

Linux - jak to zrobić , moje wskazówki autorskie , Leszek Gorzelnik , Kraków 2002-2005

Linux - jak to zrobić

- [Dodanie ścieżki z bibliotekami np. Kylix2 do środowiska](#)
- [Mysz](#)
- [DVD](#)
- [RPM](#)
- [Kompilacja jądra](#)
- [Adresy IP oraz MAC na interface'ch](#)
- [NETWORK](#)

Dodanie ścieżki z bibliotekami np. Kylix2 do środowiska

```
echo $LD_LIBRARY_PATH
```

przejsć do katalogu z bibliotekami i wykonać:

◇

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:`pwd` ◇
```

dotyczy to tylko otwartej sesji. ◇

Po skompilowaniu Kylix2 należy utworzyć skrypt z określeniem

LD_LIBRARY_PATH w którym znajdują się biblioteki Kylix'a

Przy kompilacji programu (build) należy w opcjach Project->options

zaznaczyć Build with runtime packages w zakładce "packages"

Mysz

◇

Mandrake sam rozpoznaje że masz mysz z kółkiem. Red Hat niestety nie.

Aby uaktywnić kółko w myszy musisz w pliku XF86Config-4 w /etc/X11/

dokonać wpisu w sekcji :

```
Section "Input Device"
```

```
Identifier "Mouse0"
```

```
-----
```

```
Option "Protocol" "IMPS/2"
```

```
^
```

zamiast PS/2

DVD

Po instalacji np. Ogle fajnego programu umożliwiającego odtwarzanie filmów DVD musisz utworzyć link symboliczny do urządzenia DVD.

```
ln -s /dev/hdc /dev/dvd
```

gdzie /dev/hdc to twój napęd DVD

RPM

rozszerzeniem .rpm oznaczane są pakiety oprogramowania instalowanego prawie automatycznie do systemu linux.

Zainstalować pakiet możesz rozkazem:

```
rpm -Uvh pakiet.rpm
```

lub

```
rpm --nodeps --force -U *.rpm
```

instalowane są wszystkie pakiety z katalogu w zasadzie wykonywany jest upgrade

