

# Spis treści:

## 1. Linux, to brzmi dumnie!

- 1.1 Dlaczego powstała ta publikacja?
- 1.2 Czym jest Linux
- 1.3 Historia systemu
- 1.4 Dlaczego warto go poznać?
- 1.5 Czym jest dystrybucja?
- 1.6 Przegląd najbardziej znanych dystrybucji
- 1.7 Która na początek?

## 2. Instalacja systemu

- 2.1 Czy wystarczy?
- 2.2 Jak zdobyć Linuksa?
- 2.3 Przygotowania do instalacji

## 3. Pierwszy start

- 3.1 Lilo - bootloader Linuksa
- 3.2 Start systemu

## 4. Ogółem, czyli wszystko po trochu

- 4.1 Drzewo katalogów
- 4.2 Hasła w systemie
- 4.3 Root - superużytkownik
- 4.4 UID i GID
- 4.5 Brak dostępu, czyli prawa do plików

## 5. Graficzne środowisko pracy

- 5.1 Czym jest graficzne środowisko pracy?
- 5.2 Jakie udogodnienia płyną z jego korzystania?
- 5.3 Przegląd dostępnych środowisk

## 6. U podstaw siła drzemie, czyli obsługa konsoli

- 6.1 Rzeczywista natura systemu
- 6.2 Początki poruszania się w konsoli
- 6.3 Midnight Commander - łatwej być nie może
- 6.4 Polecenia

## 7. Pakiety i źródła - instalacja oprogramowania

- 7.1 Czym są źródła?
- 7.2 Czym są pakiety?
- 7.3 Instalacja programów ze źródeł i pakietów

## 8. Urządzenia

- 8.1 Jak reprezentowane są urządzenia w Linuksie?
- 8.2 Winsprzet - kłopot?
- 8.3 Instalacja urządzeń

## 9. Jądro Linuksa

- 9.1 Tryby pracy

- 9.2 [Operowanie pamięcią w Linuksie](#)
- 9.3 [Funkcjonowanie sieci](#)
- 9.4 [Budowa jądra](#)

## 10. [Linki](#)

- 10.1 [Gdzie szukać pomocy?](#)
- 10.2 [Strony, które warto odwiedzić](#)
- 10.3 [Grupy dyskusyjne](#)

## 11. [Oprogramowanie](#)

## 12. [Dodatek A](#)

- 12.1 [Linus Torvalds - autor Linuksa](#)

## 13. [Dodatek B](#)

- 13.1 [Pingwinek w logo - skąd?](#)

## 14. [Standardowa adnotacja](#)

- 14.1 [Zgłaszanie błędów](#)
- 14.2 [Informacje odnośnie wersji dokumentu](#)
- 14.3 [Podziękowania](#)
- 14.4 [Zmiany](#)

# Linux, to brzmi dumnie!

W rozdziale tym chcę Cię wprowadzić w świat Linuksa. Dowiesz się, czym tak naprawdę jest i dlaczego warto go poznać. Przybliżę Ci historię i genezę tworzenia systemu. Po koniec rozdziału znajdziesz informacje czym jest dystrybucja i którą wybrać, aby łatwo przyswoić sobie nowe środowisko pracy.

## Dlaczego powstała ta publikacja?

Linux zdobywa coraz to większą popularność. Wiele firm wybiera go, jako system działający bez zastrzeżeń. Także wiele domowych użytkowników decyduje się na zmianę systemu, chcąc poznać coś nowego, powiększyć swą wiedzę informatyczną.

Dlatego też napisałem tą krótką publikację, mającą na celu choć w małym stopniu pomóc Ci stawiać pierwsze kroki w Linuksie. Jak się przekonasz nie jest to system, który zrobi za Ciebie to co będziesz chciał. Wiele rzeczy wykonasz samodzielnie, co z biegiem czasu zacznie przynosić Ci ogromną satysfakcję.

Publikacja ta powstała także między innymi, dlatego, że choć można znaleźć w Internecie wszystkie zawarte tu informacje, to zazwyczaj nie są one w jednym miejscu, co wydłuża czas ich wyszukiwania.

## Czym jest Linux

Linux to stabilne i bezpieczne jądro systemu operacyjnego napisane przez fińskiego studenta Linusa Torvaldsa. Co mam na myśli mówiąc jądro?

Wszystko: pliki, katalogi, które otwieramy, uruchamiane **programy** obsługiwane są przez jądro systemu. To one odpowiedzialne jest za przydzielenie odpowiedniej ilości pamięci, a późniejsze jej zwolnienie. Oczywiście to nie wszystkie zadania jądra - wykonuje one bardzo skomplikowane operacje, takie jak na przykład wysyłanie i odbieranie komunikatów od sprzętu.

Tak więc Linux to wyłącznie jądro (jednak w dalszej części publikacji będę miał na myśli cały system operacyjny). Ale praca z samym kernelem (z ang. jądro) nie przyniosłaby żadnego skutku i pożytku, gdyby nie oprogramowanie. To właśnie dostarcza nam dystrybucja, ale o tym jaśniej powiem pod koniec rozdziału.

Linux jest darmowym i co najważniejsze wolnym systemem. Tak, tak! W każdej chwili możesz pobrać jego dowolną odmianę z Internetu. Płyty z nagrany na nie Linuksem możesz także sprzedawać! Pozwala na to licencja GNU GPL (General Public Licence) na jakiej jest oparty.

Wy tłumaczeniu podlega jeszcze słowo "wolny". Nie chodzi tu bynajmniej o szybkość (a jak się przekonasz, Linux to naprawdę szybki system). Wolny oznacza, że dołączony jest jego kod źródłowy, który bez ograniczeń możemy zmieniać, poprawiać wedle naszego uznania, nie łamiąc tym samym prawa! To właśnie zasługa GPL, w której od samego początku chodziło o rozprzestrzenianie aplikacji wraz z kodem źródłowym.

Aby bardziej zrozumieć istotę systemu, należy poznać jego historię. I choć jest ona długa, to warto ją poznać od samego początku!

## **Historia systemu**

Linux powstał w roku 1991. Jednak historia systemu sięga nieco wcześniej. Linus Torvalds - autor Linuksa już dużo wcześniej przyglądał się i porównywał dwa systemy: Uniksa i Miniksa. Dzięki kontaktom w gronie użytkowników tego drugiego, Linus znajdował odpowiedzi na wszystkie jego wątpliwości.

Torvalds zaczął pracę w Asemblerze, jednak później zdecydował pisać jądro w C. Linus modyfikował kod Miniksa opierając się na Uniksie.

W sierpniu 1991 roku światło ujrzała wersja 0.01. Niestety daleko jeszcze było Linuksowi do doskonałości. Jądro zawierało tylko proste sterowniki. Brakowało mu m.in. obsługi dyskietek. Linus jednak się nie poddawał i w niecałe 2 miesiące stworzył wersję 0.02. Jajko potrafiło uruchamiać już proste **programy**.

Autor systemu postanowił rozgłosić o nim informacje. Na grupie comp.os.minix pojawiły się pierwsze listy dotyczące Linuksa. Przynęta zadziałała - nowe jądro ściągnęło 10 osób, z czego pięć dokonało własnych poprawek..

Linux zaczął się błyskawicznie rozwijać. Zaraz po tym jak wyszła wersja 0.03, pojawiła się 0.10, a w marcu 1992 roku - 0.95.

Pod koniec roku 1991 z systemu korzystało 100 osób, a samo źródło liczyło 10 tys. linii kodu! W I kwartale 1992 roku liczba użytkowników powiększyła się dziesięciokrotnie, a objętość kodu - ponad trzykrotnie.

W grudniu roku 1993 pojawiła się wersja 0.99pl14 licząca 100 tys. linii i 20 tys. użytkowników. Na drodze postępu systemu już nikt nie mógł stanąć!

W 1994 roku powstało jajo 1.0. W tym samym czasie pojawiły się pierwsze firmy zajmujące się dystrybucją systemu - Red-Hat i Caldera.

Ale skąd nazwa Linux? Torvalds proponował nazwę FREAX. Jak twierdził zawierała w sobie słowo free (wolny), freak (dziwak) i X jak Unix.

Inne zdanie miał Ari Lemmke, który umieścił system na swoim FTP. Według niego nie był to najlepszy pomysł. Zaproponował więc nazwę LINUX. Nazwa została, a zdaniem Linusa dużo lepiej brzmi, niż wcześniejsze FREAX.

Obecnie Linux ma użytkowników na całym świecie! Wiele firm używa Linuksa i to nie dlatego, że jest darmowy, ale dlatego, że jest wydajnym, wielozadaniowym systemem i co najważniejsze - jest bezpieczny.

### **Dlaczego warto go poznać?**

Jest mnóstwo argumentów będących uzasadnieniem, dlaczego naprawdę warto. Jednym z nich jest fakt, że Linux bardzo szybko się rozwija i coraz więcej firm z niego korzysta. W przyszłości będzie więc dużo łatwiej pozyskać pracę znając więcej, niż jeden system operacyjny. Warto także chociażby po to aby kształtować i doskonalić swoją wiedzę informatyczną. W Linuksie zrozumiesz na jakiej podstawie funkcjonuje sprzęt i jak ogólnie działa każdy, nie tylko Linux, system operacyjny!

Linux jest wieloplatformowy - możesz korzystać z niego na standardowych PC-tach, Macintoshach, Sparcach, maszynach Alpha, a nawet Amigach.

Poszerzysz także swoje informacje na temat Internetu, a to dlatego, że wiele serwerów obsługiwanych jest przez ten system.

Innym powodem, dla którego warto, jest satysfakcja, jaką przynoszą sukcesy w nowym systemie!

Jednak sam musisz odpowiedzieć sobie na to pytanie; nie posiadając odpowiedniej wiedzy informatycznej, zapału i czasu możesz szybko zniechęcić się i co najgorsze możesz już nigdy nie sięgnąć po ten znakomity system!

### **Czym jest dystrybucja?**

Jak już wcześniej pisałem, Linux to jądro systemu. Jednak abyśmy mogli cieszyć się z efektów pracy, niezbędne będzie jakieś oprogramowanie i te właśnie dostarcza nam dystrybucja. Dodatkowo zawiera ona instalator systemu, dzięki czemu Linux znajdzie się na naszych komputerach.

Firmy i grupy programistów wręcz prześcigają się w swojej pracy, chcąc stworzyć oryginalną i wartą [instalacji](#) pozycję. Dzięki temu możemy przebierać dziś pośród różnych odmian tego samego systemu operacyjnego, wybierając dystrybucję najbardziej nam odpowiadającą.

Pozycje różnią się przede wszystkim poziomem trudności obsługi i ilością dostępnego oprogramowania dołączonego bezpośrednio na płytach CD. Różnica może także polegać na domyślnym poziomie zabezpieczeń, choć pamiętajmy, że to wciąż ten sam system, więc w każdej chwili możemy to zmienić.

Reasumując: dystrybucja to jądro z zestawem oprogramowania wzbogacone o instalator systemu.

## Przegląd najbardziej znanych dystrybucji

Obecnie w świecie Linuksa dostępnych jest ogromna ilość dystrybucji. Istnieją także mini dystrybucje - Linux na jednej bądź kilku dyskietkach!

W podrozdziale tym opiszę jednak pięć najbardziej znanych odmian Linuksa, które bez wątplenia zasługują na szczególną uwagę.

### *Mandrake Linux*

Dystrybucja jest tworzona przez firmę o tej samej nazwie. Mandrake powstał na podstawie bardzo znanego Red Hata (czytaj niżej). Charakterystyczną cechą dystrybucji jest niebywała prostota jej obsługi. Programiści z Mandrake tworząc ją ogromny nacisk położyli na udogodnienia płynące z prostoty obsługi. W tym celu napisano wiele programów wykonujących wiele czynności za użytkownika. Także pliki konfiguracyjne zostały zmienione tak, aby domyślne ustawienia systemu były jak najlepsze dla Ciebie.

Mandrake możemy pobrać z oficjalnych serwerów firmy: <ftp.mandrake.com> Informacje odnośnie instalacji znajdziesz także na stronie Mandrake: [www.mandrakesoft.com](http://www.mandrakesoft.com)

Dystrybucja posiada także wersję płatną, do której dołączone są dodatkowe programy komercyjne.

