Zarządzanie modułami jądra Linux w Ubuntu 24.04

Cel ogólny:

Zapoznanie się z budową, zarządzaniem i wykorzystaniem modułów jądra Linux w systemie Ubuntu 24.04.

Cele szczegółowe:

- Zapoznanie się z funkcjami narzędzia modprobe i dokumentacją (man modprobe).
- Poznanie poleceń uname, lsmod, modinfo, rmmod, modprobe, insmod, depmod.
- Przeprowadzenie testów poleceń i wykonanie notatek.
- Nauka tworzenia, kompilacji i ładowania modułu jądra Linux.
- Zrozumienie różnic między modprobe a modutils (historycznie).

Wprowadzenie

Moduły jądra Linux pozwalają rozszerzać funkcjonalność systemu bez przebudowy całego jądra. W Ubuntu 24.04 używamy narzędzia modprobe do zarządzania modułami. Pakiet modutils jest przestarzały.

Przywróć pierwszą migawkę maszyny

Wymagania wstępne

- Ubuntu 24.04 z dostępem do terminala.
- Zainstalowane nagłówki jądra: `sudo apt install linux-headers-$(uname -r)`.
- Pakiet build-essential: `sudo apt install build-essential`.
- Znajomość podstaw obsługi terminala.

Tabela poleceń

Polecenie	Funkcja
lsmod	Wyświetla listę załadowanych modułów
modinfo	Informacje o module
modprobe	Ładowanie/usuwanie modułu
insmod	Wstawienie modułu z pliku .ko
rmmod	Usunięcie modułu
depmod	Aktualizacja bazy modułów

Ćwiczenie krok po kroku

1. Sprawdź dokumentację: `man modprobe`. Zapoznaj się z funkcjami oprogramowania.

Wykonaj notatkę z poniższych czynności. W systemie Ubuntu 24.04, do zarządzania modułami jądra Linux używane jest narzędzie o nazwie modprobe. Jest to standardowe narzędzie w dystrybucjach opartych na jądrze

Linux, które umożliwia ładowanie, usuwanie i zarządzanie modułami jądra. Możesz użyć polecenia modprobe w terminalu, aby ładować i usuwać moduły jądra. Pamiętaj, że korzystanie z modprobe wymaga uprawnień administracyjnych (sudo). Jeśli potrzebujesz bardziej zaawansowanych operacji na modułach jądra, warto zapoznać się z dokumentacją narzędzia modprobe oraz dostępnymi opcjami i parametrami.

2. Przetestuj polecenia uname, lsmod, modinfo, modprobe, rmmod.

a. uname -a uniksowe polecenie wypisujące informacje o systemie operacyjnym

```
sudo uname -a
```

b. Sprawdzenie czy moduł jest wbudowany

```
sudo lsmod | grep loop
```

Jeśli nic nie wyświetli, oznacza to, że loop jest częścią jądra, a nie modułem dynamicznym.

c. modinfo pobiera informacje o modułach jądra Linux, podanych w wierszu polecenia, jeśli nazwa modułu nie jest nazwą pliku, to przeszukiwany jest katalog /lib/modules/wersja

```
sudo modinfo dummy
```

d. modprobe ładuje wszystkie moduły, od których zależny jest dany moduł

```
sudo modprobe dummy
```

e. rmmod usuwa załadowany moduł

```
sudo rmmod dummy
```

W Ubuntu 24.04 (kernel 6.8) moduł dummy jest kompilowany jako wbudowany w pakiet modułów, ale nie jest dostępny jako osobny plik .ko w standardowej lokalizacji.
modprobe korzysta z bazy modułów (modules.builtin i modules.dep) i wie, że dummy jest dostępny, więc go aktywuje.

3. Załaduj i usuń moduł dummy: sudo modprobe dummy, sudo modprobe -r dummy.

```
sudo modprobe dummy
```

```
lsmod | grep dummy
```

```
sudo modprobe -r dummy
```

Efekt:

- dummy pojawi się w liście modułów po załadowaniu.
- Można go bezpiecznie usunąć.

Wniosek dla uczniów: Nie wszystkie moduły można usuwać – te wbudowane w jądro (builtin) są zawsze aktywne. Do ćwiczeń wybieramy moduły opcjonalne, np. dummy

- Co to jest? Moduł jądra tworzący wirtualny interfejs sieciowy (np. dummy0).
- Zastosowanie: Przydatny do testów sieciowych, gdy nie mamy fizycznej karty sieciowej. Można przypisać adres IP i używać w konfiguracjach.
- Bezpieczny do ćwiczeń: Tak, bo nie wpływa na realne połączenia.

4. Utwórz plik moj_modul.c z kodem modułu Hello World (poniżej kod).

Poniżej znajduje się przykładowy kod modułu:

```
#include <linux/module.h>
#include <linux/kernel.h>

int init_module(void) {
    printk(KERN_INFO "Mój moduł został załadowany!\n");
    return 0;
}

void cleanup_module(void) {
    printk(KERN_INFO "Mój moduł został usunięty!\n");
}
```

MODULE_LICENSE("GPL");

MODULE_AUTHOR("Twoje Imię");

MODULE_DESCRIPTION("Prosty moduł jądra");

5. Utwórz plik Makefile (poniżej kod) i skompiluj moduł poleceniem: make.

   Przykład Makefile

   obj-m += moj_modul.o

   all:
   ***<TAB>** make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules*

   clean:
   ***<TAB>** make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean*

   Utwórz plik Makefile (nie pisz <TAB> a go zrób)

