

Open SuSE - instrukcje

© 3bird Projects 2024, <http://edukacja.3bird.pl>

Informacje ogólne

Istnieją dwa wydania systemu *openSUSE*:

- **Tumbleweed** – szybko rozwijająca się wersja testowa zawierająca najnowsze wersje pakietów, nie jest jednak gruntownie przetestowana pod kątem stabilności i bezpieczeństwa;
- **Leap** – ustabilizowana edycja zawierająca sprawdzone, starsze pakiety (dla serwerów).

Instalacja - partycje

Instalacja zwykła

Możemy zaakceptować zaproponowaną konfigurację partycji lub ją poprawić:

(*Zaawansowany partycjoner / Rozpocznij z aktualną propozycją / Dyski twarde / Modyfikuj*)

Partycja	Rozmiar	Opis	Typ partycji	Punkt montowania
/dev/sda1	260.00 MiB	Partycja startowa BIOS / UEFI	FAT	/boot/efi
/dev/sda2	max - 4 GiB	Partycja systemowa natywna	BtrFS	/
/dev/sda3	4.00 GiB	Partycja wymiany swap	swap	swap

Partycję UEFI zakładamy tylko w przypadku, gdy nasz system (komputer) obsługuje UEFI. Jeśli korzystamy ze standardowego starego BIOS (MBR), nie zakładamy tej partycji. Nie ma także obowiązku tworzenia partycji */home* (jest to opcjonalne i zalecane dla bardziej świadomych użytkowników, którzy wiedzą po co się ją zakłada). Należy także pamiętać, że w starszych systemach (tylko BIOS) na dysku */dev/sda* można utworzyć **łącznie** tylko 4 partycje podstawowe. Jeśli jest potrzeba utworzenia większej ilości, należy jako czwartą utworzyć partycję rozszerzoną, w ramach której można dodać kolejne partycje.

Uwaga: Jeśli chcemy zainstalować Linuksa jako drugi równoległy system na dysku, należy wydzielić osobną partycję o rozmiarze minimum 70GB (najpierw instalujemy Windowsa, potem Linuksa).

Instalacja w VirtualBox

Rozmiar dysku w *VirtualBox* ustawiany na dynamiczny 70GB. Podczas instalacji tworzymy własny rozkład partycji (*Zaawansowany partycjoner / Rozpocznij z aktualną propozycją / Dyski twarde (usuwamy propozycje partycji linuxowych) / /dev/sda / Partycje / Dodaj partycję / Partycja podstawowa*):

Partycja	Rozmiar	Opis	Typ partycji	Punkt montowania
/dev/sda1	63.00 GiB	Partycja systemowa natywna	BtrFS	/
/dev/sda2	4.00 GiB	Partycja wymiany swap	swap	swap
/dev/sda3	1.00 GiB	Pliki użytkownika (<i>Data&ISV</i>)	XFS lub ext4	/home

Uwaga: Nie ma obowiązku tworzenia partycji */home* (jest to opcjonalne i zalecane dla bardziej świadomych użytkowników). Rozmiar partycji *swap* powinien wynosić dwukrotność naszej pamięci RAM (jeśli mamy 2GB RAM-u, to *swap* powinien mieć rozmiar 4GB).

Inne opcje:

- Zalecane środowisko graficzne: *KDE Plasma*.
- Pomiąć tworzenie użytkowników (będzie tylko konto administratora: *root*).
- Otworzyć port SSH (22).
- Uwaga! Zachowaj płytę DVD z Linuksem lub jego obraz **.iso* na dysku. Będzie on potrzebny przy aktualizacji systemu (musi być dokładnie ta sama wersja!).

Obsługa terminala

W systemie *Linux* może być wiele różnych terminali (konsoli), które są odpowiednikami „*Wiersza poleceń*” (cmd) w *Windows*. Polecenia możemy wydawać bądź na koncie zwykłego użytkownika, bądź na koncie administratora „*root*”. O tym, na jakim obecnie koncie jesteśmy, świadczy znak zachęty:

```
# jakieśPolecenie (to jest konto roota)
$ jakieśPolecenie (to jest konto zwykłego użytkownika)
> jakieśPolecenie (to jest konto zwykłego użytkownika)
:~> jakieśPolecenie (to jest konto zwykłego użytkownika)
```

Możemy także wydać polecenie, które jednoznacznie nam wskaże konto, na którym jesteśmy obecnie zalogowani:

```
:~> whoami
```

Instalacja pakietów

Repozytorium *OSS* (oficjalne, **O**pen-**S**ource-**S**oftware) i *non-OSS* (nieoficjalne) nigdy nie ulega zmianie, więc nie powinny mieć włączonej opcji odświeżania. Dodawanie i zarządzanie repozytoriami:

1. Metoda graficzna Gnome: *Settings / YaST / Software / Software Repositories*
2. Metoda graficzna KDE: *Applications / Settings / Administrator Settings (YaST) / Software / Software Repositories*
3. Metoda tekstowo-graficzna (tzw. ncurses):
yast
Software / Software Repositories

4. Metoda tekstowa:
zypper addrepo -f http://jakisAdres.org „Repozytorium Zbyszka”
zypper addrepo /home/zbychu/repo „Lokalne repo Zbycha”
zypper dup --from packman --allow-vendor-change (zmiana wydawcy pakietów systemowych na packman)
zypper lr (wykaz repozytoriów; „lr” pochodzi od „list repositories”)
zypper lr -P (pokazuje priorytety przypisane do repozytoriów)
zypper repos (wykaz repozytoriów)
zypper dup --allow-vendor-change --force-resolution ()
zypper mr -d numerNaLiście (deaktywacja repozytorium, bezpieczniejsze niż usuwanie)
zypper rr numerNaLiście (usuwanie repozytorium; „rr” pochodzi od „removerepo”)
zypper refresh (lub „ref”; odświeża wszystkie aktywne repozytoria)
zypper mr -e numerNaLiście (aktywuje repozytorium)

Info: Jeśli jakieś repozytorium nie jest aktualne, należy je usunąć.