### *Red Hat*

Red Hat tworzony jest przez amerykańską firmę Red Hat. Dystrybucja ta jest dziś najpopularniejszą odmianą Linuksa. Ceniona przez wielu zdobyła wysokie miejsce na podium rynku systemu. Poziom trudności oferowany przez system jest średni, dlatego też Red Hat kierowany jest do użytkowników, którzy mieli już styczność z systemem, ale i nowicjusze nie powinni mieć ogromnych problemów z jej opanowaniem. I choć dystrybucja często wybierana jest na serwery sieciowe, świetnie sprawuje się do użytku domowego.

Red Hat, podobnie jak Mandrake, posiada wersję komercyjną i tak jak w przypadku swojego "kolegi" - zawiera ona dodatkowe oprogramowanie napisane przez firmę.

Adres Red Hata: [www.redhat.com](http://www.redhat.com) i <ftp.redhat.com>

### *PLD*

PLD to nasza dystrybucja. Tak, tak Polacy także tworzą Linuksa. PLD bazuje na Red Hacie. Rodacy jednak ogromny nacisk zrobili na bezpieczeństwo i stabilność systemu. Dużo czasu poświęcono także na implementację protokołu IP w wersji 6 znanej także pod nazwą next generation - następna generacja. Nasza dystrybucja to także największy Red Hat.

PLD cieszy się uznaniem wśród administratorów serwerów. Niektórzy mówią też, że choć bazuje na Red Hacie, to wyszedł po prostu od niego lepszy! Dystrybucja kierowana jest do zaawansowanych użytkowników.

O projekcie PLD poczytamy pod [www.pld-linux.org](http://www.pld-linux.org)

### *SuSE*

SuSE to dzieło naszych sąsiadów - Niemców. Prostota obsługi i wsparcie techniczne czyni go silną pozycją. To, za co jednak SuSE posiada duże grono zwolenników, to instalator i konfigurator Yast charakteryzujący się łatwością obsługi oraz funkcjonalnością.

Oczywiście po szczegóły możemy sięgnąć do strony oficjalnej, skąd także pobierzemy SuSE lub kupimy jego rozbudowane wydanie: [www.suse.com](http://www.suse.com)

### *Knoppix*

Knoppix jest butowalnym Linuksem. Oznacza to, że system uruchamia się z płyty CD, po uprzednim butowaniu z niej. Jest ciekawym rozwiązaniem, kiedy chcemy najpierw "zobaczyć

na żywo" jak wygląda Linux. Knoppix bazuje na Debianie (czytaj niżej). Dystrybucja rozprowadzana jest na jednej płycie. Najnowsza wersja 3.2 zawiera uaktualnione pakiety. Knoppix, pomimo faktu, że jest butowalny, można go zainstalować na dysku twardym.

### *Debian*

Tym czym Debian różni się znacząco od innych dystrybucji, jest fakt, że nie stoi za nim żadna firma - tworzy go grono programistów z całego świata, w tym także Polacy! Ciesząca może być informacja, że Debian to największa dystrybucja - posiada prawie 9000 pakietów umieszczonych na 7CD + 1 dodatkowej (sytuacja w wersji stabilnej 3.0). Debian idealnie spisuje się na serwerach, a to za sprawą jego dużej stabilności., będącej wynikiem długotrwałego i szczegółowego sprawdzania każdego pakietu. Kolejnym plusem dystrybucji jest jej dostępność na wiele platform, dzięki czemu Debiana możemy uruchomić nawet na Amigach!

Dodatkowo dystrybucja znakomicie nadaje się do użytku domowego. I choć przeznaczona jest dla systemowych bywalców, docenią ją nawet osoby raczkujące w Linuksie - 8710 pakietów sprawi, że do codziennej pracy wykorzystamy oprogramowanie z CD, bez konieczności pobierania pakietów z sieci.

Debian znajduje się pod: [www.debian.org](http://www.debian.org)

### *Slackware*

Slackware ceniony jest przede wszystkim przez administratorów serwerów i dla tych osób jest kierowany. Firma tworząca dystrybucję położyła naciska na ochronę systemu. Dzięki temu Slackware to potężne narzędzie w rękach administratorów. Mankamentem wydaje się jednak mała ilość programów dołączonych z CD. Slackware wydawany jest na jednej płycie, ponadto można pobrać dodatkowy CD (sytuacja z wersji stabilnej 8.1). Wiele osób jednak to ceni - w końcu Slackware to dystrybucja na serwer, a nie system domowy!

Adres dystrybucji: [www.slackware.com](http://www.slackware.com)

### **Która na początek?**

Idealnym rozwiązaniem na początek wydaje się Mandrake, który bez wątpienia pomoże Ci stawiać pierwsze kroki w Linuksie. Alternatywnym rozwiązaniem jest SuSE i jego Yast. Aby jednak korzystać z SuSE będziesz musiał wydać trochę pieniędzy...

Moja sugestia jest taka: pobierz, bądź kup na aukcji lub w sklepie internetowym najnowszą wersję Mandrake. Bez wątpienia będzie to najmniej ryzykowny start pracy z nowym systemem!

## **Instalacja systemu**

Po zaczerpnięciu powyższych informacji za pewne chcesz teraz zainstalować po raz pierwszy Linuksa. Rozdział ten ma na celu ułatwienie Ci tego zadania. I choć nie znajdziesz tu informacji na temat przebiegu instalacji danej dystrybucji (napisanie instrukcji instalacji do poszczególnych dystrybucji zajęłoby wiele czasu i miejsca) zawarte tu wskazówki na pewno pomogą Ci w instalacji systemu.

### **Czy wystarczy?**

Aby korzystać z Linuksa nie trzeba naprawdę szybkiego sprzętu. Większość dystrybucji podczas kompilacji optymalizowana jest dla procesorów klasy i386, co oznacza, że system

będzie działał nawet na tak starym komputerze, jak 50Mhz i 4MB! Oczywiście, aby w pełni wykorzystać Linuksa, będziemy potrzebowali szybszy komputer. W zupełności wystarczy nam 366Mhz, 128MB pamięci ram i 4GB wolnego miejsca na dysku. Z taką konfiguracją bezproblemowo będziemy słycać muzyki, oglądać filmy w różnych formatach i korzystać z Internetu.

## **Jak zdobyć Linuksa?**

Jeśli dysponujesz szybkim stałym łączem, możesz łatwo pobrać obrazy wybranej dystrybucji z oficjalnych stron lub polskich serwerów lustrzanych, a później wypalić je na CD. Jeżeli jednak nie masz dostępu do Internetu bądź Twój modem jest zbyt wolny aby ściągnąć pliki, możesz łatwo i szybko nabyć system na aukcjach i w sklepach internetowych. Bardzo interesującą ofertę posiada [linux.cd.home.pl](http://linux.cd.home.pl). Przy zakupie jednego CD płacimy za niego 5 zł, przy czym im więcej płyt ma dystrybucja, tym cena jest niższa. W ten sposób za Debiana na 7 CD zapłacimy tylko 17 zł! Polecam ten sklep - tanio, szybko, a dystrybucje wypalane są na porządnych płytkach.

Powinieneś się także rozejrzeć. Może ktoś z Twoich znajomych korzysta z Linuksa i pożyczy Ci płyt z nim.

## **Przygotowania do instalacji**

Gdy mamy już płyty z Linuksem, przed pierwszą instalacją należy wykonać kilka czynności, które ustrzegą nas przed ewentualnymi problemami. Na początku dobrze jest zapoznać się przebiegiem instalacji, dzięki czemu będziemy świadomi, co może sprawić nam trudność.

Sprzęt, sprzęt i jeszcze raz sprzęt. Podczas instalacji bardzo przyda się znajomość własnego sprzętu. Chodzi tu nie tylko o jego nazwę ale np. informację o chipsecie. Duże ułatwienie przyniesie nam Mandrake, którego instalator praktycznie zawsze poprawnie określa urządzenia i ich parametry.

Za pewne wiesz, że dla Linuksa będziesz musiał wyznaczyć miejsce na dysku. Partyjce, jakich potrzebuje Linux, to "Linux Native" - partycja najczęściej w systemie plików ext2 lub ext3 i partycja wymiany - "Linux Swap". Co prawda Mandrakea można zainstalować na partycji fat32 ale znacząco obniży to wydajność systemu.

Partyjce możemy przygotować zarówno podczas instalacji jak i przed nią, np. znanym programem Partition Magic. Jeśli go nie posiadamy, możemy skorzystać z partycjonowania dysku w Mandrake. Program ten jest prosty w obsłudze i wygląda bardzo podobnie do wcześniej wymienionego Partition Magika. Przyrządzenie partycji przed instalacją posiada wiele udogodnień, między innymi instalator później sam je wykryje, co zaoszczędzi nam wiele pracy.

Przed pierwszą instalacją warto również zrobić kopię zapasową ważnych danych, dzięki którym jeśli coś pójdzie źle po Twojej myśli, łatwo przywrócisz komputer do poprzedniego stanu.

## **Pierwszy start**

Linux zawitał już w Twoim komputerze. Ale co teraz? W rozdziale tym znajdziesz informacje odnośnie samego startu Linuksa - co się uruchamia, i co dzieje się z tym dalej.

## **Lilo - bootloader Linuksa**

Podczas pierwszego uruchomienia Linuksa Twój wzrok na pewno przyciągnął ekran wyboru systemu operacyjnego. To właśnie był LILO - LInux LOader. Lilo to program, który instaluje się (domyślnie) w sektorze butującym dysku głównego. Dzięki niemu możemy używać kilku systemów, wybierając przy starcie komputera ten, który będzie nam potrzebny. Program posiada wiele funkcji, a w jego pliku konfiguracyjnym możemy podać parametry startowe systemu.

Innym znanym bootloaderem dla Linuksa jest Grub. Ostatnio zyskał dużo popularność przez oferowanie opcji, których Lilo nie posiada. Większość osób używa jednak "naszego" LILO, twierdząc, że to sprawdzony bootloader, który z niejednym się już spotkał, poza tym LILO to legenda - towarzyszy Linuksowi od jego początku.

## Start systemu

Gdy Twój Linux uruchamia się widzisz na ekranie mnóstwo komunikatów. Są to informacje wyświetlane przez uruchamiane demony. Demon, najprościej tłumacząc, to program sprawujący jakąś kontrolę. I tak podczas startu zaobserwować możemy np. uruchamianie Apache - serwera http.

Na samym początku startu Linux sprawdza nasze urządzenia i ładuje odpowiednie dla nich moduły - części jądra odpowiedzialne za daną funkcję, np. obsługę karty dźwiękowej.

Musisz wiedzieć, że start, zamknięcie, czy uruchomienie ponowne komputera w Linuksie kontroluje demon initd. Posiada on tak zwane run levele, co w wolnym tłumaczeniu oznacza poziomy uruchamiania. Określają one, co następuje w danym trybie - uruchomienie danego programu, czy jego zatrzymanie. Initd posiada 7 poziomów:

0. zamknięcie systemu
1. tryb single-user
2. tryb multi-user bez sieci
3. tryb multi-user z siecią
4. nie używany
5. to samo, co 3, ale z graficznym ekranem logowania
6. restart systemu

Działanie każdego uruchamianego programu nazywany jest procesem a jądro przydziela mu specjalny identyfikator - PID (z ang. process identifiator). Dodatkowo proces zawiera informacje, którego użytkownika jest własnością, czyli kto go uruchomił.

Jednak w systemie musi jeszcze istnieć proces macierzysty, nadzorujący pracę nad innymi procesami. Jest nim właśnie initd. Zawsze kiedy dany proces zostanie zabity, czyli zakończony, następuje odwołanie do inita. Także, kiedy wystąpi błąd w samym inicie, nastąpi restart systemu. To tak, jakby jądro systemu przestało odpowiadać - init nie może kontrolować procesów, więc wszystkie kończy, co daje restart systemu.

Init ma zawsze PID równy 1. Informuje nas to o tym, że jest pierwszym procesem uruchamianym przez jądro systemu. Wyraźnie widać to poniżej:

```
PID TTY          TIME CMD
   1 ?            00:00:04 init
   2 ?            00:00:00 keventd
```

Jest to wynik działania ps -e (czytaj rozdział [Polecenia](#)).