6. Załaduj moduł: sudo insmod moj_modul.ko, sprawdź lsmod i dmesg.

Sprawdź, czy moduł moj_modul jest w jądrze:

```
lsmod | grep moj_modul
```

Zobacz komunikaty w logu:

```
dmesg | tail
```

Powinieneś zobaczyć np. *Mój moduł został załadowany*! (zależnie od kodu w moj_modul.c), insmod ładuje moduły do uruchomionego jądra

7. Skopiuj moduł do /lib/modules/…/extra, wykonaj depmod i załaduj przez modprobe.

   a. Skopiuj moduł do katalogu modułów:

```
sudo mkdir -p /lib/modules/$(uname -r)/extra
```

   b. Skopiuj swój moduł do katalog extra w katalogu modułów jądra:

```
sudo cp moj_modul.ko /lib/modules/$(uname -r)/extra/
```

   c. Zaktualizuj bazę modułów:

```
sudo depmod
```

   d. Załaduj moduł:

```
sudo modprobe moj_modul
```

e. Sprawdź, czy działa:

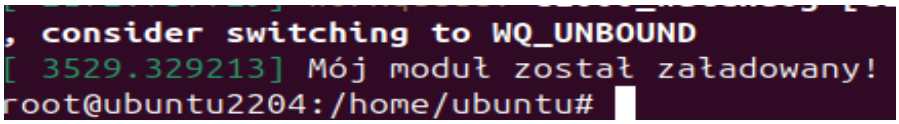
lsmod | grep moj_modul

dmesg | tail

Jeśli kompilacja przebiegnie pomyślnie, w katalogu powinien pojawić się plik moj_modul.ko, który można załadować do jądra Linux za pomocą polecenia insmod.

Upewnij się, że znajdujesz się w katalogu, w którym został skompilowany plik .ko

Możesz również użyć polecenia dmesg, aby wyświetlić logi systemowe i sprawdzić, czy moduł został poprawnie załadowany.



Sprawdź, czy kod źródłowy modułu został poprawnie skompilowany bez błędów.

Jak sprawdzić wynik kompilacji modułu w Linux?

a. Uruchom kompilację: make w katalogu, gdzie masz plik Makefile i kod modułu.
b. Jeśli wszystko jest OK:
    - W terminalu nie pojawią się błędy.
    - W tym samym katalogu pojawi się plik z rozszerzeniem .ko (np. moj_modul.ko). To jest Twój gotowy moduł jądra.
c. Jeśli są błędy lub ostrzeżenia:
    - Zobaczysz je w terminalu podczas kompilacji.
    - Błędy zwykle pokazują:
        - nazwę pliku,
        - numer linii,
        - opis problemu.
    - Przykład: moj_modul.c:12:5: error: expected ';' before '}' To znaczy, że w pliku moj_modul.c w linii 12 brakuje średnika.
d. Co zrobić, gdy są błędy?
    - Przeczytaj komunikat.
    - Otwórz plik źródłowy (.c) w edytorze.
    - Popraw wskazane miejsce.

- Zapisz plik i uruchom make ponownie.

W skrócie dla ucznia:

- Brak błędów > masz plik .ko.
- Są błędy > czytaj komunikat, popraw kod, kompiluj ponownie.

8. Usuń moduł: sudo rmmod moj_modul.

a. Aby usunąć moduł z jądra, użyj polecenia rmmod:

sudo rmmod moj_modul

Nie musisz być w katalogu z plikiem .ko, bo moduł jest już w jądrze.

Nie podajesz ścieżki do pliku – wystarczy nazwa modułu (bez .ko).

Możesz podać pełną ścieżkę do pliku .ko jako argument polecenia rmmod. Na przykład:

sudo rmmod /home/ubuntu/modul/moj_modul.ko

```
oot@ubuntu2204:/home/ubuntu/modul# sudo rmmod /home/ubuntu/modul/moj_modul.ko
mmod: ERROR: Module moj_modul is not currently loaded
```

b. Sprawdź, czy moduł został poprawnie usunięty z jądra, używając polecenia lsmod:

Polecenie lsmod – co robi i jak je używać?

- lsmod pokazuje listę wszystkich modułów, które są aktualnie załadowane do jądra Linux.
- Dzięki temu możesz sprawdzić:
  - Czy Twój moduł został poprawnie załadowany.
  - Czy został usunięty (jeśli go nie ma na liście, to znaczy, że został odłączony).

lsmod | grep moj_modul

Jeśli nic nie wyświetla – moduł został usunięty.

zgłoszenie 1

Przykłady:

Po sprawdzeniu przez prowadzącego przywróć pierwszą migawkę.

Podsumowanie

Tworzenie, kompilowanie, ładowanie i usuwanie modułów w Ubuntu to proces, który wymaga kilku kroków:

1. Napisać kod modułu (np. w pliku .c).
2. Przygotować plik Makefile do kompilacji.
3. Skompilować moduł poleceniem make – powstanie plik .ko.
4. Załadować moduł do jądra (insmod lub modprobe).
5. Sprawdzić, czy działa (lsmod, dmesg).
6. Usunąć moduł (rmmod).

Pamiętaj:

- Budowa modułów jądra może być trudniejsza w zależności od tego, co moduł robi.
- Zawsze warto sprawdzić dokumentację i materiały w sieci, jeśli pojawią się problemy.

Po wykonaniu wszystkich kroków z instrukcji **sprawdź, czy osiągnąłeś cel ogólny i cele szczegółowe** (są na pierwszej stronie instrukcji).

Jeśli czegoś nie udało się zrealizować:

- Powtórz ćwiczenie w szkole lub w domu.
- Ćwicz do momentu, aż wszystkie cele zostaną osiągnięte.