Adresy repozytoriów:

```
Główne → http://download.opensuse.org/update/leap
OSS → http://download.opensuse.org/distribution/leap/15.4/repo/oss/
OSS Update → http://download.opensuse.org/update/leap/15.4/oss/
NON-OSS → http://download.opensuse.org/distribution/leap/15.4/repo/non-oss/
NON-OSS Update → http://download.opensuse.org/update/leap/15.4/non-oss/
Lokalne → cd:///?devices=/dev/disk/by-id/ata-VBOX_CD-ROM_VB0-01f003f6
```

Podstawowe polecenia

Instalowanie programów:

```
# zypper in nazwaPakietu
lub
```

```
# yast -install nazwaPakietu
```

Usuwanie pakietów:

```
# zypper rm nazwaPakietu
```

Czy pakiet jest zainstalowany:

```
# zypper se nazwaPakietu („se” pochodzi od „search”)
```

Uruchamianie usług:

```
# service nazwaSerwera start
```

Sprawdzanie statusu usług:

```
# service nazwaSerwera status
```

Edytowanie konfigów:

```
# nano /etc/nazwaFolderuSerwera/plikKonfiguracyjnySerwera.conf
```

Update systemu:

```
# zypper dup
```

Restart systemu:

```
# systemctl daemon-reexec
```

Sieć

W systemie *Linux* istnieje kilkanaście sposobów na konfigurację sieci, jest wiele aplikacji (często nawzajem się wykluczających). Oto kilka z nich:

1. Metoda graficzna (Gnome): *Activities / Settings / Network / Wired* → konfiguracja / IPv4.

2. Metoda graficzna (Gnome): *Activities / YaST / System / Network Settings*.

3. Metoda graficzna (KDE): *Pulpit: KinfoCenter / Network Information / Network Interfaces*

4. Metoda graficzno-tekstowa:

```
# yast
```

System / Network Settings

5. Metoda tekstowa:

```
# ip address add 192.168.17.120 dev enp0s3 (tymczasowe, znika po restarcie)
```

6. Metoda tekstowa:

```
# nano /etc/sysconfig/network/config
```

```
# nano /etc/sysconfig/network/dhcp
```

```
# nano /etc/sysconfig/network/ifcfg-enp0s3
```

```
BOOTPROTO=none (inne opcje: DHCP)
```

```
DEVICE=enp2s0
```

```
IPADDR=192.168.17.10 (statyczny adres IP)
```

```
NETMASK=255.255.255.0
```

```
NETWORK=192.168.17.0
```

```
ONBOOT=yes (interfejs uruchamia się przy starcie systemu)
```

Sprawdzanie IP (metody tekstowy):

```
# ip addr
```

lub

```
# ip addr show
```

lub

```
# ip a
```

lub

```
# ip a show enp0s3
```

Uruchamianie / zamykanie interfejsów sieciowych (metody tekstowe):

```
# service NetworkManager start | stop | restart | status (Uwaga! Wielkość liter ma znaczenie!)
```

lub

```
# ip link set enp0s3 down | up
```

lub

```
# systemctl start | stop | restart | status network  
lub  
# ifdown enp0s3 (alternatywa wobec NetworkManager)  
# ifup enp0s3  
lub  
# ifconfig enp0s3 down | up (domyślnie nie jest zainstalowany)
```

Sprawdzenie tablicy routingu (metoda tekstowa):
ip route show

Dodawanie statycznej tablicy routingu (metoda tekstowa):
ip route add 192.168.17.0/24 via 192.168.17.1 dev enp0s3 (wszystkie pakiety przeznaczone dla sieci 192.168.17.0 mają przechodzić przez router o adresie 192.168.17.1 i interfejsie enp0s3)

Dodawanie domyślnej bramy (metoda tekstowa):
ip route add default via 192.168.17.1 (domyślnie każdy ruch ma przechodzić przez 192.168.17.1)

Usuwanie wpisu w tablicy routingu (metoda tekstowa):
ip route del 192.168.17.0/24

Firewall

W nowszych wersjach *openSUSE* należy wyłączyć firewalla w trybie graficznym: *Application Launcher / Applications / System / Administrator Settings YaST / Security and Users / Firewall* (After writing configuration: STOP; After reboot: Do not start).

Uwaga: Jeśli nie posiadasz "Administrator Settings YaST", zainstaluj go:

```
# zypper in yast2-control-center  
# zypper in yast2-firewall
```

Firewall można także wyłączyć w trybie tekstowym:

```
# service firewalld stop  
lub:  
# systemctl stop firewalld  
Aby sprawdzić, czy jest wyłączony:  
# systemctl status firewalld (stan firewalla powinien być „inactive”)
```

Jeśli posiadasz **starszą wersję** systemu *openSUSE*, musisz wyłączyć serwis *SuSEfirewall2*, który korzysta z reguł *iptables* (jest po prostu nakładką na *iptables*).