## Ogółem, czyli wszystko po trochu

Informacje na temat obsługi Linuksa można by było pisać bez końca. System ten posiada wiele mechanizmów, których opisanie zajęłoby wiele czasu. Dlatego też wprowadzając Cię w świat Linuksa staram się przekazać Ci wiedzę w sposób ogólnikowy, pomijając sprawy szczegółowe, które początkującej osobie mogłyby wydawać się trudne, co komplikowałoby z pewnością pojęcie podstaw.

Dział ten ma na celu wprowadzenie Cię do obsługi systemu. Konieczne do tego będzie poznanie kilku haseł, które jak sam się później przekonasz, towarzyszyć Ci będą przez całą dalszą pracę z systemem.

### Drzewo katalogów

Każdy system posiada swoje drzewo katalogów, będące hierarchią plików. I też nie inaczej jest z Linuksem. System ten posiada hierarchie zgodną z rodziną Uniksów, dzięki czemu systemy takie jak Linux, czy rodzina BSD mają tak samo wyglądającą strukturę katalogów.

Dyski w Linuksie nazywane są "po imieniu", tak więc nie znajdziemy tu dysku o nazwie 'C', czy 'D'. Dysk pierwszy (primary) posiada etykietę 'hda', a kolejno występujące po nim: 'hdb', 'hdc', itd.

Zasada ta dotyczy również partycji. I tak: główna partycja na dysku pierwszym będzie nosiła nazwę 'hda1', a na dysku drugim - 'hdb1'. Jak więc widzisz zasada jest prosta, a czytając 'hda1', możesz z pewnością powiedzieć dużo więcej niż w przypadku nazwy dysku 'C'.

Katalog główny oznaczany jest znakiem slash, czyli '/'. Tu mieszczą się wszystkie katalogi i pliki. A więc '/' reprezentuje całą przestrzeń dyskową, na której zainstalowany jest Linux. Poniżej zamieściłem wszystkie katalogi podgłówne wraz z ich opisami, co zawierają:

Katalog:	Zawira:
/bin	Pliki binarne; narzędzia systemowe
/dev	Pliki obsługujące urządzenia: cdrom, modem
/boot	Informacje odnośnie bootowania systemu
/etc	Globalne pliki konfiguracyjne
/home	Katalogi użytkowników (oprócz katalogu roota)
/lib	Skompilowane biblioteki potrzebne do obsługi systemu
/proc	Informacje o procesach
/root	Katalog superużytkownika (czytaj kolejne podrozdziały)
/sbin	Pliki binarne; narzędzia do użytku
/tmp	Pliki tymczasowe, w tym pliki internetowe
/usr	Narzędzia i aplikacje
/var	Pliki pocztowe, kolejki drukarki i logi systemowe

### Hasła w systemie

Linux, jak już wcześniej wspomniałem, to bezpieczny system operacyjny. Tutaj każdy nowo dodany użytkownik posiada własne hasło. Wyrażenia szyfrowane są jednostronnym

algorytmem MD5, więc są one bardzo trudne do złamania. Oczywiście to, czy dane hasło zostanie odszyfrowane, czy też nie, zależy od poziomu jego skomplikowania. Powinno się więc stosować duże i małe litery, a ponadto cyfry lub inne znaki. Przykładem takiego hasła może być:

```
dY71-GZ1.
```

Linux zazwyczaj nie pozwala na stosowanie haseł o mniejszej ilości znaków niż 5, choć można to oczywiście zmienić. Ponadto próba podania hasła takiego jak użytkownika w dystrybucjach o dużym bezpieczeństwie kończy się komunikatem mówiącym, że hasło jest zbyt proste!

Hasła w systemie zapisane są w `/etc/shadow`.

## Root - superużytkownik

Wyróżniającym się spośród wszystkich użytkowników w Linuksie jest root, czyli superużytkownik. Root posiada specjalne przywileje do każdego katalogu, pliku i konfiguracji. Może zaglądać do każdego zakamarka systemu bez żadnych ograniczeń. Pełni on rolę administratora całego systemu. Dlatego też roota używa się tylko do celów administracyjnych, wyłączając go z normalnej pracy.

Ważnym elementem jest hasło superużytkownika. Powinno być w sposób szczególny przemyślane i co najważniejsze trudne do odgadnięcia. Przejęcie administratora przez obcą osobę może przynieść opłakane skutki, w tym usunięcie całego systemu!

## UID i GID

Każdy użytkownik w systemie posiada własny identyfikator - jest nim UID (user identifier). Drugim numerem identyfikującym, jest GID (group identifier) - identyfikator grupy (czytaj dalej), do której należy. Na podstawie tych danych podejmowane są decyzje, czy osoba posiada prawo dostępu do danych plików (opisane poniżej)

Informacja o numerach identyfikacyjnych zapisana jest w `/etc/passwd`. Plik taki może wyglądać następująco:

```
root:x:0:0:root:/root:/bin/bash
imoteph:x:1000:1000:SimpleUser:/home/imoteph:/bin/bash
```

Pola odznaczone są znakiem dwukropka ":". Pierwsze z nich określa login użytkownika. Z powyższego przykładu odczytać możemy że w systemie są dwaj użytkownicy o loginach root i imoteph.

Kolejne pole to hasło. Przy stosowaniu shadowingu haseł dzięki pakietowi shadow, hasła kodowane są algorytmem MD5 i umieszczane w `/etc/shadow`, a nie `passwd`, dlatego też zamiast naszego hasła w `/etc/passwd` widzimy "x".

Kolejne dwa pola to nasz UID (pierwsze) i GID (drugie).

Dalej mamy nazwę użytkownika, ścieżkę do katalogu i domyślną powłokę, w tym wypadku jest to Bash.

## Brak dostępu, czyli prawa do plików

W Linuksie każdy katalog, każdy plik posiada informację o prawach dostępu do niego. Informacja ta mówi nam, kto może otworzyć plik, kto zapisać, a kto go uruchomić. Każda flaga ma swoją skróconą postać:

r : read - plik do odczytu  
w : write - plik do zapisu  
x : execute - plik do uruchomienia

W przypadku katalogu "execute", czyli uruchomienie, oznacza wejście do jego wnętrza.

Dodatkowo każdy plik podaje prawa dostępu nie tylko dla właściciela, ale grupy do której należy właściciel oraz innych użytkowników.

Pojawiło się nowe pojęcie: "grupa". Użytkownicy w Linuksie mogą podzieleni być na kategorie. Dzięki temu administrator ma możliwość utworzenia np. grupy "nowi" i przypisania nowym użytkownikom przynależność do tej grupy. Pozwala to na łatwą administrację, a przede wszystkim szybkie określanie praw. Wystarczy bowiem, że administrator da prawo odczytu pliku grupie nowi, a wszyscy jej członkowie będą mieli możliwość jego czytania.

Tak więc chcąc określić prawa do pliku podajemy:

prawa\_dla\_właściciela prawa\_dla\_grupy prawa\_dla\_inych

Więc przypuśćmy, że chcemy nadać prawo odczytu i zapisu właścicielowi, pozostawiając innym użytkownikom możliwość tylko otwarcia pliku:

rw-r-r--

Znak "-" określa, że danej flagi użytkownik nie posiada.

A jak nadać wszystkie prawa właścicielowi, pozbawiając praw innych?

rwX-----

Istnieje także zapis praw za pomocą cyfr. Przedstawia to niżej zamieszczona tabelka:

Liczba	Odpowiednik znakowy	Opis
1	x	uruchomienie
2	w	zapis
4	r	odczyt

Określanie praw za pomocą cyfr jest wiele wygodniejsze i wiele krótsze. Aby określić prawa dla użytkownika należy dodać do siebie liczby odpowiadające prawom:

Prawo	Opis
1	Tylko uruchomienie
2	Tylko zapis
3	Zapis i uruchomienie
4	Odczyt
5	Odczyt i uruchomienie
6	Odczyt i zapis
7	Odczyt, zapis i uruchomienie

Wiec prawa określają czy mamy dostęp do pliku, czy też nie. Odwołanie się do pliku, do którego nie mamy dostępu powiedzie się zgłoszeniem komunikatu przez system:

Brak dostępu

## Graficzne środowisko pracy

Wiele osób pozostaje w dystansie od Linuksa, uważając, że wprowadzanie długich komend z linii poleceń może być trudne i nastęrczać problemów. Niestety osoby takie szybko zniechęcają się do tego systemu, nie chcąc nawet poznać całego oblicza Linuksa.

Rozdział ten ma na celu zapoznać Cię z uzgodnieniami jakie niesie ze sobą korzystnie z graficznego środowiska. Dzięki niemu pokochasz system i przekonasz się, że Linux wcale nie musi być trudny!

### Czym jest graficzne środowisko pracy?

Prawdziwa potęga Linuksa drzemie u jego podstaw - linii poleceń. Jednak całkowite jej poznanie może okazać się bardzo trudne dla początkującego. Dlatego też powstało graficzne środowisko pracy, czyli pulpit, na którym wykonywać możemy zamierzone działania.

Programiści do dziś doskonalą swoje środowiska, dzięki czemu możemy wykonać w nim mnóstwo rzeczy, nie uruchamiając konsoli. Powstają także różne odmiany, co pozwala nam na dobranie odpowiedniego dla nas. Możemy więc używać rozbudowanych, jak i skromnych, a co za tym idzie bardzo szybkich menedżerów.

### Jakie udogodnienia płyną z jego korzystania?

Graficzne środowisko powstało przede wszystkim po to aby ułatwić i umilić nam pracę w systemie. Korzystając więc z niego, możemy nie tylko cieszyć oko piękną grafiką, ale także możemy zacząć powoli wprowadzać się na stałe do domu Pingwinka.

Początkujący za pewne o wiele szybciej zrobią zamierzone przez siebie czynności, wykonując je w tym środowisku.

### Przegląd dostępnych środowisk

Od czasu pojawienia się pierwszych dostępnych dla Linuksa środowisk graficznych, programiści z całego świata udoskonalali już istniejące i pisali nowe. Zaowocowało to szeroką gamą wyboru, spośród których mamy możliwość używania najbardziej nam odpowiadającej.

Najpopularniejsze opisałem poniżej:

#### *KDE*

KDE - K Desktop Enviroment jest obecnie bardzo zaawansowanym zbiorem narzędzi, który rozwijany przez lata stał się wysoką pozycją. Środowisko posiada własny menedżer plików - Konqueror. Pracę bez wątpienia umila nam nietuzinkowa grafika, której dodatkowo mamy możliwość zmiany.

KDE posiada Centrum Sterowania - program, w którym dokonać możemy zmian, począwszy do ustawień graficznych, kończąc na sieci.

Informacje odnośnie KDE znajdują się oczywiście na oficjalnej stronie - [www.kde.org](http://www.kde.org).

Mamy także możliwość pobrania nowych ikon, ekranów uruchamiania (and. splash screen), dekoracji okien i wiele innych rzeczy ze strony [kde-look.org](http://kde-look.org).

### *GNOME*

Gnome jest drugim najbardziej rozwiniętym środowiskiem pracy. Wraz z KDE stoją na najwyższym podium "ryнку". Gnome korzysta z przenośnej biblioteki GTK+. Nie posiada on własnego menedżera plików, ale dostępny mamy tu SawFish oraz świetnego Nautiliusa. Tak jak w przypadku KDE, w Gnome wszystkich ustawień możemy dokonać w jednym miejscu - Centrum Sterowania Gnome.

Informacje odnośnie środowiska znajdziemy pod oficjalnym adresem - [www.gnome.org](http://www.gnome.org).

Istnieje także strona, która pozwoli nam upiększyć Gnome - [art.gnome.org](http://art.gnome.org).

### *Rodzina BlackBox*

BlackBox i jego odmiany takie jak Fluxbox zasługują na szczególną uwagę. Są to nieduże środowiska graficzne, lecz charakteryzujące się bardzo małymi wymaganiami sprzętowymi. Choć nie posiadają one ani ikon, ani żadnego menedżera plików, bez problemu możemy ściągnąć odpowiedni pakiety i zamienić naszego skromnego BlackBoxa w przyjazny interfejs. BlackBox dostępny jest praktycznie w każdej dystrybucji, więc warto go wypróbować. Jest idealnym rozwiązaniem dla starszych komputerów, dla których takie środowiska jak Gnome, czy KDE to zbyt wiele.