```
rpm --rebuilddb
```

wykonywana jest odbudowa listy pakietów zainstalowanych w systemie

opcje:

--percent pokazuje procent wykonania instalacji

-v pokazuje nazwe instalowanego pakietu

-h pokazuje postęp instalacji wypełniając liniijkę # 'em

`rpm --nodeps --force -U -h -v *.rpm` instaluje wszystkie pakiety z katalogu bez sprawdzania zależności oraz bezwzględnie zastępując stare zainstalowane

Odinstalować natomiast go możesz :

`rpm -e pakiet` tutaj podajemy nazwę bez rozszerzenia

Kompilacja jądra

Każda dystrybucja zawiera już skompilowane jądro systemu. Kompilacja jego jest uniwersalna. Nieraz musimy jednak dodać dodatkowe usługi tak by zostały one wkompileowane do wnętrza jądra lub aby system był zdolny do korzystania z nich . Procedura wbrew pozorom jest bardzo prosta lecz czasochłonna. Każdy nawet niedoświadczony użytkownik systemu Linux może to wykonać . Oczywiście jest to że wszelkie operacje wykonujemy jako użytkownik root . Dobrym rozwiązaniem jest kompilowanie nowego jądra pod inną nazwą niż obecnie używana wersja . Daje nam to możliwość uruchamiania tego samego systemu z różnymi wersjami jąder o różnych właściwościach co w systemie Windows jest nieosiągalne !

Procedura kompilacji własnego jądra jest następująca:

1. Skopiuj źródła jądra systemu do katalogu `/usr/src/linux-(wersja)` .
2. Przejdź do tego katalogu i uruchom `make mrproper` - komenda ta czyści pozostałości z poprzedniej konfiguracji gdyby taka wcześniej miała miejsce .
3. Uruchom konfigurację jądra - wykonuje się to poleceniem `make xconfig` - uruchamia się konfigurator graficzny . Wszystkie parametry musimy ustawić ręcznie . Metoda ta jest wygodna do poprawiania już istniejącej konfiguracji .

Konfiguracja jest zawsze zapisywana w pliku `.config` . Możemy również używać

komend `make menuconfig` lub `make config` . Obie komendy pracują w trybie

tekstowym. Pierwsza z nich uruchamia konfigurator w postaci podświetlanego MENU podobnie jak w trybie graficznym poprzednia komenda pozwala na ręczną konfigurację jądra. Druga natomiast udostępnia nam potokowy konfigurator w którym odpowiadając na kolejne pytania tworzymy plik konfiguracyjny .

W przypadku nowej konfiguracji lepszym rozwiązaniem jest zastosowanie komendy

półautomatycznej konfiguracji. Wykonujemy to poleceniem `make oldconfig` .

4. Uruchom `make dep` .
5. uruchom `make clean` - polecenie to robi nam porządek w katalogach. Usuwa niepotrzebne już pliki - obiekty z rozszerzeniem `.o` .
6. należy teraz spakować zbudowane jądro do pliku z którego system będzie go wczytywać do pamięci . Wykonujemy to na dwa sposoby.

Pierwszy z nich robi mniejsze upakowanie - wykonuje to komenda `make zImage` .

Drugi robi większe upakowanie pliku jądra tak by zmieściło się na dyskietce.

Wykonuje to komenda *make bzImage* .

Po skompilowaniu jądra w katalogu */usr/src/linux-(wersja)/arch/i386/boot* pojawi się plik *zImage* lub *bzImage* zawierający nasze nowe jądro systemu.

7. Do prawidłowej pracy systemu z nowym jądrem konieczne jest wygenerowanie niezbędnych modułów . W tym celu należy wykonać polecenie *make modules* .

Wszystkie czynności związane z generacją nowego jądra trwają bardzo długo !

Należy się na to przygotować . Generacja modułów nie zawsze przebiegnie pomyślnie.

W przypadku wystąpienia błędu należy przeanalizować ostatnie komunikaty i określić czego dotyczy błąd. Błędy zwykle dotyczą jakichś urządzeń . Musimy wrócić do punktu 3.) i poprawić istniejącą konfigurację wydając komendę *make xconfig* lub

make menuconfig . Koniecznie należy wyłączyć te urządzenia podczas konfiguracji z wbudowania ich modułów w jądro podając opcję *<m>* .Oznaczać to będzie że moduły tych urządzeń będą ładowane oddzielnie podczas startu systemu lub ręcznie. Możemy też zrezygnować całkiem z ich obsługi przez jądro. Podajemy wówczas opcję *<n>* .

Dzieje się tak dlatego iż źródło programu opisującego moduł danego urządzenia nie pasuje do naszego sprzętu. Możemy również próbować poprawić taki moduł lecz jest to wyższa szkoła jazdy.

Jako przykład mogę podać moje doświadczenia z kompilacją Linux Mandrake 8.2

tak aby przystosować go współpracy z softmodemem Pentagram.

Wymagało to przekompilowania jądra w wersji 2.4.3-20mdk . Podczas kompilacji wystąpiły błędy wskazujące na urządzenia SCSI DC395 ,AIC7xxx,old AIC7xxx.

Musiałem wyłączyć te urządzenia z obsługi ustawiając opcję *<n>* .

Uwaga!

W przypadku kiedy kompilujemy pracującą obecnie wersję jądra należy zabezpieczyć cały katalog */lib/modules/(wersja)* zawierający funkcjonujące moduły.

8. Po bezbłędnym wygenerowaniu modułów musisz je zainstalować .

Robimy to wykonując polecenie instalacji *make modules_install* .

9. Uruchamiamy następnie uaktualnienie powiązań modułów komendą

depmod -a

10. Na zakończenie kopiujemy nowe jądro *zImage* lub *bzImage* do katalogu

/boot linkując je pod właściwą nazwę taką jaka jest używana pliku konfiguracyjnym Grub czy Lilo. standardowo jest to *vmlinuz* lub *vmlinuz-(wersja)*. Dodatkowo należy przekopiować do katalogu */boot* plik *System.map* z katalogu */usr.src/linux-(wersja)* zmieniając nazwę na

System.map-(wersja).

11. Ostatnim krokiem jest przeładowanie komputera. System uruchomi nam się z nowym naszym jądrem.

Sprawdzenie wersji jądra możemy dokonać otwierając okno terminala i wydając

polecenie `uname -a`

Adresy IP oraz MAC na interface'ch

◊#dodawanie adresu secondary

```
ip address add 10.37.10.190/20 brd + dev eth0 label eth0:sec
```

lub jako podstawowy dodatkowy

```
ip address add 10.37.10.190/20 dev eth0
```

#usuwanie adresu secondary

```
ip address del 10.37.10.190/20 brd + dev eth0 label eth0:sec
```

#usuwanie wszystkich adresów z interface'u

```
ip address flush dev eth0
```

#pokazywanie wszystkich adresów przypisanych do interface'ów

```
ip address show
```

#dodawanie trasy do głównej tablicy routingu

```
ip route add 10.37.10.190/20 via 10.37.100.252
```

#zmiana adresu MAC na interface'ie oraz adresu IP

```
ifconfig eth0 down
```

```
ifconfig eth0 hw ether 32:42:53:04:36:09 arp
```

```
ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

```
ifconfig eth0 up
```

◊

NETWORK

◊System podnosząc interface eth0 czyta plik:

```
/etc/sysconfig/network-scripts/ifcfg-eth0
```

w którym widnieją wpisy:

```
DEVICE=eth0
```

```
BOOTPROTO=static
BROADCAST=192.168.0.255
IPADDR=192.168.0.1
NETMASK=255.255.255.0
NETWORK=192.168.0.0
ONBOOT=yes
HWADDR=32:42:53:04:36:09
```

W pliku `/etc/sysconfig/network` widnieją wpisy określające bramę :

◇

```
NETWORKING=yes
GATEWAY=192.168.4.1
GATEWAYDEV=eth0
```

Serwery DNS ustawia się w pliku `/etc/resolv.conf`

```
search localhost
nameserver 62.233.128.17
nameserver 213.77.115.28
```

Leszek Gorzelnik @ Kraków 2005