Wyłączenie *SuSEfirewall2*:

```
# service SuSEfirewall2 stop  
# service SuSEfirewall2 status
```

Edytowanie pliku konfiguracyjnego *SuSEfirewall2*:

```
# kwrite /etc/sysconfig/SuSEfirewall2
```

Istnieją trzy strefy monitorowane przez firewalla:

```
FW_DEV_EXT="any wlan0" (sieć niezauważalna zewnętrzna, Internet)  
FW_DEV_INT="eth0 wlan1" (sieć wewnętrzna zauważalna)  
FW_DEV_DMZ="wlan2" (strefa sieci wewnętrznej dostępna dla sieci zewnętrznej)
```

Wczytanie wprowadzonych zmian (przez roota):

```
# /sbin/SuSEfirewall2
```

Nano - edytor konfigów

Nano to najbardziej uniwersalny edytor tekstów, a więc i plików konfiguracyjnych. Funkcjonuje w trybie tekstowym, dzięki czemu mamy do niego dostęp nawet wtedy, gdy środowisko graficzne nie uruchomiło się.

```
# nano /etc/nazwaPliku.conf (edycja pliku konfiguracyjnego)
```

Zapisanie zmian w pliku: **Ctrl + O**

Wyjście z *nano*: **Ctrl + X**

Serwer Samba

Przed przystąpieniem do pracy, upewnij się, że masz w systemie konto zwykłego użytkownika (oprócz konta administratora **root**). Tylko na koncie zwykłego użytkownika możemy udostępniać foldery (foldery utworzone na koncie */root* nie mogą być udostępniane!).

Jeśli nie masz zwykłego konta, utwórz je teraz:

```
# useradd -g users -G wheel,lp -m -d /home/nazwaUżytkownika -s /bin/bash -c 'twoje-  
Imię Nazwisko' nazwaUżytkownika
```

Tworzymy hasło dla utworzonego użytkownika:

```
# passwd nazwaUżytkownika
```

...i przelogowujemy się na tego użytkownika:

```
# su nazwaUżytkownika
```

Teraz tworzymy sobie folder, który chcemy udostępnić, a następnie tworzymy w nim plik z naszym imieniem i nazwiskiem:

```
~> mkdir /home/nazwaUżytkownika/nazwaUdostępnionegoFolderu
```

```
~> touch /home/użytkownik/nazwaUdostępnionegoFolderu/plik.txt (tworzymy pusty plik)
```

```
~> echo "Imię Nazwisko" > /home/użytkownik/nazwaUdostępnionegoFolderu/plik.txt
```

```
~> whoami
```

```
~> su (jeśli nie jesteś root'em, to przełącz się na roota)
```

```
# whoami
```

```
# zypper dup (robimy update systemu)
```

```
# zypper in samba gedit
```

```
# ip address (sprawdzamy swoje bieżące IP)
```

```
# gedit /etc/hosts (wewnątrz pliku musi być wpis: twojeIP twojaNazwaKomputera, np.)
```

```
127.0.0.1 localhost.localdomain localhost
```

```
192.168.6.112 linux12.zs4 linux12
```

```
# echo "linux12" > /etc/hostname (zamiast „linux12” wpisz swoją nazwę komputera)
```

Konfigurujemy Sambę. Zamiast edytora "gedit" można użyć edytora "nano":

```
# gedit /etc/samba/smb.conf
```

Uwaga: W pliku konfiguracyjnym należy usunąć wszystko, co się w nim znajduje i wpisać tylko to, co znajduje się poniżej:

```
[global]
```

```
server role = standalone server
```

```
netbios name = nazwaTwojegoKomputera (np. linux14, nie więcej niż 15 znaków, bez odstępów)
```

```
server string = Serwer plikow
```

```
workgroup = nazwaTwojejGrupyRoboczej (np. ZS4, ZS6, itp.)
security = USER
map to guest = Bad User
usershare allow guests = Yes
disable netbios = Yes (opcja deaktywuje SMB1)
protocol = SMB3 (Windows 10 obsługuje tylko tę wersję protokołu, Windows 8 obsługuje SMB2)
```

[publiczny]

```
comment = Folder do użytku publicznego
path = /home/nazwaUżytkownika/udostępnianyFolder
browseable = Yes
guest ok = Yes
read only = Yes
hosts allow = all
create mask = 0755
directory mask = 0777
```

```
# testparm (system sprawdza błędy w pliku konfiguracyjnym Samby)
# systemctl stop firewalld (dla pewności wyłączamy firewall)
```

Na koniec, instalujemy w Linuksie pakiet **WSDD** (od 2020 roku, jest to domyślny protokół do wykrywania SMBv2/SMBv3): <https://software.opensuse.org/ymp/network/15.4/wsdd.ymp>
Jeśli wystąpi błąd podczas instalacji (nie znajduje ścieżki do repozytorium), należy uaktualnić cały system:

```
# zypper dup
```

Jeśli nadal występują błędy, można spróbować pakietów społecznościowych:

<https://software.openSUSE.org/package/wsdd> (preferowany pakiet społecznościowy „Show community packages” od **lemmy04**).

Po instalacji, konfigurujemy WSDD:

```
# nano /etc/sysconfig/wsdd
```

```
WSDD_WORKGROUP="ZS4" (teoretycznie WSDD powinno pobierać nazwę grupy roboczej z Samby, ale w openSUSE z jakiś przyczyn to nie działa, więc wpisujemy tutaj nazwę swojej grupy roboczej, np. ZS4 / ZS6, itp.)
```

```
WSDD_ARGS="--ipv4only" (komunikacja ma się odbywać za pomocą IPv4)
```

Następnie wyłączamy protokół IPv6 w *openSUSE* (kliknij prawym przyciskiem myszy w ikonkę sieci obok zegara, zakładka *IPv6: Ignored*). Wyłączamy go także w systemie *Windows* (Panel sterowania: Duże ikony / Centrum sieci i udostępniania / Ethernet / Właściwości).

I na koniec, na **klientach windowsowych**, włączamy logowanie gości:

Uruchom: **gpedit.msc**

Konfiguracja komputera / Szablony administracyjne / Sieć / Stacja robocza LANman:

Włącz niezabezpieczone logowanie gościa: Włączone.

W systemie Windows należy także **wyłączyć** protokół SMB1 (jeśli jest włączony):

Panel sterowania: Duże ikony / Programy i funkcje / Włącz lub wyłącz funkcje systemu Windows / Obsługa udostępniania plików SMB 1.0/CIFS → Wyłącz;
SMB Direct → Włącz.

