

Zarządzanie modułami.

Celem ogólnym lekcji jest zapoznanie się z budową, zarządzaniem i wykorzystaniem modułów jądra Linux w systemie Ubuntu 22.04.

Cele szczegółowe:

1. Zapoznanie się z funkcjami oprogramowania modprobe poprzez zapoznanie się z dokumentacją (man modprobe) i zrozumienie jego możliwości.
2. Zapoznanie się z pomocą systemową dotyczącą poleceń: uname, lsmod, modinfo, rmmod, modprobe, insmod i depmod.
3. Wykonanie notatek na temat działania tych poleceń oraz przeprowadzenie testów, aby lepiej zrozumieć ich zastosowanie.
4. Zapoznanie się z krokami tworzenia modułu dla systemu Ubuntu 22.04, włączając w to tworzenie pliku z kodem źródłowym, kompilację modułu za pomocą polecenia make i załadowanie modułu do jądra za pomocą polecenia insmod.
5. Przeprowadzenie testów, aby sprawdzić, czy moduł został poprawnie załadowany do jądra za pomocą polecenia lsmod.
6. Wykonanie notatek na temat sposobu usuwania modułów z jądra za pomocą polecenia rmmod oraz sprawdzenie czy moduł został poprawnie usunięty z jądra za pomocą polecenia lsmod.
7. Zrozumienie różnic między używaniem narzędzia modprobe a wcześniejszego narzędzia modutils w starszych wersjach systemu Ubuntu.
8. Zrozumienie roli modułów jądra Linux w rozszerzaniu możliwości i zasobów dostępnych w systemie operacyjnym.
9. Świadomość konieczności zapoznania się z dokumentacją i materiałami dostępnymi w sieci przed pracą z modułami jądra oraz korzystanie z narzędzi modprobe, gcc i innych odpowiednich narzędzi z pełnym zrozumieniem.

Uwaga: Powyższe cele szczegółowe są dostosowane do podanego kontekstu lekcji i mają na celu zapoznanie się z podstawowymi aspektami zarządzania modułami jądra Linux w systemie Ubuntu 22.04. W praktyce, bardziej zaawansowane zagadnienia mogą wymagać dodatkowej wiedzy i umiejętności.

Wstęp:

Moduły nie są wbudowane w jądro systemu operacyjnego. Moduły pozwalają rozszerzać możliwości i zasób dostępnych funkcji komputera. Do zarządzania modułami w systemie Linux niezbędne jest oprogramowanie modutils-x.y.z.tar.gz - w wersji odpowiedniej dla posiadanego jądra systemu operacyjnego. Po instalacji (tar zxvf modutils...) dostępne będą programy: insmod, rmmod, ksyms, lsmod, genksyms, modprobe i depmod w katalogu /sbin.

Przywróć pierwszą migawkę maszyny

Zadanie

Pakiet modutils był kiedyś używany w starszych wersjach systemów operacyjnych do zarządzania modułami jądra Linux. Jednak w nowszych wersjach systemu Ubuntu został zastąpiony przez nowsze narzędzia i rozwiązania.

A. Wykonaj notatkę z poniższych czynności. **Zapoznaj się z funkcjami oprogramowania modprobe** poprzez wpisanie w terminalu polecenia: **man modprobe**

W systemie Ubuntu 22.04, 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. Oto kilka przykładów użycia modprobe:

1. Ładowanie modułu jądra:

```
sudo modprobe nazwa_modułu
```

2. Usuwanie modułu jądra:

```
sudo modprobe -r nazwa_modułu
```

3. Wyświetlanie informacji o modułach jądra:

```
modprobe -l
```

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.

B. Wykonaj notatkę z poniższych czynności. **Zapoznaj się z pomocą systemową** na temat następujących poleceń, **przetestuj je**:

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

2. `lsmod` - wyświetla listę załadowanych modułów
3. `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`
4. `rmmod` - usuwa załadowany moduł
5. `modprobe` - ładuje wszystkie moduły, od których zależy dany moduł
6. `insmod` - ładuje moduły do uruchomionego jądra
7. `depmod` - sprawdza wszystkie zależności pomiędzy modułami

Polecenia można wykonać w terminalu, wpisując je bezpośrednio po znaku promptu.

C. Wykonaj notatkę i testowanie poniższych poleceń w terminalu poprzez wywołanie każdego z nich i obserwowanie wyniku działania. Kroki do stworzenia modułu dla Ubuntu 22.04:

1. Otwórz terminal.
2. Upewnij się, że masz zainstalowany kompilator GCC na swoim systemie. Możesz to zrobić, wykonując polecenie: `apt -y install build-essential`
3. Utwórz katalog, w którym zostanie umieszczony plik z kodem źródłowym modułu. Możesz to zrobić za pomocą polecenia `mkdir`, na przykład:

```
mkdir modul
```

4. Przejdź do katalogu, w którym został utworzony katalog dla modułu, używając polecenia `cd`, np.:

```
cd modul
```

5. Stwórz plik z kodem źródłowym modułu, np. `moj_modul.c`, za pomocą edytora tekstu, na przykład:

```
nano moj_modul.c
```

6. Wpisz kod źródłowy modułu, zgodnie z tym, co chcesz osiągnąć w module. Poniżej znajduje się przykładowy kod modułu:

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

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

```
MODULE_LICENSE("GPL");
```

```
int init_module(void)
```

```
{  
    printk(KERN_INFO "Witaj świecie.\n");  
    return 0;  
}
```

```
void cleanup_module(void)
```

```
{  
    printk(KERN_INFO "Koniec.\n");  
}
```