Oczywiście istnieje wiele więcej środowisk, jednak opisanie ich wszystkich zajęłoby dużo miejsca... Wspomnę tylko, że inne znane środowiska to WindowMaker oraz niedawno powstałe XPde - środowisko bazujące na interfejsie Windows XP.

## **U podstaw siła drzemie, czyli obsługa konsoli**

W poprzednim rozdziale zapoznałem Cię z wygodnym środowiskiem graficznym. To w nim postawisz pierwsze kroki, w nim będziesz powoli odkrywał tajniki Linuksa. Musisz jednak wiedzieć, że grafika i wygoda to nie wszystko... Aby w pełni posługiwać się systemem musisz poznać jego korzenie - konsolę!

### **Rzeczywista natura systemu**

Linux od podstaw pisany był na wzór Uniksa, dlatego też odziedziczył po nim wiele cech. Jedną z nich jest potęga płynąca z konsoli, a nie jak w przypadku innego znanego Ci systemu operacyjnego - graficznego środowiska pracy.

Pomimo, że każdemu początkującemu wydaje się, że to zbyt trudne nauczyć się tylu poleceń, jak się sam przekonasz po niedługim czasie, konsola jest niezbędną, a jej nauka to czysta przyjemność.

### **Początki poruszania się w konsoli**

Aby zacząć pracę z konsolą musisz poznać najpierw kilka pojęć i symboli charakteryzujących tą stronę systemu. Podrozdział ten ma na celu zapoznać Cię z nimi, ułatwiając Ci tym samym pracę. Skup się więc uważnie, co z pewnością zaowocuje łatwym przyswojeniem materiału,

który wbrew pozorom przyjmowanych przez początkujących - nie jest taki trudny.

Kiedy pracujesz z Linuksem, masz do dyspozycji kilka konsol, które możesz przełączać i wykonywać na nich oddzielną pracę. Domyślnie ich liczba wynosi 6, lecz możesz ją zmienić w każdym momencie.

Do przełączania konsol służą skróty klawiszowe:

Alt + Ctrl + klawisz F odpowiadający żadanemu numerowi - przełączenie z graficznego środowiska

Alt + klawisz F odpowiadający żadanemu numerowi - przełączenie z konsoli

Podczas pracy zauważyłeś z pewnością ciąg znaków poprzedzający Twoje polecenia:

```
adrian@localhost:~$
```

Tłumacząc go po kolei oznacza:

adrian - nazwa Twojego użytkownika, czyli Twój login

localhost - nazwa Twojego komputera; pod taką będzie widniał w sieci

\$ - można go określić jako stan użytkownika; symbol \$ oznacza zwykłego użytkownika, # - roota

Logując się będziesz posiadał oczywiście \$, chyba, że pracę rozpoczniesz jako superużytkownik!

Musisz wiedzieć, że "to", gdzie wpisujesz polecenia zwane jest powłoką. W Linuksie mamy do wyboru kilka powłok, np. oprócz standardowego Basha na płytках naszej dystrybucji z pewnością znajdziemy takie jak Zsh, czy Korn...

Bash jest domyślną powłoką w Linuksie i właśnie w oparciu o nią będziemy wklepywać przeróżne polecenia...

Ciekawym udogodnieniem Basha jest dokończanie nazw plików i poleceń. Wystarczy, że wpiszemy kilka liter i wciśniemy tabulator, a powłoka dokończy za nas oczekiwaną rzecz. Jednak gdy w danym katalogu mamy dwie podobnie brzmiące nazwy, np. 'skaner' i 'skaner\_portow', po wpisaniu "skan" i wciśnięciu tabulatora z brzęczyka naszego komputera usłyszymy "piknięcie". Bash informuje nas, że znalazł dwie podobne nazwy i aby dokończyć polecenie bądź nazwę pliku należy podać jeszcze dodatkowe informacje. Aby wyświetlić wszystkie możliwe "dokończenia" odnośnie podanego przez nas ciągu znaków, naciskamy dwa razy tabulator...

## **Midnight Commander - łatwej być nie może**

Aby ułatwić i czasem przyspieszyć pracę w konsoli, programiści napisali ciekawe narzędzie, za pewne znane Ci już, jeśli wcześniej korzystałeś z Windows. Midnight Commander, dostępny w większości dystrybucji, pozwala na przenoszenie, kopiowanie, edytowanie plików itp. Program w systemie dostępny jest pod nazwą "mc" (skrót od jego pełnej nazwy). Obsługi programu nie trzeba tłumaczyć - prosta i przejrzysta nie powinna nikomu sprawić trudu.

## **Polecenia**

Na początku chciałbym wytłumaczyć pojawienie się słowa "polecenie" w tej publikacji. Otóż samo one źle oddaje rzeczywiste przesłanie. Mówiąc "polecenie" mamy na myśli pewien rozkaz, pewną instrukcję. Ja mówiąc "polecenia" w publikacji mam na myśli odpowiedni program. Linux to nie "sławny" Dos. Tu wszystko jest od siebie oddzielone, tzn. w każdej chwili możemy nawet usunąć program odpowiadający na przenoszenie plików! Trudno jest mi jednak użyć tu pojęcia "program", co może wprowadzać Cię w zakłopotanie. Tak więc, będę nadal posługiwał się terminem "polecenie", musisz jednak pamiętać - Linux to nie polecenia!

Sposób wywołania aplikacji zależy od jej usytuowania - globalnego, czy lokalnego. Wpisując w linii poleceń (synonim konsoli):

```
imoteph@debian:~$ mc
```

Linux przeszuka katalogi z plikami uruchomieniowymi globalnymi, czyli /bin, /sbin, /usr/bin, /usr/sbin. Jeśli tam znajduje się wskazany przez nas plik, jak nie trudno się domyśleć - zostanie on uruchomiony.

Jak jednak uruchomić plik lokalny, np. z naszego katalogu?  
Wystarczy nazwę programu poprzedzić znakiem kropki i slesha - "./":

```
imoteph@debian:~$ ./"nazwa pliku"
```

Zauważ, że plik znajduje się w cudzysłowie. Należy tak podawać wszystkie nazwy zawierające spację. Innym na to sposobem jest wstawienie przed spacją znaku ukośnika - "\":

```
./nazwa\ pliku
```

Niżej podane są podstawowe polecenia, bez których praca w konsoli jest bezużyteczna:

#### ProgramOpis

ls	Wyświetla pliki w podanym katalogu; samo 'ls' wyświetli bierzący katalog imoteph@debian:~\$ ls
cd	Zmienia katalog na podany imoteph@debian:~\$ cd mp3/
cp	Kopiuje podany plik do podanej ścieżki imoteph@debian:~\$ cp skaner.cpp /home/inch/
mv	Przenosi podany plik do podanej ścieżki; może być użyty do zmiany nazwy pliku imoteph@debian:~\$ mv skaner.cpp /home/inch imoteph@debian:~\$ mv skaner.cpp skaner-1.0.cpp
cat	Wyświetla zawartość podanego pliku imoteph@debian:~\$ cat skaner.cpp
more	To samo, co 'cat', jednak możliwość przewijania w dół tekstu imoteph@debian:~\$ more skaner.cpp
less	To samo, co 'cat', jednak możliwość przewijania w obydwie strony imoteph@debian:~\$ less skaner.cpp
mkdir	Tworzenie katalogu imoteph@debian:~\$ mkdir linux
rm	Usuwa pliki imoteph@debian:~\$ rm skaner.cpp
chmod	Zmiana praw dostępu do pliku/katalogu imoteph@debian:~\$ chmod 666 skaner.cpp

```
chown  Zmiana właściciela pliku/katalogu
       imoteph@debian:~$ chown inch skaner.cpp
passwd Ustawia nowe hasło dla użytkownika
su     Tymczasowe zalogowanie na innego użytkownika; samo 'su' oznacza logowanie na
       konto roota
whoami Wyświetla nazwę użytkownika, w jakim obecnie pracujemy
mount  Montuje określone partycje w określonym miejscu
       imoteph@debian:~$ mount /dev/cdrom
```

Wyżej podane polecenia, to tylko namiastka tego, co oferuje nam Linux. Wszystko jednak zależy od zainstalowanego oprogramowania w naszym systemie.

Informacje na temat użycia danego programu znajdziesz bezpośrednio w podręczniku systemowym man. Wystarczy wpisać `man mount`, a na ekranie pojawi się opis, składnia i informacje na temat tego programu. Oczywiście mamy możliwość wyświetlenia "streszczenia" opcji po dodaniu argumentu `--help` lub (niekoniecznie występujący w każdym programie) `-h`. Niekiedy wystarczy wpisać samą nazwę programu, a program sam wyświetli sposób użycia i ewentualnie, jak możemy uzyskać informacje o nim!

Wielką zaletą systemowego Basha jest łączenie poszczególnych poleceń. Jest to bardzo pożyteczne, gdy np. chcemy wyświetlić zawartość dużego katalogu - np. `/usr/bin`. Nazwy plików nie zmieściłyby się na ekranie, więc nie ujrzelibyśmy ich wszystkich. Tu z pomocą idzie na chociażby wyżej wymieniony 'less':

```
imoteph@debian:~$ ls /usr/bin | less
```

Pionowa kreska "łączy" dane polecenia, dzięki czemu możemy w tym przypadku przewijać listę plików.

Aby ujrzeć wszystkie globalne programy, wystarczy, że wciśniemy dwa razy tabulator. Bash tylko spyta czy je wyświetlić. Poda oczywiście ile tych plików będzie, dzięki czemu łatwo możemy sprawdzić ile ich mamy w systemie!

## **Pakiety i źródła - instalacja oprogramowania**

Linux już zaraz po instalacji oferuje nam dużo. Nastąpi jednak moment, kiedy będziesz musiał coś zainstalować, aby wykonywać zamierzone cele. Rozdział ten ma na celu zapoznanie Cię z rodzajami rozpowszechnianego oprogramowania dla tego systemu. Nauczysz się także jak je instalować i odinstalowywać programy.

### **Czym są źródła?**

Linux jak już wcześniej wspomniałem jest na licencji GNU, która mówi o rozpowszechnianiu oprogramowania zawsze z kodem źródłowym. Także większość dostępnego oprogramowania dla tego systemu jest na tej licencji. Programy takie potocznie zwane są źródłami i aby je zainstalować musimy najpierw je skompilować. Aplikacje tego typu praktycznie zawsze są kompresowane, co zmniejsza oczywiście ich objętość.

### **Czym są pakiety?**

Najprościej tłumacząc, pakiety to skompilowane źródła, których instalacja to chwila. Właśnie w takiej formie znajduje się oprogramowanie w naszej dystrybucji. Pakiety mogą mieć różną postać, w zależności od naszej dystrybucji i sposobu zarządzania nimi. Trzy główne typy:

RPM - Red Hat Package - jak sama nazwa wskazuje, występują w rodzinie Red Hata.

DEB - Pakiety występujące w Debianie; nazwa pochodzi od trzech pierwszych liter nazwy dystrybucji

TGZ - archiwa tar; pakiety obecne w Slackware

Pomimo, że pakiety znacząco różnią się sposobem zarządzania nimi, łączy je wspólna cecha - pakiety są kompresowane, a podczas instalacji rozpakowywane.

## Instalacja programów ze źródeł i pakietów

Po ściągnięciu programu na dysk, z pewnością chcesz go zainstalować. Istnieją pewne kroki, które należy podjąć w tym celu. Aby zainstalować program ze źródeł, pierwsze co należy wykonać, to rozpakowanie pliku:

```
imoteph@debian:~$ tar -zxvf gettheport.tar.gz
```

Tar to program rozpakowujący. Zawarte w nim opcje to kolejno:

z - kompresja/dekompresja przez gzip (końcówka ".gz")

x - rozpakowanie plików z archiwum

v - wyświetla szczegóły związane z przetwarzanym plikiem

f - określa, który plik ma użyć

Aby rozpakować pliki tar.bz2, opcję -z zastępujemy opcją -j.