Upewnij się także, że są włączone w systemie Windows następujące usługi w trybie automatycznego opóźnionego uruchamiania (Win+R: *services.msc*):

- Grupowanie sieci równorzędnej;
- Host dostawcy odnajdywania funkcji (to jest WSDD);
- Host urządzenia UPnP;
- Menedżer tożsamości sieci równorzędnej;
- Odnajdywanie SSDP (SSDP Discovery);
- Połączenia sieciowe;
- Protokół rozpoznawania nazw równorzędnych;
- Publikacja zasobów odnajdywania funkcji (to jest WSDD);

Teraz chwila prawdy. Sprawdzamy, czy to wszystko będzie działać:

```
# systemctl start smb nmb wsdd (uruchamiany serwer)
```

Info: Można także na stałe uaktywnić inicjację Samby przy starcie systemu:

```
# systemctl enable smb nmb wsdd
```

Dla pewności sprawdzamy jeszcze, czy nie ma żadnych błędów:

```
# ip address (czy nasze IP wciąż jest takie samo?)
```

```
# systemctl status smb nmb wsdd firewalld (wszystko powinno mieć status ACTIVE oprócz firewalla, który powinien być wyłączony i mieć status „inactive”; jeśli pojawią się błędy, uważnie przeczytaj komunikat i popraw)
```

Serwer plików proFTPd

1. Instalujemy edytor tekstu *nano* (jeśli jeszcze tego edytora nie mamy):

```
~> whoami (kim jestem?)
```

```
~> su (jeśli nie jestem rootem, przełączam się na konto roota)
```

```
# whoami
```

```
# zypper in nano
```

2. Sprawdzamy ustawienia naszego DNS:

```
# cat /etc/resolv.conf
```

```
search nazwaNaszejDomeny (tu powinna być nazwa naszej domeny, np. nazwa firmy / szkoły, np. ZS4, ZS6, itp.)
```

```
domain nazwaNaszejDomeny (tu powinna być nazwa naszej domeny, np. nazwa firmy / szkoły)
```

```
nameserver 8.8.8.8
```

```
nameserver 8.8.4.4
```

Info: Powyższa sekcja tworzona jest automatycznie (dane pobierane są z routera). Jeśli nie ma wpisanej domeny lub publicznych numerów DNS, to znaczy że masz źle ustawiony router i możesz skorygować to w tym pliku za pomocą polecenia „**nano /etc/resolv.conf**”.

```
# nano /etc/hostname (tu wpisujemy nazwę komputera, np. linux18)
```

```
# reboot (robimy restart Linuksa)
```

```

:~> whoami    (Czy jestem rootem? Jeśli nie, przeloguj się na roota)
# hostname   (sprawdzamy, czy nazwa komputera została ustawiona)
# ip address (sprawdzamy na Linuksie nasze IP i zapisujemy na kartce)
# nano /etc/hosts (poniższa linia ma mieć dokładnie taką zawartość)
127.0.0.1      localhost.localdomain      localhost
Info: Poniżej tej linii wpisujemy nasze dane (nasze IP), np.:
192.168.6.numer   naszaNazwaKomputera.naszaDomena   naszaNazwaKomputera
np.:
192.168.6.3      asus.dom                                     asus
(poniżej mogą znaleźć się wpisy dotyczące innych komputerów w naszej sieci)

```

3. Tworzymy foldery, które chcemy udostępnić:

```

# mkdir      /home/nazwaFolderuFTP
# mkdir      /home/nazwaFolderuFTP/pub
# touch      /home/nazwaFolderuFTP/pub/nazwisko.txt    (tworzymy pusty plik)
# echo       "Imię Nazwisko" > /home/nazwaFolderuFTP/pub/nazwisko.txt
# mkdir      /home/nazwaFolderuFTP/uploads

```

Info: Poniżej wpisujesz np. *tomek.users*:

```

# chown -R   mojaNazwaUzytkownika.users /home/nazwaFolderuFTP/
# chmod -R   777 /home/nazwaFolderuFTP/

```

4. Instalujemy serwer:

```

# zypper dup (na wszelki wypadek aktualizujemy cały system)
# zypper in proftpd

```

5. Wyłączamy firewalla.

```

# service firewalld stop (zob. też instrukcja w poprzednim rozdziale!)
# service firewalld status (sprawdzamy, czy rzeczywiście wyłączony --> inactive)

```

6. Edytujemy i poprawiamy plik konfiguracyjny:

```

# nano /etc/proftpd/proftpd.conf

```

Info: Zjeżdżamy w dół, szukamy sekcji **Anonymous**. Należy odkomentować wpis <Limit LOGIN>, czyli usunąć hash'e (#). Ma być tak (i nie inaczej!), sprawdź dokładnie każdą linię:

```

<Anonymous /home/mojFolderFTP>
<Limit LOGIN>
    Order Deny,Allow
    Deny from zlaStrona.net
    Allow from All
</Limit>
UserAlias anonymous ftp

```

Uwaga: Poniżej powyższych wpisów znajduje się dalsza część kodu konfiguracyjnego zakończonego na samym dole elementem </Anonymous>. Nie zmieniamy nic więcej.

