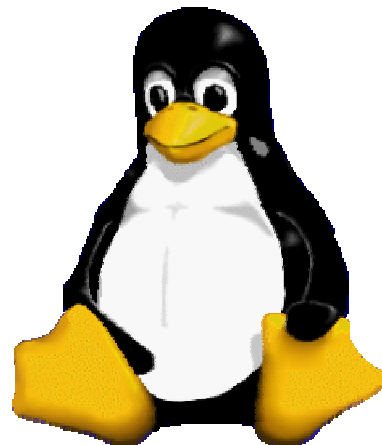


SZKOŁA LINUXA

Wprowadzenie do systemu LINUX



Opracowanie: Fabian Przedborski

Prawa do rozpowszechniania, kopiowania i udostępniania treści lub jej fragmentów posiadają autor i Szkoła Linuxa

Warszawa 2004

<u>Wersje systemu Unix.....</u>	<u>.....</u>
<u>Krótką historia systemu Linux.....</u>	<u>.....</u>
<u>Najważniejsze dystrybucje Linuxa.....</u>	<u>.....</u>
<u>RED HAT.....</u>	<u>.....</u>
<u>SLACKWARE.....</u>	<u>.....</u>
<u>SUSE.....</u>	<u>.....</u>
<u>PLD.....</u>	<u>.....</u>
<u>DEBIAN.....</u>	<u>.....</u>
<u>KNOPPIX, SLACKWARE LIVE, AUROX LIVE, WARLINUX.....</u>	<u>.....</u>
<u>Oprogramowanie Free Software a Open Source.....</u>	<u>.....</u>
<u>Możliwości Linuksa.....</u>	<u>.....</u>
<u>Instalacja systemu Linux.....</u>	<u>.....</u>
<u>Jądro systemu Linux.....</u>	<u>.....</u>
<u>Budowa systemu Linux.....</u>	<u>.....</u>
<u>Konsola, konsole wirtualne.....</u>	<u>.....</u>
<u>Środowisko X-Window.....</u>	<u>.....</u>
<u>Konfiguracja środowiska graficznego.....</u>	<u>.....</u>
<u>Przeglądarki stron internetowych.....</u>	<u>.....</u>
<u>Programy biurowe.....</u>	<u>.....</u>
<u>Programy internetowe i pocztowe.....</u>	<u>.....</u>
<u>Logowanie do systemu.....</u>	<u>.....</u>
<u>Zamykanie systemu.....</u>	<u>.....</u>
<u>Podstawowe polecenia i ich opcje.....</u>	<u>.....</u>
<u>Gdzie szukać pomocy ?.....</u>	<u>.....</u>
<u>Skróty klawiszowe w konsoli.....</u>	<u>.....</u>
<u>Midnight Commander.....</u>	<u>.....</u>
<u>Podstawy edytora vi.....</u>	<u>.....</u>
<u>Kopiowanie, przenoszenie, usuwanie plików i katalogów.....</u>	<u>.....</u>
<u>Montowanie dyskietki, CD-ROMu i innych systemów plików.....</u>	<u>.....</u>
<u>Konfiguracja /etc/fstab.....</u>	<u>.....</u>
<u>Shelle i zmienne środowiskowe.....</u>	<u>.....</u>
<u>Prawa dostępu do plików i katalogów.....</u>	<u>.....</u>
<u>FTP - File Transfer Protocol.....</u>	<u>.....</u>
<u>Aliaszy.....</u>	<u>.....</u>
<u>Dowiązania - linki.....</u>	<u>.....</u>

<u>Wyszukiwanie i sortowanie.....</u>	
<u>Archiwizacja plików i katalogów.....</u>	
<u>„Spolszczenie” Linuksa.....</u>	
<u>Przekierowania i potoki.....</u>	
<u>POTOKI.....</u>	
<u>FILTRY.....</u>	
<u>Instalacja pakietów RPM.....</u>	
<u>Instalacja programów ze źródeł.....</u>	
<u>Procesy.....</u>	
<u>Narzędzia dyskowe: fsck i fdformat.....</u>	
<u>Dodatek A: Indeks podstawowych poleceń systemu Linux.....</u>	
<u>Dodatek B: Spis poleceń informacyjnych systemu Linux.....</u>	

