | hwinfo |

## 4. Зміст звіту

4.1. Номер варіанту і відповідне завдання.
4.2. Результати виконання завдань по визначенню параметрів ядра ОС Linux (вказати результат та спосіб його отримання):
4.2.1. Версія ядра ОС Linux.
4.2.2. Список групи параметрів ядра зі значеннями (згідно варіанту).
4.2.3. Кількість доступних для завантаження модулів ядра.
4.2.4. Кількість завантажених модулів ядра.
4.2.5. Стан «забруднення» ядра ОС Linux.
4.3. Результати дослідження роботи команд та системних утиліт для визначення конфігурації апаратних засобів комп’ютера.

## 5. Контрольні питання

5.1. Архітектура ядра ОС Linux та її особливості.
5.2. Основні функціональні компоненти ядра ОС Linux та групи параметрів ядра, які їм відповідають.
5.3. Призначення системної утиліти systcl.
5.4. Завантажуваний модуль ядра (loadable kernel module).
5.5. Призначення та використання команд lsmod і modinfo.
5.6. Стан «забруднення» ядра ОС Linux.
5.7. Призначення команд lscpu, lspci, lsusb.
5.8. Призначення системних утиліт lshw та hwinfo.

## 6. Джерела

1. Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, 3rd edition, O'Reilly Media, 2005. - 944 p.

2. Robert Love, Linux Kernel Development, 3rd edition, Addison-Wesley Professional, 2010. - 440 p.

3. Онлайн репозиторій вихідних кодів ядра ОС Linux, www.kernel.org

4. The Linux Kernel documentation, www.kernel.org/doc/html/latest/

5. The kernel's command-line parameters (in The Linux Kernel documentation), https://www.kernel.org/doc/html/latest/admin-guide/kernel-parameters.html

6. Documentation for /proc/sys (in The Linux Kernel documentation), www.kernel.org/doc/html/latest/admin-guide/sysctl/index.html

7. Yama (in The Linux Kernel documentation), https://www.kernel.org/doc/html/latest/admin-guide/LSM/Yama.html

8. Tainted kernels (in The Linux Kernel documentation), www.kernel.org/doc/html/latest/admin-guide/tainted-kernels.html

9. kernel-chktaint, git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/plain/tools/debugging/kernel-chktaint