7. Uruchamiamy i testujemy:

```

# service proftpd start
# service proftpd status

```

Info: Czy pojawia się słowo „active” czy „failed”? Jeśli jest komunikat o braku użytkownika "ftp" i grupy "ftp" - musimy takiego użytkownika i grupę stworzyć za pomocą poleceń "useradd" oraz "groupadd". Po analizie błędów i ich poprawieniu:

```

# service proftpd restart
# service proftpd status

```

8. Chwila prawdy, testujemy serwer, najpierw z samego Linuksa:

```

# ftp 192.168.6.numerIP (numer IP twojego Linuksa)

```

```
Name: anonymous
```

```
Password: adres@email.pl
```

```
ftp> dir
```

```
ftp> cd pub
```



```
ftp> dir
ftp> get nazwisko.txt
ftp> exit
```

Info: Jeśli powyższe działa, próbujemy połączyć się z naszym serwerem z poziomu Windowsa:
C:\> **ftp 192.168.6.numerIP_Linuxa**

Serwer wydruku CUPS

1. Instalacja CUPS:

```
# zypper in cups
```

2. Instalowanie dodatkowych sterowników do drukarek:

```
# zypper in hplip
# zypper in gutenprint
# zypper in OpenPrintingPPDs
# zypper in ghostscript
# zypper in cups-backends
# zypper in cups-filters
# zypper in epson-inkjet-printer (tylko jeśli masz drukarki Epson)
```

3. Edytowanie pliku konfiguracyjnego:

```
# kwrite /etc/cups/cupsd.conf (można także użyć innych notatników: gedit, nano, .itp)
```

4. Uruchamianie / restartowanie serwera CUPS:

```
# service cups start
# service cups restart
# service cups status
```

5. Konfigurujemy technologię "IPP Everywhere" (dynamiczne tworzenie sterowników w locie):

```
# nano /etc/cups/cups-browsed.conf
```

Info: Muszą być odkomentowane następujące opcje:

```
CreateIPPPrinterQueues Yes
CreateIPPPrinterQueues Everywhere
CreateIPPPrinterQueues Driverless
```

Uruchamiamy serwisy:

```
# service cups-browsed start
# service cups-browsed status
# service avahi-daemon start
# service avahi-daemon status
```

Uwaga: Sterowniki powinny zostać wygenerowane po kilku minutach do folderu `/etc/cups/ppd`.

6. Dodawanie drukarki (w przeglądarce internetowej):

```
http://localhost:631
```

Protokół: Przy dodawaniu drukarki wpisujemy protokół / port. Jaki? O tym mówi instrukcja obsługi drukarki (ewentualnie wujek *Google!*). W naszym konkretnym przypadku będzie to `"ipp://192.168.17.10:631/ipp/"`.

Sterownik: Sterownik przetestowany w praktyce nosi nazwę `"HP ColorLaserJet M253-M254 - IPP Everywhere"` - jeśli nie będzie go na wykazie sterowników, pobierz go ze strony `http://edukacja.3bird.pl` w formie pliku `*.ppd`.

Uwaga: Nie działa sterownik o nazwie `"HP ColorLaserJet M253-M254 - Postscript"`.

7. Na końcu: testujemy wydruk! Korzystając z dowolnego edytora tekstu w systemie *openSUSE*, wydrukuj swoje imię i nazwisko.

Usługa Cron

Zanim rozpoczniesz pracę, musisz wiedzieć, o co w ogóle „cho” z tą usługą *Cron*, do czego ona służy.

1. Instalacja serwera *Cron*:

```
# zypper in cron
```

2. Definiowanie zadań w pliku tekstowym */home/piotrek/Desktop/definicje-crona.txt* (musisz go stworzyć w jakimś linuksowym notatniku, np. *gedit*, *kate*, *nano*, itp.). Jako przykład wpisujemy definicję, która co minutę wysyła aktualny czas do *aktualny-czas.txt*:

```
* * * * * date >> /home/nazwaUzytkownika/Pulpit/aktualny-czas.txt
```

3. Wczytywanie definicji z pliku do *Crona*:

```
# crontab /home/nazwaUzytkownika/Pulpit/definicje-crona.txt
```

4. Uruchamianie / restartowanie *Crona*:

```
# service cron start
# service cron restart
# service cron status
```

Info: Jeśli usługa działa, to stwórz swój własny skrypt napisany w *bash*'u, który będzie wykonywał jakąś czynność o określonej porze lub w określonych odstępach czasu. Na temat tej usługi możesz dowiedzieć się więcej w osobnym dokumencie „*Cron*” (znajdziesz go na <http://edukacja.3bird.pl>).

Kompilacja jądra

Najpierw upewniamy się, że nasz Linux ma co najmniej 60GB miejsca na dysku, z tym że na partycję systemową powinno to być co najmniej 50GB:

```
# df -h
```

Jeśli nie ma, to powinno przeprowadzić się nową instalację *openSUSE* i zastosować indywidualny zaawansowany podział partycji, czyli:

/	50GB	ext4	(partycja systemowa)
/home	1GB	ext4	(foldery użytkowników)
swap	8GB	swap	(pamięć wirtualna = dwukrotność naszej pamięci RAM)

Jeśli partycje mają dobry podział, to upewniamy się, że jesteśmy na koncie *roota*:

```
# whoami
```

Zobaczmy, jak nazywa się nasze jądro (zazwyczaj będzie to „*vmlinuz-jakiśNumer*“):

```
# ls -al /boot
```

Instalujemy źródła jądra:

```
# zypper in kernel-source
```

Doinstalowujemy zależności:

```
# zypper in ncurses-devel
```

Sprawdzamy, czy mamy źródło jądra:

```
# ls -al /usr/src
```

Wchodzimy do źródeł jądra:

```
# cd /usr/src/linux
```

Wczytajmy obecną konfigurację:

```
# zcat /proc/config.gz > .config
```

Robimy kopię konfiguracji (na wszelki wypadek):

```
# cp .config /boot/config-2017-05-14
```

Uruchamiamy konfigurację:

```
# make menuconfig
```

Kompilujemy jądro:

```
# make && make modules_install
```

Archiwizujemy stare jądro:

```
# cp /boot/vmlinuz /boot/vmlinuz-stare-ale-jare
```

Kopiujemy nowe jądro do folderu boot:

```
# cp /usr/src/linux/arch/x86/boot/bzImage /boot/
```

Robimy dowiązanie symboliczne do nowego kernela:

```
# ln -sv /boot/bzImage /boot/vmlinuz
```

Uwaga: Jeśli pojawi się komunikat „*Failed to create a symbolic link: file exist*”, należy najpierw usunąć stary link (**# rm /boot/vmlinuz**), a potem spróbować jeszcze raz wydać powyższe polecenie.