Wersje systemu Unix

System operacyjny Unix jest obok systemu MS-DOS i Windows jednym z najpopularniejszych obecnie systemów operacyjnych. Jest to system, który został napisany przez Kena Thompsona i Dennisa Ritchi'ego w roku 1969 w AT&T Bell Laboratories i który po dotarciu na amerykańskie uniwersytety około roku 1974 był bardzo intensywnie rozwijany na wydziałach nauk komputerowych. W chwili obecnej występuje w bardzo wielu odmianach rozwijanych głównie przez producentów sprzętu komputerowego. Istnieją także ogólnie dostępne (darmowe) wersje Unixa takie jak BSD Unix czy GNU/Linux. BSD Unix, to Unix rozwijany na Uniwersytecie Kalifornijskim w Berkeley i rozprowadzany w pakiecie o nazwie Berkeley System Distribution (w tym jego wersję na platformę 8086 o nazwie FreeBSD). Jądro systemu GNU/Linux napisał fiński student 21-letni Linus Torvalds w 1991 r. Od tamtego czasu system jest stale intensywnie rozwijany przez ogromną rzeszę entuzjastów z całego świata. Poprzez sieć Internet oraz na płytach CD-ROM rozpowszechnianych jest wiele tzw. dystrybucji Linuxa (RedHat, SuSE, Caldera, Debian, itd.), czyli zestawów oprogramowania, na które składają się system operacyjny Linux oraz programy ułatwiające instalację oraz konfigurację systemu, a także cała gama programów użytkowych i narzędziowych.

W przeciwieństwie do systemu MS-DOS system Unix (a także Linux) został pomyślany i zaprojektowany jako system wielozadaniowy, wieloużytkownikowy i interaktywny. Oznacza to, że jeden bądź wielu użytkowników może równocześnie zlecać systemowi wykonanie szeregu oddzielnych zadań. W istocie system komputerowy (wyposażony w jeden procesor) nie obsługuje zleceń wszystkich użytkowników w tej samej chwili, czyli nie wykonuje wszystkich zadań naprawdę równocześnie, jak czyni to system wieloprocesorowy, lecz tylko sprawia takie wrażenie. System Unix jest bowiem systemem z podziałem czasu, w którym czas jednostki centralnej jest kolejno równo dzielony pomiędzy poszczególne zadania. Ponieważ jednostka centralna działa bardzo szybko każdy użytkownik odnosi wrażenie, że system obsługuje tylko jego zlecenia (oczywiście przy bardzo obciążonym systemie komputerowym to wrażenie znika). Zatem w każdej chwili czasu, a ściślej w trakcie ustalonego kwantu czasu, system operacyjny zajmuje się tylko jednym procesem (wykonującym się programem) pojedynczego użytkownika. Architektura systemu oraz sprzęt, który ją wspiera zapewniają ochronę integralności zasobów użytkowników oraz oddzielnych zadań, dzięki czemu setki różnorodnych procesów może się wykonywać **równocześnie** wzajemnie sobie nie przeszkadzając.

Wybrane wersje Unixa i ich producenci:

- At&T System V Release 4
- AIX – produkcji IBM
- IRIX – Silicon Graphics
- HP-UX – produkcji Hewlett-Packard
- Solaris – produkowany przez Sun Microsystem Ltd (dawniej SunOS)
- Ultrix - Digital Equipmant Corporation
- Tru64Unix - Compaq
- SCO

Wersje darmowe:

- Berkley Unix (FreeBSD, OpenBSD) rozwijany na Uniwersytecie Kalifornijskim w Berkeley
- Linux – dystrybucje wydawane przez różnych producentów

Krótką historia systemu Linux

Początki systemu Linux datuje się na rok 1991. Sama historia systemu jest jednak nieco dłuższa. Linux powstał niejako przypadkiem, gdyż jego autor wcale nie zakładał stworzenia takiego systemu operacyjnego. Linus Torvalds - dużo wcześniej przyglądał się i porównywał dwa systemy: Unixa i Minixa. Z początku chciał jedynie poznać bliżej pracę procesora 386 w trybie pamięci chronionej na przykładnie Minixa. Dzięki kontaktom w gronie użytkowników tego drugiego, Linus znajdował odpowiedzi na wszystkie jego wątpliwości.