Możemy także rozpakowywać źródła tylko do tar. W tym celu dla plików \*.gz wpisujemy:

```
$ gunzip plik.tar.gz
```

W wyniku otrzymamy czyste archiwum tar. Jest to także przydatne, kiedy mamy plik skompresowany samym gzipem.

Aby w ten sam sposób rozpakować pliki \*.bz2 wykonujemy:

```
$ bunzip2 plik.tar.bz2
```

Następnie wchodzimy do katalogu poleceniem cd. Kolejnym krokiem jest wyświetlenie plików należących do tego katalogu, co pozwoli nam zorientować się, jakie czynności będziemy musieli wykonać w dalszym etapie.

Aby zrozumieć i zapamiętać instalację ze źródeł, musisz najpierw poznać istotę dwóch plików: `configure` i `makefile`. Ten pierwszy sprawdza, czy w systemie zainstalowane są wymagane biblioteki lub programy, co ustrzeże Cię przed błędami kompilacji. Innym ważnym plikiem jest wspomniany `makefile`. To on przeprowadza cały proces kompilacji programu. Dzieli się przeważnie na trzy części:

`make` - sekcja kompilująca źródła

`install` - sekcja instalująca skompilowane pliki

`uninstall` - sekcja odinstalowująca pliki zainstalowane

Tak więc, jeśli w katalogu znajduje się plik `configure`, to od niego powinniśmy zacząć proces instalacji. Dostępne jego opcje, widoczne są po uruchomieniu go z argumentem `-h` lub `--help`. Przed instalacją możemy więc określić interesujące nas opcje.

Po pomyślnym wykonaniu się skryptu `configure`, możemy przejść do etapu kompilacji, wydajemy więc polecenie `make`. Gdy i to wykona się pomyślnie, możemy zainstalować skompilowany już program poleceniem `make install`. Po chwili mamy go na dysku. Abyś w przyszłości mógł odinstalować oprogramowanie, musisz zachować jego źródła... `Make` bowiem zapisuje informacje co zostało przekompilowane i gdzie skopiowane. Katalog możesz spokojnie jednak spakować i wrzucić na płytę, a gdy stwierdzisz, że program nie jest Ci już potrzebny, wystarczy, że rozpakujesz zachowane źródła i wpiszesz:

```
make uninstall
```

Gdy brakuje pliku `configure`, tzn. nie zamieszczono go, bardzo prawdopodobne jest, że ściągnięte archiwum, to nie źródła, lecz plik binarne. Jeżeli jest `makefile`, wystarczy, że wpiszesz `make` lub `make install`, a pliki skopiują się do odpowiednich katalogów.

Inną sytuacją jest całkowity brak `configure` i `makefile`. Jest to typowe dla dużych programów zajmujących po kilkadziesiąt, kilkaset MB. Przykładem może być tu świetny pakiet biurowy `OpenOffice.org`. Nieopłacalne byłoby jego kompilowanie, więc autorzy stwierdzili, że lepiej będzie rozprowadzanie go domyślnie w formie binarnej (oczywiście kod źródłowy dostępny jest na oficjalnej stronie projektu). Dołączają jednak plik instalacyjny `setup`, który po prostu uruchamiamy, tak jak już wcześniej to omawiałem. Często do tego typu programów dołączany jest także skrypt Basha, będący zazwyczaj pod nazwą `install.sh`. Sposób instalacji jest identyczny jak w wyżej wymienionym `OpenOffice.org`.

Instalacje pakietów dla poszczególnych dystrybucji jest wiele prostsza, niż instalacja programów ze źródeł. Ja jednak nie będę opisywał dokładnie tego procesu. Wszystkie dostępne opcje możemy uzyskać uruchamiając program zarządzający pakietami z argumentem `-h` lub `-help`.

#### *RPM:*

```
rpm -i nazwa_pakietu.rpm - instalacja
rpm -q nazwa_pakietu.rpm- wyświetla informacje o pakiecie
rpm -r nazwa_pakiety.rpm - usunięcie
```

#### *DEB:*

```
dpkg -i nazwa_pakietu.deb - instalacja
apt-get remove nazwa_pakietu.deb - usunięcie
dpkg --info nazwa_pakietu.deb - wyświetla informacje o pakiecie
dpkg --reconfigure nazwa_pakietu.deb - ponowna konfiguracja pakietu
dpkg --list nazwa_pakietu.deb - wyświetlenie listy pakietów o podanym wzorcu nazwy
dpkg --unpack nazwa_pakietu - rozpakowanie pakietu
```

#### *TGZ:*

```
installpkg nazwa_pakietu.tgz - instalacja
removepkg nazwa_pakietu.tgz - usunięcie
```

Jak wcześniej wspomniałem - aby uzyskać wszystkie dostępne opcje, uruchom program zarządzający z argumentem `-h` lub `-help`. Alternatywnym rozwiązaniem jest zajrzenie do podręcznika systemowego, gdzie znajdziemy szczegółowy opis każdej funkcji (czytaj na końcu dokumentu).

Nie każdy jednak na początku chce instalować programy przy pomocy powłoki. Dlatego też istnieją graficzne programy, które ułatwią nam to zadanie. Chyba najlepszym wyborem na początek będzie `kpackage`.

`Kpackage` to narzędzie dostępne wraz ze środowiskiem KDE. Służy do instalowania pakietów RPM, DEB TGZ Slackwara, a nawet pakiety rodziny BSD. Zaraz po starcie programu ujrzymy dwa okna - listę pakietów oraz okno informacji o danym pakiecie. Na bocznym pasku znajdziemy takie przyciski, jak `Znajdź pakiet`, `Znajdź plik`, `Rozwiń drzewo` i `Zwiń drzewo`. Dodatkowo okno z listą podzielone jest na zakładki, dzięki czemu szybko sprawdzimy, jakie pakiety są zainstalowane, jakie nowe, a jakie aktualizowane.

By zainstalować pakiet należy po prostu na niego kliknąć. W oknie obok pokaże się informacja o pakiecie a niżej podświetli się przycisk instalacji. Po jego wciśnięciu zostaniemy oczywiście zapytani o hasło roota.

`Kpackage` daje nam możliwość instalowania także wielu pakietów na raz. W tym celu należy klikać na pola obok pakietów. Na ich miejscu ujrzymy "V", co oznacza, że pakiet jest zaznaczony. Teraz wystarczy nacisnąć `Zainstaluj zaznaczone`, pod listą pakietów.

`Kpackage` to wygodne narzędzie, które idealnie nadaje się dla osób początkujących. Istnieją także inne programy, które są pisane wraz z systemem pakietów, i tak w Debianie mamy `APT` oraz `Dsececta`, a w Slackware `PKGTool`.

## Urządzenia

Linux od początku był pisany, aby można było w nim obsłużyć wiele urządzeń różnego typu. Dlatego też powinieneś wiedzieć jak system je traktuje. Z rozdziału tego dowiesz się przede wszystkim jak są one reprezentowane i jak system się z nimi komunikuje...

### Jak reprezentowane są urządzenia w Linuksie?

W `/dev` (skrót od `devices` - urządzenia) znajdują się pliki, przez które następuje komunikacja ze sprzętem. Widoczne są tu takie pliki, jak `modem`, `mouse` itp. Tak właśnie Linux traktuje urządzenia. W ww. katalogu znajdują się jakby informacje, gdzie nasz sprzęt jest zainstalowany.

W konfiguracji wielu programów musimy właśnie podać ścieżki do urządzeń z tego katalogu.

A więc aby przesłać komunikat jakiemuś urządzeniu wystarczy przesłać go do odpowiedniego pliku. Sytuacja jest identyczna w przypadku odczytu komunikatów zwracanych przez sprzęt. Aby lepiej się temu przyjrzeć wystarczy na przykład napisać prosty program w `c++`:

```
#include <iostream>
#include <fstream>
```

```

#include <string>

using namespace std;

int main()
{
    ofstream modem("/dev/modem");
    ifstream od_modemu("/dev/modem");
    string odpowiedz;
    modem << "ATZ\n";
    while(getline(od_modemu, odpowiedz))
    {
        cout << odpowiedz << endl;
        exit(1);
    }
}

```

Program kompilujemy za pomocą g++:

```
$ g++ modem.cpp -o modem
```

Opcja -o określa plik wyjściowy (przekompilowany).

I uruchamiamy:

```
$ ./modem
```

Inną metodą, właściwie dużo łatwiejszą i szybszą, jest przesłanie komunikatu do modemu z poziomu Basha:

```
$ echo "ATZ" > /dev/modem
```

Modem w odpowiedzi na pytanie powinien po prostu piknąć. Jeśli jednak tego nie wykonał, to bardzo prawdopodobne, że widnieje on pod innym adresem. Nie zapominajmy, że /dev/modem to tylko dowiązanie symboliczne, czyli "odsyłacz". Modem tak naprawdę jest (prawdopodobnie) pod /dev/ttySx, gdzie x to numer portu modemu (**UWAGA:** w Unikсах liczenie zaczyna się od 0). Tak więc możesz poprawić dowiązanie:

```
# ln -s /dev/ttySx /dev/modem
```

bądź po prostu zmienić w programie czy poleceniu echo adres...

I tak na przykład możesz za pomocą echo odtwarzać muzykę! Tak, tak - trochę dziwnie to brzmi, ale to prawda... Jest tylko jedno ale - odtwarzane pliki nie mogą być kodowane, tak jak sławne ogg, czy mp3. A więc jak?

```
cat /home/imoteph/sound.au > /dev/audio &
```

Na koniec tego tematu wytłumaczę jeszcze dwa symbole, które się tu pojawiły: ">" i "&". Pierwszy z nich oznacza przekierowanie standardowego wejścia na dany plik z jego nadpisaniem. ">>" także oznacza przekierowanie, tylko z tą różnicą, że plik docelowy nie jest nadpisywany, lecz dane są do niego dopisywane.

"&" oznacza pracę procesu (programu) w tle. Dzięki temu będziemy mogli wykonywać inne czynności podczas słuchania muzyki (oczywiście mowa o pracy na konsoli, z której został

uruchomiony proces).

Na temat urządzeń nie musisz wiedzieć zbyt dużo. Na początek wystarczy Ci miejsce ich pobytu i sposób komunikacji z nimi, a wraz z każdym krokiem na przód sam poznasz tajniki obsługi sprzętu przez Linuksa.

### **Winsprzęt - kłopot?**

Za pewne z samej nazwy domyślasz się co oznacza owa nazwa... No właśnie dziś sprzęt komputerowy tanieje na naszych oczach. Ale czy tanieje, bo wychodzą coraz to nowsze urządzenia? I tak, i nie... Producenci często wydają na rynek tak zwany winsprzęt - urządzenia "przystosowane" do pracy z Windows. Bardzo często widać to w modemach, które nabyć można już nawet za 60 zł! Z racji, że na dołączonej płycie znajdziemy tylko sterowniki dla Windows, możemy mieć kłopoty z zainstalowaniem sprzętu w Linuksie. W najlepszym wypadku możemy poszukać nieoficjalnych sterowników w Internecie, często się jednak zdarza, że zakupiony przez nas sprzęt będziemy musieli odłożyć na półkę.

Kupując nowe urządzenia powinniśmy zapoznać się z informacją, z jakimi systemami sprzęt jest w pełni kompatybilny. Informację taką możemy uzyskać choćby od sprzedawcy w sklepie lub na oficjalnej stronie producenta. Dobrze jest także spytać na grupach dyskusyjnych (adresy na końcu), czy miał ktoś problemy, z wybranym przez nas sprzętem.

### **Instalacja urządzeń**

Większość urządzeń wykryło odpowiednio już skonfigurowane jądro Linuksa, bądź instalator dystrybucji. Zdarza się jednak, że instalator nie zainstaluje naszych wszystkich urządzeń, a jądro nie ma wkompileowanych odpowiednich modułów (czytaj dział Jądro Linuksa). W tym rozdziale pomogę zainstalować Ci kartę dźwiękową, modem, klawiaturę multimedialną i drukarkę.