Kopiujemy mapę systemu:

```
# cp /usr/src/linux/System.map /boot/
```

iptables

Najpierw zamykamy domyślnego firewalla:

```
# service SuSEfirewall2 stop
```

Sprawdzamy czy istnieje *iptables*:

```
# iptables -L
```

Przykładowo zezwalamy na otwarty port www (zawszad):

```
# iptables -A INPUT -p tcp -d 0/0 --dport www -m state --state NEW -j ACCEPT
```

```
# iptables -A INPUT -p tcp -d 0/0 --dport https -m state --state NEW -j ACCEPT
```

Na koniec odrzucamy wszystkie inne próby połączenia:

```
# iptables -A INPUT -m state --state NEW -j REJECT
```

Zapisujemy na stałe nasze ustawienia:

```
# iptables-save
```

Serwer Apache

```
# whoami
```

```
# zypper in apache2
```

Info: Poniżej, w pliku konfiguracyjnym, nic nie zmieniamy, ale warto wyświetlić jego zawartość, aby zobaczyć, jak jest zbudowany.

```
# cat /etc/apache2/httpd.conf
```

Info: Stwórzmy w notatniku przykładową stronę internetową:

```
# nano /srv/www/htdocs/index.html
```

```
<html>
```

```
<head></head>
```

```
<body>
```

```
Wasze imię i nazwisko...
```

```
</body>
```

```
</html>
```

```
# service apache2 start
```

Info: Wyłącz firewalla (instrukcja gdzieś powyżej). Testujemy działanie w przeglądarce internetowej (podajemy nasz numer IP, np. http://192.168.6.*).

Info: Jeśli wszystko jest w porządku, pobaw się pakietem **XAMPP** w systemie Windows.

Klient SSH

Windowsowym klientem może być program *Putty*. My jednak testujemy całość na linuksowym kliencie SSH. Utwórzmy go:

```
# whoami (dla pewności sprawdzamy, czy jesteśmy rootem)
```

```
# zypper in openssh
```

Uwaga: Istnieją dwa pliki konfiguracyjne, serwera i klienta:

```
# nano /etc/ssh/ssh_config (warto ten plik przejrzeć; to jest klient!)
```

```
# exit (wychodzimy z konta roota)
```

```
:~> whoami (upewniamy się, czy jesteśmy teraz na koncie zwykłego użytkownika)
```

```
:~> mkdir $HOME/.ssh
```

```
:~> ssh-keygen -t dsa
```

Uwaga: Program proponuje ścieżkę klucza, zazwyczaj „*\$HOME/.ssh/id_dsa*”. Akceptujemy klikając ENTER. Następnie podajemy „przepustkę” (*passphrase*), która jest rozbudowanym hasłem.

Sprawdzamy czy istnieją klucze:

```
:~> cd $HOME/.ssh
```

```
:~> ls -al
```

Uwaga:

Klucz prywatny: *\$HOME/.ssh/id_dsa* (nikomu nie udostępniamy i pozostawiamy go na naszym kliencie ssh)

Klucz publiczny: *\$HOME/.ssh/id_dsa.pub* (przenosimy na inny komputer-serwer za pomocą pendrive'a)

Uwaga: Wyłączamy firewall (instrukcja w tym dokumencie).

Serwer SSH

```
# whoami (upewniamy się, że jesteśmy rootem)
```

```
# zypper in openssh
```

```
# nano /etc/ssh/sshd_config (konfiguracja serwera)
```

```
PasswordAuthentication no
```

```
PubkeyAuthentication yes
```

```
# exit
```

```
:~> whoami (upewniamy się, czy jesteśmy teraz na koncie zwykłego użytkownika)
```

```
:~> mkdir $HOME/.ssh
```

Uwaga: Klucz publiczny utworzony na kliencie kopiujemy na pendrive'a, a następnie przenosimy do serwera na „*\$HOME/.ssh/*” i zmieniamy nazwę klucza na *authorized_keys* jak poniżej:

```
:~> cat /run/media/użytkownik/nazwaPena/id_dsa.pub >> $HOME/.ssh/authorized_keys
```

Uwaga: Bez skopiowania klucza publicznego, nie będzie możliwe łączenie się z serwerem. Klucz publiczny wpuszcza klienta, który go wygenerował i który posiada klucz prywatny. Klucz publiczny jest odpowiednikiem zamka w drzwiach, a klucz prywatny odpowiednikiem klucza do tego zamka.

```
:~> nano $HOME/.ssh/config (przełóżamy lub tworzymy plik konfiguracyjny; nadpisuje od główny globalny plik konfiguracyjny; krok nieobowiązkowy)
```

```
:~> su (ponownie logujemy się na roota)
```

```
# service sshd start (uruchamiamy serwer)
```

Uwaga: Wyłącz firewall na serwerze i na kliencie (instrukcja w tym dokumencie).

```
# ip address (sprawdzamy numer IP serwera)
```

Przenosimy się ponownie na klienta i testujemy połączenie z **klienta**:

```
:~> whoami (upewniamy się, że jesteśmy zwykłym użytkownikiem)
```

```
:~> ssh użytkownik@numerIP_serweraSSH
```

Serwer Postfix

Przed rozpoczęciem konfiguracji, zakłada się, że uczeń ma za sobą już konfigurację klienta Outlook i wie, co to są serwery SMTP i POP3 (jeśli nie wiesz... wróc do pierwszej klasy).