7. Zapisz plik `moj_modul.c` w edytorze `nano`, używając klawiszy `Ctrl + X, Y, Enter`.
8. Można skorzystać z polecenia `make` i pliku `Makefile`, który określa sposób kompilacji modułu i jego zależności. Oto zawartość przykładowego pliku `touch Makefile` (przed `make` jest `TAB`):

```
obj-m := moj_modul.o
```

```
all:
```

```
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
```

```
clean:
```

```
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

Powyższe polecenie należy wykonać w katalogu, w którym znajduje się plik z kodem źródłowym modułu. Polecenie korzysta z narzędzi `make` oraz `gcc`, a opcja `-C` wskazuje na katalog, w którym znajdują się pliki nagłówkowe jądra, niezbędne do poprawnego skompilowania modułu.

Opcja `M=$(PWD)` wskazuje na bieżący katalog, w którym znajduje się kod źródłowy modułu.

Po poprawnym wykonaniu tego polecenia, powinien powstać plik `.ko`, który zawiera skompilowany moduł.

9. Po zapisaniu pliku jako "Makefile" można skompilować moduł za pomocą polecenia make.
W przypadku braku błędów powinien pojawić się plik `moj_modul.ko`, który można załadować do jądra Linux.
10. Aby skompilować moduł za pomocą pliku Makefile, należy przejść do katalogu zawierającego plik Makefile i uruchomić w nim terminal. Następnie należy wpisać polecenie `make`, aby rozpocząć kompilację modułu.
11. Jeśli kompilacja przebiegnie pomyślnie, w katalogu powinien pojawić się plik `moj_modul.ko`, który można załadować do jądra Linux za pomocą polecenia `insmod`.
12. Aby załadować moduł do jądra systemu operacyjnego, użyj polecenia:

```
sudo insmod moj_modul.ko
```

```
root@ubuntu-VirtualBox:/home/ubuntu/modul# sudo insmod moj_modul.ko
```

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

```
root@ubuntu-VirtualBox:/home/ubuntu/modul# pwd
/home/ubuntu/modul
root@ubuntu-VirtualBox:/home/ubuntu/modul# ls
Makefile      Module.symvers  moj_modul.ko    moj_modul.mod.c  moj_modul.o
modules.order  moj_modul.c     moj_modul.mod  moj_modul.mod.o
```

13. Sprawdź, czy moduł został poprawnie załadowany do jądra, używając polecenia `lsmod`:

```
lsmod
```

```
root@ubuntu-VirtualBox:/home/ubuntu/modul# lsmod
Module          Size  Used by
moj_modul      16384  0
```

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

```
[ 3824.987613] moj_modul: loading out-of-tree module taints kernel.
[ 3824.987648] moj_modul: module verification failed: signature and/or required
key missing - tainting kernel
[ 3824.987839] Witaj świecie.
root@ubuntu-VirtualBox:/home/ubuntu/modul# dmesg
```

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

Po wykonaniu polecenia `make` w terminalu, możesz ocenić wynik kompilacji. Jeśli kompilacja przebiegła pomyślnie, nie powinny pojawić się żadne błędy ani ostrzeżenia. W takim przypadku,

w katalogu, w którym znajduje się plik Makefile, powinien pojawić się plik wynikowy modułu (z rozszerzeniem .ko).

Jeśli kompilacja zakończyła się błędami lub ostrzeżeniami, możesz sprawdzić logi błędów, które zostały wygenerowane podczas procesu kompilacji. W terminalu, w którym wykonywałeś polecenie make, powinieneś zobaczyć wypisywane na bieżąco informacje o błędach lub ostrzeżeniach. Błędy mogą zawierać informacje o konkretnych liniach kodu lub plikach, które spowodowały problem podczas kompilacji. Analizując logi błędów, możesz zidentyfikować konkretne problemy i dostosować kod źródłowy modułu, aby rozwiązać występujące błędy.

15. Aby usunąć moduł z jądra, użyj polecenia `rmmod`:

```
sudo rmmod moj_modul
```

Upewnij się, że znajdujesz się w katalogu, w którym został skompilowany plik .ko, ponieważ polecenie `rmmod` usuwa moduł z jądra, który został załadowany z pliku .ko znajdującego się w bieżącym katalogu. Jeśli nie znajdujesz się w odpowiednim katalogu, możesz podać pełną ścieżkę do pliku .ko jako argument polecenia `rmmod`. Na przykład:

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

```
root@ubuntu-VirtualBox:/home/ubuntu/modul# sudo rmmod /home/ubuntu/modul/moj_modul.ko
rmmod: ERROR: Module moj_modul is not currently loaded
```

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

```
lsmod
```

Polecenie `lsmod` wyświetla listę załadowanych modułów w jądrze systemu. Jeśli moduł został poprawnie usunięty, nie powinien być widoczny na liście.

Należy pamiętać, że przed załadowaniem modułu należy upewnić się, że wymagane zależności są już załadowane, co można sprawdzić za pomocą polecenia `lsmod`.

To są ogólne kroki do stworzenia, skompilowania, załadowania i usunięcia modułu w systemie Ubuntu 22.04. Pamiętaj, że budowa modułów jądra może być skomplikowana i wymagać dodatkowych działań w zależności od konkretnych potrzeb i zależności modułu. Przed rozpoczęciem pracy z modułami jądra zawsze warto zapoznać się z odpowiednią dokumentacją oraz materiałami dostępnymi w sieci.

```
zgłoszenie 1
```

Po sprawdzeniu przez prowadzącego przywróć pierwszą migawkę.

Podsumowanie:

Po wykonaniu wszystkich czynności z powyższej instrukcji przeczytaj ponownie z zrozumieniem cel ogólny i cele szczegółowe, które znajdują się na pierwszej stronie instrukcji. Jeżeli one zostały niezrealizowane to powtarzaj wykonanie tej instrukcji w szkole lub/i w domu do momentu zrealizowania.