### ***Dźwięk w Linuksie***

Rzadko zdarza się, że w systemie nie działa nasza karta dźwiękowa. Wiele częściej występuje problem praw dostępu do tych urządzeń, przez co programu nie mogą korzystać ze sprzętu. Pierwszym krokiem, który wykonamy, to właśnie sprawdzenie praw. W tym celu logujemy się na konto roota i sprawdzamy do jakiej grupy należą pliki urządzenie dźwięku:

```
woody:/home/imoteph# cd /dev/  
woody:/dev# ls -al | less
```

Ujrzymy długą listę... teraz skupimy się na lini z plikiem audio:

```
crw-rw----  1 root      audio      14,  4 mar 14  2002 audio
```

Widać, że właścicielem pliku jest root. Plik należy także do grupy audio. Najlepszym sposobem jest teraz dodanie swojego użytkownika do tej grupy. Służy do tego polecenie adduser:

```
# adduser imoteph audio
```

Powinniśmy ujrzeć komunikat o dodaniu użytkownika do grupy. Jeśli jednak adduser zwrócił informację, że użytkownik należy już do tej grupy, prawdopodobne jest, że to nie w prawach

dostępu, lecz w sterownikach leży problem.

Neleży teraz sprawdzić na jakim chipsecie jest nasza karta dźwiękowa. Najlepiej jest to zrobić przy pomocy polecenie lspci:

```
imoteph@woody:~$ lspci -vvv
```

Program wyświetli informacje o naszym sprzęci podłączonym do PCI, a także zintegrowanym z płytą główną. W moim przypadku podał, że kontroler dźwięku obsługuje chipset nvidii - nForce2.

Teraz musimy sprawdzić jakie moduły mamy dostępne w jądrze. Wpisujemy:

```
woody:/dev# lsmod
```

Oto wyniki na moim Debianie, z rekompilowanym jądrze 2.4.20:

Module	Size	Used by	Tainted: P
ppp_deflate	3008	0	(autoclean)
zlib_deflate	18048	0	(autoclean) [ppp_deflate]
zlib_inflate	18688	0	(autoclean) [ppp_deflate]
bsd_comp	4032	0	(autoclean)
ppp_async	7008	0	(autoclean)
ppp_generic	18732	0	(autoclean) [ppp_deflate bsd_comp]
ppp_async]			
slhc	4736	0	(autoclean) [ppp_generic]
hcfpciserial	17952	0	(autoclean)
hcfpciengine	637944	0	(autoclean) [hcfpciserial]
hcfpciassoc	15272	1	(autoclean) [hcfpciserial hcfpciengine]
i810_audio	22464	1	
ac97_codec	10304	0	[i810_audio]
nvnet	25824	2	
dummy	1088	0	(unused)
nvidia	1467808	10	

Teraz wystarczy załadować odpowiedni moduł: i810\_audio i możemy cieszyć się dźwiękiem. Zdarza się nieraz jednak, że nie mamy do dyspozycji modułów ze sterownikami do naszego sprzętu. Rozwiązaniem okazuje się rekompilacja jądra (jak w moim przypadku). Oto lista dostępnych opcji w sekcji Sound jądra 2.4.20:

```
ALi5455 audio support
BT878 audio dma
C-Media PCI (CMI8338/8738)
Creative SBLive! (EMU10K1)
Crystal SoundFusion (CS4280/461x)
Crystal Sound CS4281
Ensoniq AudioPCI (ES1370)
Creative Ensoniq AudioPCI 97 (ES1371)
ESS Technology Solo1
ESS Maestro, Maestro2, Maestro2E driver
ESS Maestro3/Allegro driver (EXPERIMENTAL)
ForteMedia FM801 driver (EXPERIMENTAL)
Intel ICH (i8xx), SiS 7012, NVidia nForce Audio or AMD 768/811x
RME Hammerfall (RME96XX) support
S3 SonicVibes
Trident 4DWave DX/NX, SiS 7018 or ALi 5451 PCI Audio Core
Support for Turtle Beach MultiSound Classic, Tahiti, Monterey
Support for Turtle Beach MultiSound Pinnacle, Fiji
```

VIA 82C686 Audio Codec  
OSS sound modules

### **Modem**

Modem to podstawowe drzwi na świat Internetu. Linux bez sieci, nie jest już tym samym Linuxem, dlatego w systemie powinieneś skonfigurować także swój modem. Linux powinien bezproblemowo wykryć wszystkie modemu zewnętrzne, z racji, iż są to prawdziwe sprzętowe urządzenia, nie potrzebujące sterowników do swojej obsługi. Inaczej jest jednak z modemami wewnętrznymi. Bardzo często zdarza się, że są to urządzenia, które potrzebują wykorzystywać procesor główny, zamiast swoich układów. Pisane są wtedy dla nich sterowniki, aby system mógł z nich korzystać. Niestety tylko pod jeden system zazwyczaj pisane są takie sterowniki - pod Windows, przez co inne systemy, w tym Linux nie mogą używać takich modemów.

Na szczęście jest na rynku kilka modemów wewnętrznych, które działają pod Linuxem. Obecnie chyba tylko firma Pentagram produkuje tanie modemy, do których dołączają także starowinki dla Linuksa.

Innymi modemami dość dobrze pracującymi pod Linuxem są modemy firmy Zoltrix. Ta co prawda nie dostracza żadnych sterowników, ale znalazły się osoby, które napisały takie dla modemów pracujących na chipsecie Conexant. W ten sposób możemy łączyć się z Internetem modemami na tych układach.

Przed pobraniem sterowników musimy sprawdzić jakim modemem jest nasz modem. Oczywiście posłużu nam do tego lspci. Oto wyniki na moim komputerze:

```
Communication controller: Conexant HCF 56k Data/Fax/Voice/Spkp (w/Handset)
Modem (rev 08)
```

Widać, że modem pracuje pod tym chipsetem. Dodatkowa informacja to typ - HCF. Udajemy się teraz pod [www.linuxant.com](http://www.linuxant.com) i pobieramy najnowsze starowinki dla modemów Conexant HCF. Instalacja jest dziecinnie prosta. Musimy wykonać tylko parę poleceń i możemy sprawdzać modem. Najpierw upenijmy się że w katalogu /usr/src/linux mamy źródła, bądź nagłówki obecnego jądra. Jeśli okaże się, że nie, sterowniki nie skompilują się. Dlatego też musimy wcześniej zainstalować paczkę z odpowiednimi plikami. Włączamy w tym celu dowolny menedżer pakietów i przeszukujemy listę ciągami: "kernel", "header" bądź "source". Po znalezieniu odpowiedniego pakietu oczywiście go instalujemy. Musimy sprawdzić jeszcze, czy napewno źródła dostępne będą w /usr/src/linux. Jeżeli tak nie jest tworzymy dowiązanie symboliczne:

```
# ln -s /usr/src/linux/ /usr/src/katalog_jadra/
```

Przechodzimy następnie do samej już instalacji sterowników:

1. rozpakowujemy pobrany plik,
2. przechodzimy do katalogu z rozpakowanymi sterownikami poleceniem 'cd',
3. wydajemy polecenie z roota 'make install'

Po chwili będziemy mieli już gotowe moduły w jądrze. Teraz należy je odpowiednio skonfigurować. My posłużymy się automatyczną konfiguracją modemu przez program hcfpciconfig. Uruchamiamy program i odpowiadamy twierdząco na pytania. Na samym końcu program zapyta nas o ścieżkę do jądra. Podajemy oczywiście /usr/src/linux/.

Od tej chwili nasz modem powinien już bezproblemowo działać. Jedyne co powinniśmy teraz zrobić to odpowiedni link do niego:

```
# rm /dev/modem
# ln -s /dev/modem /dev/ttySHCF0
```

Sprawdzamy czy wszystko jest w porządku:

```
$ echo "ATZ" >> /dev/modem
```

Urządzenie w odpowiedzi powinno piknąć. Teraz możemy już surfować po sieci, używając do tego aplikacji z naszej dystrybucji!

### ***Klawiatura multimedialna***

Klawiatury multimedialne bardzo przydają się podczas słuchania muzyki, oglądania filmów, a także pracy w sieci. Dlatego też powinniśmy mieć dostępną obsługę tago urządzenia w Linuksie.

Dość dobrym narzędziem do konfiguracji dodatkowych klawiszy jest program HotKeys. Powinien być on dostępny w większości dystrybucji, więc uruchamiamy menadżer pakietów i instalujemy go. Jedynym minusem programu jest fakt, że działa tylko z serwerem X, czyli w graficznym środowisku pracy, choć konfiguruje się go z poziomu powłoki. Za każdym razem, gdy zostaje uruchomiony trzeba podawać mu nazwę klawiatury do obsługi. Lista obsługiwanych modeli dostępna jest pod opcją '-l'. Szybko zorientujesz się że program ma dostępnych zaledwie kilka modeli, a Twojej klawiatury prawdopodobnie wśród nich nie ma. Jedyne co nam pozostaje to stworzenie własnych definicji. Na początek może trochę informacji na temat plików HotKeys. Program posiada plik konfiguracyjny hotkeys.conf w /etc/. W /usr/share/hotkeys/ znajdują się pliki z definicjami różnych modeli. Pliki te mają postać zbliżoną do:

```
<Opcja keycode="numer" />
```

Czyli abyśmy mogli korzystać z dodatkowych klawiszy, potrzebujemy ich kody. Pomoże nam teraz program Xev. Jeśli nie mamy go zainstalowanego, to znajdziemy go na płytce dystrybucji. Uruchamiamy program i wciskamy interesujące nas klawisze. Program następnie wyświetli potrzebne nam kody.

Następnie edytujemy dowolny plik z katalogu z definicjami. W miejsca starych kodów podstawiamy, nowe. Zmieniamy jeszcze sekcję 'model', gdzie podajemy nazwę naszej klawiatury. Definicje dla mojej klawiatury przedstawiają się następująco:

```
<?xml version="1.0"?>
<CONFIG model="IBM kb99-30">

<!-- Traki -->

<PrevTrack keycode="164" />
<Play keycode="162" />
<Pause keycode="160" />
<NextTrack keycode="146" />

<!-- Audio -->

<VolUp keycode="163" adj="1" />
```

```
<VolDown keycode="161" adj="1"/>
<Mute keycode="158"/>

<!-- Internet -->

<Email keycode="178"/>
<Search keycode="166"/>
<WebBrowser keycode="165"/>
```

```
</CONFIG>
```

Plik zapisujemy do katalogu w postaci znawa.def. W moim przypadku - IBMkb-9930.def. Jeśli nic nie pomyliliśmy, to po wpisaniu w powłoce:

```
$ hotkeys -l IBM-kb9930
```

Powinniśmy ujrzeć ekran powitalny HotKeys, a system powinien reagować na wciskane klawisze dodatkowe.

HotKeys posiada bardzo wygodny sposób definiowania skrótów. Dzięki temu mamy możliwość wykonywania operacji, których autor nie zawarł w programie. Definicji dokonuje się bardzo prosto, np:

```
<userdef keycode="230" command="/usr/bin/mozilla -remote
'openURL(http://google.com)'">Go to URL</userdef>
```

Linie taką oczywiście dołączamy do pliku definiującego.

Jeśli wszystko działa tak jak chcieliśmy pora zabrać się za globalny plik konfiguracyjny. Edytujemy więc /etc/hotkeys.conf. W sekcji 'Kdb' warto podać naszą klawiaturę, dzięki czemu nie będziemy musieli podawać jej podczas startu programu. Opcje ustawiamy wedle naszych upodobań. Zatrzymajmy się jednak przy opcjach OSD (wytświetlania na ekranie). Domyślny kolor zielony lekko razi, proponuję więc wybrać inny, np. pomarańczowy. Kolor ustawiamy w sekcji osd\_color. Wartość powinna być w postaci #xxxxxx.

Także wyświetlanie na dole ekranu to nienajlepsze rozwiązanie. Aby to zmienić ustawiamy opcję osd\_offset na 1/2 wysokości ekranu, czyli np. na 384.