Na początek, sprawdzamy czy jesteśmy rootem:

```
# whoami  
# sudo su
```

Instalujemy Postfiksa:

```
# zypper in postfix
```

Sprawdzamy działanie SSMTP (rodzaj prostego serwera SMTP). Wysłamy list do siebie samego:

```
# mail -s tematListu użytkownik
```

treść listu, jakieś blablabla...

Ctrl+D

I odbieramy go:

```
# mail  
? 1 (wpisujemy numer e-maila, który chcemy wyświetlić)  
? q (wychodzimy)
```

W razie problemów (list nie dochodzi), sprawdzić logi:

```
# journalctl -f --b -u postfix.service  
# service postfix status
```

Sprawdzić także plik „hosts”:

```
# cat /etc/hosts (sprawdzić powiązanie naszego IP z naszą nazwą komputera)
```

```
192.168.17.23    nazwaHosta.nazwaDomeny  nazwaHosta
```

Możliwe, że nie trzeba będzie zmieniać domyślnych ustawień, ale dla pewności edytujemy plik konfiguracyjny:

```
# gedit /etc/postfix/main.cf
```

I mamy szczególne baczenie na następujące opcje:

```
queue_directory = /var/spool/postfix
```

```
mail_spool_directory = /var/mail
```

```
myhostname = nazwaNaszegoLinuksa.teb
```

```
mydomain = teb
```

```
mydestination = nazwaMojegoLinuksa, localhost, teb
```

```
mynetworks = 192.168.17.0/24, 127.0.0.0/8
```

Uwaga: Istnieje osobny samouczek przeznaczony dla *Gentoo* (niektóre polecenia i adresy plików są w nim inne), ale warto tam zajrzeć:

<http://edukacja/3bird.pl> → szukaj dokumentu „Postfix – informacje ogólne”.

Jeśli wszystko jest w porządku, uruchamiamy serwer:

```
# service postfix start
```

Instalujemy klienta *Claws-Mail* (lub *Sylpheed*), za pomocą którego przetestujemy Postfiksa (konfiguracja podobna do *Outlook*):

```
# zypper in claws-mail
```

i spróbujemy wysłać list do samych siebie:

Nasz adres: root

Serwer POP3: 192.168.17.23

Serwer SMTP: 192.168.17.23

Użytkownik: root

Hasło: (pozostawiamy na razie puste)

Port: 25

Jeśli działa, dodajemy do Postfixa autoryzację SASL:

```
# zypper in cyrus-sasl-saslauthd
```

VirtualBox wewnątrz openSUSE

Instalujemy *VirtualBox*:

```
# zypper dup
# zypper in virtualbox
# zypper in virtualbox-qt
# zypper in virtualbox-host-kmp-default
```

Dodajemy zwykłego użytkownika do grupy "vboxusers":

```
# usermod -a -G vboxusers nazwaUżytkownika
```

Pobieramy *ExtensionPack* ze strony:

<https://www.virtualbox.org/wiki/Downloads>

Uruchamiamy program:

```
# VirtualBox %U (program jest także dostępny w menu Applications / System / Virtual Machine)
```

Wczytujemy *ExtensionPack* w menu *VirtualBox*:

File / Preferences / Extensions / Add: Install

Następnie należy przelogować się na graficzne konto zwykłego użytkownika i uruchomić *Virtual-Box*.

Obsługa modemu

Modem USB: **Huawei E3276**, operator *Plus GSM*.

Uwagi wstępne: Rozwiązanie **nie** bazuje na pakiecie **wvdial** (został on wycofany z dystrybucji *openSUSE*). Konfiguracja odbywa się poprzez **NetworkManager** (ikonka znajdująca się obok zegara, ta sama, która odpowiada za połączenia *Wi-Fi*).

```
# lsusb (czy nasz modem został wykryty i jaki ma fizyczny identyfikator)
```

```
...
```

```
Bus 001 Device 004: ID 12d1:1506 Huawei Technologies Co., Ltd. Modem/Networkcard
```

```
...
```

```
# zypper in usb_modeswitch
```

```
# ls /dev/ttyUSB* (czy zostało utworzone logiczne urządzenie dla modemu, np. ttyUSB0)
```

Jeśli wszystko jest w porządku:

PPM → **NetworkManager** → *Configure Network Connections* → + *Add new connection: Mobile Broadband (Komórkowe szerokopasmowe)* → *Huawei E3276* → *Poland* → *Plus* → *Select your plan: My plan is not listed...* → *APN: internet* → zakładka *Mobile Broadband (gsm)*:

Number: *99#

Username: puste

Password: puste

APN: internet

PIN: brak

zakładka **PPP**: zostawiamy wszystko domyślnie (na pewno musi być **CHAP**);

zakładka **IPv4 / IPv6**: zostawiamy domyślnie (*Method: Automatic*).

Wstępnym dowodem, że modem zaczął działać, jest zielona dioda (światło ciągłe).

Problemy i tipsy

1. Nie działają polecenia:

```
# kwite /etc/jakisPlik.conf
```

a także:

```
# gedit /etc/jakisPlik.conf
```

Rozwiązanie problemu: zaloguj się na konto *root* w wersji graficznej (nie przez tryb tekstowy).

2. Brak polecenia *ifconfig*? Należy zainstalować:

```
# zypper in net-tools-deprecated
```

Info: Polecenie „*ifconfig*” uważane jest za przestarzałe. Zamiast niego zaleca się używanie polecenia „*ip*”, np.:

```
# ip address
```

```
# ip link (nazwy wszystkich interfejsów sieciowych)
```

3. Podczas próby aktualizacji systemu pojawia się komunikat o braku repozytorium. Należy:

```
# zypper addrepo https://download.openSUSE.org/repositories/(dalszaCzęśćAdresu)...
```

4. Po zmianie „*theme*” okien / ikon / kursorów... system *KDE Plasma* i jego elementy nie reagują na kliknięcia myszki. Należy przełączyć się na sesję *KDE Plasma Wayland*, wybrać odpowiednie schematy okien / ikon, a następnie z powrotem przełączyć się na sesję *KDE Plasma Wayland*.

5. Aby zlikwidować automatyczną blokadę ekranu po 5 minutach, należy (w **Gnome**):

Application Launcher / Applications / Settings / Configure Desktop / Screen Locking (Blokada ekranu)...

lub (w **KDE**):

Application Launcher / Applications / Settings / System settings / Desktop behavior (Zachowanie przestrzeni roboczej) / Screen Locking...

6. W przypadku blokady ekranu, należy wydać polecenie jako *root* na pierwszym terminalu (*Ctrl+Alt+F1*):

```
# loginctl list-sessions (wykaz sesji)
```

```
# loginctl unlock-session 1 (lub inny numer sesji)
```

lub

```
# loginctl unlock-sessions
```

Następnie należy wrócić na terminal nr 7 (*Alt+F7*).

Inne rozwiązanie:

```
# kill -9 light-locker
```

Uwaga: Jeśli *openSUSE* jest na *VirtualBox*, a systemem macierzystym jest także *Linux* - nie będzie działać kombinacja klawiszy *Ctrl+Alt+F1*.

7. Brak obramowań okien / dekoracji w *KDE*.

Należy doinstalować następujące pakiety:

```
# zypper in kde-plasma
```

```
# zypper in kdebase4-runtime
```

```
# zypper in kdebase-workspace
```

```
# zypper in kdelibs4
```

```
# zypper in libkdecoration2-devel
```

```
# zypper in libkdecoration2-5
```

Następnie wybieramy dany styl dekoracji:

Application Launcher / Applications / Settings / System settings / Application Style / Window Decorations

8. Rozdzielczość ekranu w *KDE*.

START / Programy / Ustawienia / Ustawienia systemowe / Sprzęt: Wyświetlanie i monitor

9. Edycja *GRUB2* w *KDE*.

START / Programy / System / Ustawienia administratora YaST / System / Program rozruchowy / zakładka: Opcje menedżera rozruchu / Domyślna sekcja rozruchu

10. Ciągłe zapytania *KDEwallet* o hasło.

Ustawienia systemu / Szczegóły konta / [] Włącz podsystem portfela KDE

11. Podczas aktualizacji *VirtualBox Guest Additions* pojawia się komunikat: „*Kernel Headers Not Found For Target Kernel Error*”.

zypper in kernel-devel

12. System nie wykrywa karty dźwiękowej pomimo dobrej konfiguracji i uruchomionego serwera *pulseaudio*.

Rozwiązanie: Przetestuj na koncie zwykłego użytkownika (musi być dodany do grup „*audio*”, „*pulse*”). Root ze względu na bezpieczeństwo nie jest dodawany do tych grup.

13. Przeglądarka *Firefox* uruchamia video na *YouTube*, ale już nie na *Facebooku*.

Rozwiązanie:

zypper in ffmpeg libavcodec libavformat libavresample libx264 x264 libx265

zypper in libavfilter libswscale libavutil lame x265 x264-libs

zypper rm MozillaFirefox

zypper in MozillaFirefox

Jeśli to nie pomoże, należy zainstalować pakiety z repozytorium *packman*:

zypper in opi

opi codecs (w szczególności chodzi o *vlc-codecs*)

Info: Po kolejnych aktualizacjach systemu, być może trzeba będzie całą procedurę powtórzyć.

14. Aby zainstalować sterowniki drukarki *HP LaserJet 1018* potrzebny jest pakiet **foo2zjs.tar.gz**, ale strona domowa tego projektu już nie istnieje.

Rozwiązanie: Poszukaj pakietu na stronie <http://software.opensuse.org>.

Uwaga: Urządzenie drukarki powinno być obecne jako */dev/usb/lp0*. Jeśli nie ma takiego urządzenia po wpięciu drukarki, sprawdzić czy istnieje moduł *usb-lp*:

modinfo usb-lp

15. Po odpięciu zasilacza, porty USB otrzymują niższe napięcie i nie są w stanie zasilić modemu wpiętego do USB. Zmiana ustawień nie pomaga.

Rozwiązanie: W systemie uruchomione są aż trzy nadpisujące się usługi do zarządzania energią. Dwie z nich należy usunąć i zablokować:

systemctl status tlp

zypper rm tlp

zypper al tlp

systemctl status power-profiles-daemon

zypper rm power-profiles-daemon

zypper al power-profiles-daemon

Usługa, którą zachowujemy, to **upower** (wchodzi w skład ustawień KDE).

16. Automatyczne ładowanie modułów podczas startu (na przykładzie modułu *nouveau*):

nano /etc/default/grub

GRUB_CMDLINE_LINUX="nomodeset"

Jeśli to nie pomoże, to:

cat /etc/modules-load.d/nouveau.conf

nouveau

nano /etc/X11/xorg.conf.d/50-device.conf

Section "Device"

Identifier "Stara Nvidia"

Driver "nouveau"

EndSection

Należy także odinstalować w tym wypadku wszystkie pakiety mające w nazwie „*nvidia*”, a jednocześnie doinstalować pakiety mające w nazwie „*Mesa*”, „*libdrm*”, „*dri*”. Należy także w GRUB uaktywnić opcję *acpi=on*. W niektórych przypadkach dobrze jest odinstalować „*drm-kmp-default*”.

Jeśli sterownik „*nouveau*” nie działa poprawnie, należy go odinstalować i wypróbować „*modesetting*” jako alternatywę.

Ostatnia aktualizacja: 5 stycznia 2024.