Zmienić możemy także czas wyświetlania. Dokonujemy tego za pomocą osd\_timeout. Tak wygląda mój plik hotkeys.conf:

```
Kbd=IBMkb-9930
CDROM=/dev/cdrom

PrevTrack=xmms --rew
Play=xmms --play
Stop=xmms --stop
Pause=xmms --pause
NextTrack=xmms --fwd

WebBrowser=mozilla
Email=evolution
Search=mozilla -remote 'openURL(http://google.com) '

#czerwony:
#osd_color=#aa0000
```

```
#niebieski:
#osd_color=#005cad

#pomaranczowy
osd_color=#ff9400

osd_timeout=2
osd_position=bottom
osd_offset=384
```

Ostatnim krokiem, jaki wykonamy, to utworzenie skryptu startowego, dzięki czemu program włączy się automatycznie po starcie np. KDE. Uruchamiamy dowolny edytor umieszczamy w pliku:

```
#!/bin/bash
hotkeys
```

Skrypt zapisujemy do .kde/Autostart w naszym katalogu domowym. Restartujemy KDE i możemy cieszyć się naszą klawiaturą w Linuksie.

### ***Instalacja drukarki***

Drukarka jest bardzo przydatnym urządzeniem, czy to w biurze, czy też w zwykłym domu. W instalacji pomoże nam system drukowania CUPS. Powstał w 1999 roku i oferuje mnóstwo funkcji, zaczynając na sprawnym drukowaniu dokumentów na naszej maszynie, kończąc na bardzo dobrej obsłudze zadań drukowania w sieci lokalnej.

Będziemy potrzebować kilka pakietów, aby drukowanie było jak najbardziej wygodne. Uruchamiamy więc menadżer pakietów i instalujemy:

```
cupsys
cupsys-client
qtcups
```

Po instalacji system CUPS powinien być już gotowy do pracy. Możemy to sprawdzić wpisując w dowolnej przeglądarce adres <http://127.0.0.1:631>. Wszystkie opcje drukarki możemy ustawić właśnie tutaj.

Kolejny nasz krok, to instalacja drukarki w systemie. Najlepiej teraz udać się na stronę [www.linuxprinting.org](http://www.linuxprinting.org). Znajduje się tam ogromna baza sterowników do drukarek dla systemu CUPS. Możemy także sprawdzić jak nasza drukarka sprawuje się w Linuksie, czy może sprawia kłopoty, itp.

Po odnalezieniu naszego modelu pobieramy odpowiedni dla niego plik PPD. Dla mojej drukarki HP 640c jest to plik HP-DeskJet\_640C-hpijs.ppd. Następnie kopiujemy go do odpowiedniego katalogu z plikami CUPSA:

```
# cp HP-DeskJet_640C-hpijs.ppd /usr/share/cups/model/
```

i zmieniamy prawa na odczyt-zapis, odczyt, odczyt:

```
# chmod 644 HP-DeskJet_640C-hpijs.ppd
```

Teraz plik powinien być już dostępny w menu sterowników drukarek. Restartujemy jeszcze

tylko CUPSa poleceniem

```
# /etc/init.d/cupsys restart
```

i możemy przejść do dodawania drukarki.

W przeglądarce otwieramy panel CUPSa i przechodzimy do zakładki Printers, następnie Add printer. W oknie podajemy login root i oczywiście hasło administratora.

W polu Name podajemy nazwę dla drukarki, np. HP. Pozostałe pola możemy zostawić puste. Przechodzimy dalej. Z listy Device wybieramy typ podłączenia drukarki. W moim przypadku jest to port równoległy, więc wybieram Parallel Port#1.

Jeśli nic nie pomyliliśmy podczas dodawania pliku PPD, to teraz na liście drukarek powinien być widoczny także nasz sprzęt. Po jego wybraniu i zatwierdzeniu nasz drukarka powinna być już gotowa do pracy.

Z powrotem przechodzimy do zakładki Printer, gdzie możemy wydrukować stronę testową. Dodatkowo w zakładce Configure printer możemy dokonać ustawień jakości wydruku itp rzeczy. Jeśli coś pomyliliśmy podczas dodawania drukarki, możemy to poprawić w zakładce Modify Printer.

Aby sprawnie drukować użyjemy dodatkowo programu qtcups, dzięki któremu będziemy mogli odwoływać się do CUPSa z innych programów, takich jak OpenOffice, czy Mozilla.

Wystarczy tylko zmienić polecenie drukowania w tych programach z lp lub lpr na qtcups z opcją -d określającą nazwę drukarki. Przykładowo może być to:

```
qtcups -d HP
```

Jak widać instalacja drukarki przy pomocy systemu CUPS nie jest taka trudna. System ten potrafi obsłużyć wiele modeli, także nie powinienes mieć problemu z instalacją swojej drukarki.

## Jądro Linuksa

Mam nadzieję, że dotychczas poruszone tematy stanowią dla Ciebie porządną podstawę. Czas na poznanie środka systemu - jego jądra. W rozdziale mam zamiar przedstawić kernel Linuksa.

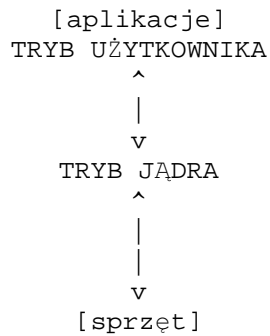
Uwaga: rozdział ten przeznaczony jest dla osób, które znają podstawy funkcjonowania komputera!

### Tryby pracy

Na początku, kiedy pojawiły się pierwsze komputery, programy, które uruchamiane były przez użytkowników potrafiły wywalić całą maszynę. Działo się to dlatego, że wszystko pracowało w tym samym trybie, na tym samym poziomie.

Rozwiązaniem problemu okazało się podzielenie pracy komputera na dwa tryby - tryb użytkownika (user mode) oraz tryb jądra (kernel mode).

W trybie kernela komputer przede wszystkim komunikuje się ze sprzętem i operuje pamięcią. Tryb użytkownika ma możliwość jedynie uruchamiania aplikacji. Sytuacja przedstawia się mniej więcej następująco:



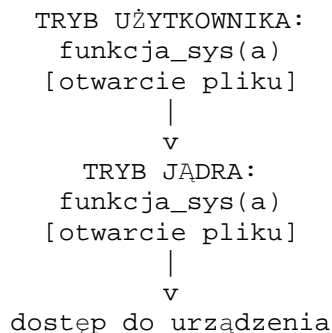
Czyli użytkownik nie komunikuje się bezpośrednio ze sprzętem. Zanim nastąpi transfer z urządzeniem tryb użytkownika musi przejść w tryb jądra.

Kiedy więc następuje przełączenie z jednego trybu do drugiego? Są dwa zasadnicze przypadki:

*Funkcje systemowe (system calls):*

- dostęp wejścia/wyjścia (I/O) urządzeń lub plików (np. czytanie, zapisywanie)
- potrzeba uzyskania informacji uprzywilejowanych, np. PID
- potrzeba zmiany wykonywanego procesu (rozwidlanie procesów, wykonywanie innych aplikacji)
- potrzeba wykonania poleceń, takich jak chmod, kill

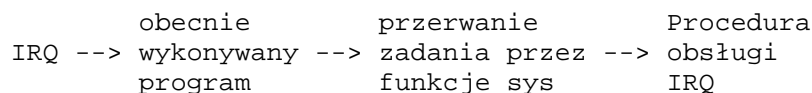
Można zilustrować to w następujący sposób:



Gdy użytkownik zamierza odczytać plik, komputer wykonuje funkcję systemową jego trybie. Ta zaś najpierw przechodzi w tryb jądra, zanim otrzyma dostęp do sprzętu.

*Przerwania*

Kiedy nadchodzi IRQ, program, który jest obecnie wykonywany zostaje przerwany na rządzanie procedury obsługi IRQ (IRQ Handler). Kiedy już przerwanie zostanie obsłużone, wykonywanie programu zostaje z powrotem przywrócone, tak jak gdyby nic się nie stało. Wygląda to następująco:



[tryb użytkownika]

[tryb jądra]

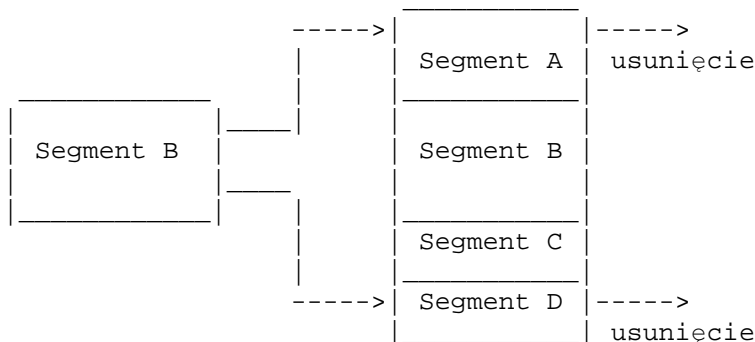
Zaraz po zakończeniu obsługi przerwania przez Procedurę obsługi IRQ, praca programu znów powraca do normy. Oczywiście najpierw zadanie wraca do trybu użytkownika.

## Operowanie pamięcią w Linuksie

Pierwszą metodą podziału pamięci jest segmentacja. Pozwala na kompilację programów, bez martwienia się o problem z umiejscowieniem ich w pamięci. Segment możemy podzielić na trzy zasadnicze części:

- stos
- dane (zainicjalizowane oraz niezainicjalizowane)
- tekst (kod programu, stałe)

Niestety segmentacja ma pewne wady. Wyobraźmy sobie taką sytuację:



Chcemy usunąć proces A (w segmencie A) oraz proces D (w segmencie D). Na jego miejsce chcemy "włożyć" proces B. Jak widać mamy odpowiednio dużo miejsca aby to wykonać, ale... jednak zrobić tego nie możemy. Segment to jedna ciągła arena, nie możemy więc jej dzielić.

Tu z pomocą idzie nam stronicowanie (pagination). Polega ono na dzieleniu pamięci na "n" fragmentów, z których każdy ma ustalony rozmiar. Proces może być załadowany do jednej lub kilku stron, a kiedy pamięć jest zwalniana (proces ginie), strony także są zwalniane.

Stronicowanie nie jest także pozbawione wad, a raczej problemów wynikających z niego... Gdy jakiś proces zostanie załadowany do strony, cały jej obszar jest zajęty. Jest to całkiem logiczne. Jak już wcześniej wspomniałem, strony są przecież określonymi rozmiarami. I dlatego też nawet jeśli proces zajmuje mniejszy obszar pamięci, niż pojedyncza strona, wolne miejsce jest bezużyteczne.

Nasuwa się więc pytanie, czy możliwe jest rozwiązanie obu problemów na raz? Oczywiście, że tak. Proces A w segmencie A możemy stronicować, czyli go dzielić (problem segmentacji rozwiązany), a dodatkowo możemy określić bardzo małe stony, co da man większą optymalizację pamięci (problem stronicowania).

## Funkcjonowanie sieci

Jak wiadomo, Linux oparty jest na Uniksie. Ten zaś stworzony został na potrzeby Internetu. Stąd też system idealnie sprawuje się jako maszyna podpięta do sieci, zarówno w roli klienta, jak i serwera. Standard ISO-OSI wyróżnia siedem głównych poziomów architektury sieci:

1. Warstwa fizyczna (np.: PPP, Ethernet)
2. Warstwa łącza danych (np.: PPP, Ethernet)
3. Warstwa sieciowa (np.: IP, X.25)
4. Warstwa transportowa (np.: TCP, UDP)
5. Warstwa sesji (SSL)
6. Warstwa prezentacyjna (kodowanie FTP ascii-binary)
7. Warstwa aplikacji (np.: przeglądarka internetowa)

Pierwsze dwa punkty są zazwyczaj implementowane przez sprzęt, kolejne dwa, jak można się domyśleć - programowo.

W warstwie fizycznej, jądro odbiera lub wysyła komunikaty do sprzętu sieciowego, czyli mówić krótko - komunikuje się z nim. W etapie drugim (warstwie łącza danych) jądro musi sprawdzić poprawność pakietu. Następnie w warstwie sieciowej następuje ustalanie routingu (transportowania), czyli dodawane są adresy IP. Warstwa transportowa ma za zadanie dołączyć do pakietu nagłówki i całość przesłać do odpowiedniego gniazda (np. portu 25).

## Budowa jądra

Jądro Linuksa ma charakter monolityczny. Oznacza to, że jest to całość zbudowana, z mniejszych dobrze przemyślanych komponentów. Przeciwnieństwem tego jest mikrojądro, np. Mach 3.0. Ma ono za cel minimalizację jądra, poprzez ograniczenie jego komponentów.

Linux jest kompilowany i łączony (linkowany) dynamicznie. Cóż to oznacza? W każdej chwili możemy załadować do niego jakiś fragment kodu, bądź go usunąć. Pozwala to na między innymi zmniejszenie rozmiarów kernela, przyspieszając tym samym jego pracę, a skracając czas ładowania. Są to tak zwane moduły. Aby przekonać się, jakie moduły mamy obecnie do dyspozycji, wystarczy wydać polecenie `lsmod`. Wynik może być następujący:

```
debian:/home/imoteeph# lsmod
Module                Size  Used by    Tainted: P
parport_pc            14980  1  (autoclean)
lp                    6464  0  (autoclean)
parport               24576  1  (autoclean) [parport_pc lp]
apm                   9116  2  (autoclean)
vfat                  9276  1  (autoclean)
msdos                 4860  0  (unused)
fat                   29112  0  [vfat msdos]
ppp_async             6176  0  (unused)
ppp_generic          15720  0  [ppp_async]
slhc                  4352  0  [ppp_generic]
dummy                 928  0  (unused)
nvidia               1467232 10
ide-scsi              7360  0
```

Teraz gdy np. potrzebujemy załadować moduł `ide-scsi`, wpisujemy `lsmod ide-scsi`, co wygląda tak:

```
debian:/home/imoteeph# insmod ide-scsi
Using /lib/modules/2.4.18/kernel/drivers/scsi/ide-scsi.o
```

Równie łatwo możemy usunąć moduł. Służy do tego `rmmod`.

Charakterystyczną rzeczą dla Linuksa jest jego numerowanie. Najnowszą stabilną wersją w chwili obecnej jest jądro 2.4.21.

Pierwsze dwie cyfry opisują wersję kernela, ostatnia - jego wydanie. Dodatkowo możemy stwierdzić, czy jest to wydanie stabilne, czy testowe. Określa to druga cyfra - jeśli jest parzysta, oznacza stabilną wersję, nieparzysta - niestabilną. Czyli:

2.2.19 - wersja 2.2 (stabilna), wydanie 19-te

2.5.42 - wersja niestabilna, wydanie 42-gie

2.4.20 - wersja stabilna 2.4, wydanie 20-te

Jądro to środek systemu. Właśnie na nim użytkownik Linuksa skupia największą uwagę. Wielką rolę w pracy systemu odgrywa rekompilacja jądra. Wkompilowanie jedynie najważniejszych sterowników i opcji jądra pozwala na znaczną poprawę wydajności systemu.

Na początku Twojej przygody wystarczą Ci wyżej podane informacje. Jądro to bardzo skomplikowany element, będący jednocześnie rdzeniem każdego systemu. Z czasem, kiedy nabierzesz wprawy w korzystaniu z Linuksa, zaczniesz z pewnością interesować się jego kodem. Możliwe, że będziesz pisał własne moduły, o ile jesteś biegłym programistą C...

## Linki

W Internecie istnieje ogromna ilość adresów poświęconych tematyce Linuksa. W tym rozdziale zapoznasz się z tymi, które pomogą ci w rozwiązywaniu ewentualnych problemów. Dowiesz się także co robić, gdy nie jesteś podłączony do sieci, a chcesz zdobyć pewne informacje.

### Gdzie szukać pomocy?

Jeśli natrafiłeś na jakiś problem i nie wiesz jak sobie z nim poradzić, wiele pomocy może przynieść Ci manual systemowy. Jest to zbiór dokumentacji odnośnie konkretnego programu. np. wpisując

```
man mkdir
```

otrzymasz informacje odnośnie znanego Ci już polecenia `mkdir`.

Wiele informacji odnajdziesz także w bezpośredniej dokumentacji oprogramowania. Znajduje się ona zazwyczaj w katalogu `/usr/share/doc/nazwa_programu/`.

I choć instalując Linuksa nie trzymasz w rękę żadnej instrukcji - po instalacji masz dostępną ogromną dokumentację, wyjaśniającą działanie poszczególnych programów.

### Strony, które warto odwiedzić

Poniżej zamieściłem listę adresów stron, które ze względu na ich treść i zawartość zasługują na odwiedzinę:

[www.linux.pl](http://www.linux.pl)  
[www.newbie.linux.pl](http://www.newbie.linux.pl)  
[www.linuxnews.pl](http://www.linuxnews.pl)  
[www.linuxnewbie.prv.pl](http://www.linuxnewbie.prv.pl)  
[www.linux.gery.pl](http://www.linux.gery.pl)  
[www.linux.sky.pl](http://www.linux.sky.pl)  
[www.plug.pl](http://www.plug.pl)  
[www.kernel.org](http://www.kernel.org)  
[www.linux.com](http://www.linux.com)  
[www.linuxdoc.com](http://www.linuxdoc.com)  
[www.jtz.org.pl](http://www.jtz.org.pl)  
[www.pld.org.pl](http://www.pld.org.pl)  
[www.debian.org](http://www.debian.org)  
[www.debianusers.pl](http://www.debianusers.pl)  
[www.slackware.com](http://www.slackware.com)  
[www.slackware.com.pl](http://www.slackware.com.pl)

Oczywiście adresów jest mnóstwo, wystarczy się o tym przekonać wpisując w wyszukiwarce ciąg znaków "linux" lub "linux + pl".

### **Grupy dyskusyjne**

pl.comp.os.linux  
pl.comp.os.linux.programowanie  
pl.comp.os.linux.sprzet  
pl.comp.os.linux.sieci  
alt.pl.comp.os.linux.newbie  
alt.pl.comp.os.linux.debian

## **Oprogramowanie**

Za pewne na początku będziesz czuł się trochę zagubiony wśród oprogramowanie dostępnego dla Linuksa. Dlatego też w tym dziale przybliżę Ci informacje na temat programów które pomogą Ci sprawnie korzystać z systemu.

#### *Multimedia:*

MPlayer - odtwarzacz filmów, zawiera także koder plików multimedialnych  
Xine - odtwarzacz filmów (przede wszystkim DVD)  
XMMS - odpowiednik WinAmpa  
Grip - program służący do kompresji plików dźwiękowych

#### *Grafika:*

Gimp - program do obróbki grafiki rastrowej  
Blender - jeden z najlepszych programów do grafiki 3D  
GQView - przeglądarka plików graficznych

*Pakiety biurowe:*

OpenOffice - idealny pakiet dla użytku domowego

KOffice - pakiet biurowy KDE

AbiWord i Gnumeric - edytor tekstu i arkusz kalkulacyjny Gnome

*Edytory HTML:*

Bluefish

Screen

Quanta Plus

*Przegrywanie płyt CD:*

XCDRoast

CDBakeOven

GToaster

*Przeglądarki internetowe:*

Mozilla

Konqueror

Galeon

Netscape

Links

*Poczta i grupy dyskusyjne:*

Evolution - poczta

Kmail - poczta

Sylpheed - poczta

Mozilla - Kurier poczty

Pan - grupy dysk.

KNod - grupy dysk.

*FTP i ściąganie plików:*

gFTP

wget

*Archiwizacja:*

tar

gzip

bzip2

zip

Ark - nakładka graficzna obsługująca programy kompresujące

*Bazy danych:*

MySQL

PostgreSQL

*Serwery sieciowe:*

Apache - serwer WWW

ProFTPD - serwer FTP

PureFTPD - serwer FTP

Sendmail - serwer poczty

QMail - serwer poczty  
IRCd - serwer IRC  
INN - serwer news  
Bind - DNS

Wyżej wymieniałem tylko programy, których licencja pozwala na ich darmowe korzystanie, bez żadnych ograniczeń. Po większą listę programów oczywiście sięgnij do Twojego menedżera pakietów. Wystarczy wybrać przeszukiwanie listy i podać np. wyraz mail, a menedżer znajdzie programy służące do obsługi poczty.

## **Dodatek A**

### **Linus Torvalds - autor Linuksa**

Linus Torvalds urodził się w stolicy Finlandii - Helsinkach 28 grudnia 1968 roku. Linus to dość nietypowe i niespotykane imię. Rodzice nadali mu je na cześć znanego chemika XX wieku, dwukrotnego laureata Nagrody Nobla, Linusa Paulinga.

Rodzice Linusa byli rozwiedzeni. Z tego też powodu spędzał on wiele czasu u swojego dziadka - Leo Toernqvista, który był wykładowcą statystyki na uniwersytecie w rodzinnym mieście. Leo często pracował przy komputerze, a młody geniusz przypatrywał się pracy dziadka. Po śmierci Leo komputer trafił na biurko Linusa. Jego przygoda w świat informatyki zaczęła się na dobre.

Z biegiem czasu pogłębiał swoją wiedzę, zaczął programować. Pod koniec lat 80-tych, kiedy komputery klasy PC stały się popularne rodzice kupili Linusowi taką maszynę. Nowy komputer wciągnął go w świat gier. Dopiero kiedy kupił nowego, szybszego PC-ta zajął się na poważnie programowaniem. Udało mu się nawet przenieść zawartość starego sprzętu na nowy.

W latach 90 bardzo popularny stał się Unix. System jednak był skomplikowany, wymagał wiele ze strony użytkownika. Nic jednak w tym dziwnego... Unix napisany był dla Internetu. Powiedzieć można, że to on podniósł Internet na nogi. Fakt trudnej obsługi to nie jedyny mankament systemu - Unix był bardzo drogi, przez co niewielka liczba osób mogła bezproblemowo go nabyć.

Powstał więc Minix - system bazujący na Uniksie (nie zapominajmy że Unix miał otwarty kod). Celem jego miało być "zlikwidowanie" trudności Uniksa. Autor Miniksa rozpowszechnił go wśród swoich studentów. Wydał także książkę opisującą genezę oraz sposób działania systemu. Niestety Minix pozostawał tak jak Unix wciąż drogim systemem.

Linus przyglądając się sytuacji panującej w ówczesnym świecie systemów postanowił spróbować swoich sił i napisać własny system operacyjny. W swojej pracy wykorzystał dwa wyżej wymienione pozycje - Miniksa i Uniksa. Opierając się na Uniksie wykorzystywał kod Miniksa. I choć pracę zaczął w Assemblerze, później przepisał kod w C. W niecałe trzy lata powstała wersja 1.0 jądra.

W roku 1997 Linus wyjechał do Stanów. Mieszka w Santa Clara w Kalifornii. Jest przewodnikiem całego projektu tworzenia jądra systemu.

W swym życiu prywatnym wiele czasu spędza ze swoimi trzema córkami oraz żoną. W wolnym czasie lubi pić piwo, jeździć swoim BMW i strzelać z broni palnej.

## **Dodatek B**

### **Pingwinek w logo - skąd?**

Skąd wziął się akurat pingwin w logo Linuksa? Historia ta jest dość zawiła, a zaczęła się w roku 1996, kiedy na liście dyskusyjnej poświęconej Linuksowi ogłoszono nieoficjalny konkurs, który przyciągnął spore grono ochotników. Prace były naprawdę różne. Często padały propozycje umieszczenia w logo zwierząt. Rekiny, czy nawet orły miały symbolizować mocne strony systemu - szybkość, niezawodność czy nawet drapieżność. I choć oddawały świetnie naturę Linuksa, nie pasowały do jego autora. Wybór padł więc na niepozornego pingwinka. Swoją decyzję Linus opublikował w liście.

Torvalds kojarzy te zwierzątko z sympatyczną przytulaną. Nie jest tak silny by utrzymać Ziemię jak Atlas, ale świat dla niego to fantastyczne miejsce. Właśnie zjadł kilka ryb, a teraz siedzi i uśmiecha się. Jest zbyt najedzony by wstać.

Zaraz po wypowiedzi Linusa na liście dyskusyjnej pojawiła się informacja o nowym konkursie, tym razem na najlepszego pingwinka. Najlepsze nadesłane prace umieszczone są pod adresem [www.cs.earlham.edu/~jeremiah/linux-pix/linux-logo.html](http://www.cs.earlham.edu/~jeremiah/linux-pix/linux-logo.html).