Z czasem zaczął modyfikować kod źródłowy Minixa. Torvalds zaczął pracę w Asemblerze, jednak później zdecydował pisać jądro w C. Jądro nowego systemu było niejako kompilacją systemów Minix i Unix, gdyż na Unixie właśnie Torvalds się wzorował.

Wersja 0.01 ukazała się w sierpniu 1991 roku. Jądro zawierało tylko proste sterowniki i to nie wszystkich urządzeń (nie obsługiwało np: stacji dyskietek. Po 2 miesiącach pojawiła się wersja 0.02, która mogła już uruchamiać proste programy. Wówczas system nie miał jeszcze dzisiejszej nazwy.

Dzięki umieszczeniu informacji o nowym systemie operacyjnym na grupie **comp.os.minix** oraz opublikowaniu kodu źródłowego jądra pod koniec roku 1991 z systemu korzystało 100 osób Samo źródło liczyło już 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. Umieszczenie źródeł kodu jądra rozpoczęło proces tworzenia i modyfikacji jądra przez programistów z całego świata

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, która zawierała w sobie słowo *free* (wolny), *freak* (dziwak) i *X* jak Unix. Ari Lemmke, który umieścił system na swoim FTP zaproponował więc nazwę LINUX – zlepek słów Linus i Unix. Nazwa została zaaprobowana przez autora kodu i używana jest do dzisiaj.

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

Obecnie w świecie Linuksa dostępnych jest ogromna ilość dystrybucji. Istnieją także mini dystrybucje - Linux na jednej bądź kilku dyskietkach. Poniżej opisane zostały najbardziej znane dystrybucje Linuksa, które bez wątplenia zasługują na szczególną uwagę.

Najważniejsze dystrybucje Linuxa

Red Hat

Red Hat jak zostało powiedziane wcześniej był jedną z pierwszych dystrybucji. Tworzony jest przez amerykańską firmę Red Hat. Dystrybucja ta jest dziś najpopularniejszą odmianą Linuxa. Ceniona przez wielu zdobyła wysokie miejsce na podium rynku systemu. O wysokiej jakości tej odmiany świadczy fakt iż wiele innych dystrybucji (Mandrake, Aurox, PLD) jest wzorowanych właśnie na niej. 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 większych problemów z jej opanowaniem. Ta dystrybucja zarówno nadaje się na serwery sieciowe jak i na komputery użytku domowego. Red Hat, podobnie jak inne dystrybucje, posiada wersję komercyjną, która zawiera dodatkowe oprogramowanie napisane przez firmę. Adres Red Hata: www.redhat.com i [ftp.redhat.com](ftp://ftp.redhat.com)

Slackware

Slackware to jedna z najstarszych dystrybucji o najniższych wymogach sprzętowych (4MB RAM, 20 MB na HD, Intel 386, FDD 5,25''). Ceniony jest przede wszystkim przez administratorów serwerów i dla tych osób jest kierowany. Patrick Volkerding i jego firma tworząca dystrybucję położyli duży nacisk na ochronę systemu. Dzięki temu Slackware to potężne narzędzie w rękach administratorów. Mankamentem wydaje się jednak mała ilość programów dołączonych z CD. Wiele osób jednak to ceni - w końcu Slackware to dystrybucja przede wszystkim na serwer. Najnowsza wersja Slackware (9.1) wydana jest na 2 płytach + 2 dodatkowych CD. System wydawany jest też w wersji uruchamialnej bezpośrednio z bootowalnej CD; bez instalacji (Slackware Live). Adres dystrybucji: www.slackware.com

SuSE

SuSE zwana popularnie „suseł” to niemiecka dystrybucja. 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

PLD

PLD to skrót od Polish Linux Distribution. To polska dystrybucja wzorowana na systemie Red Hat. Deweloperzy 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. Dystrybucja mimo iż powstała na bazie Red Hata, który można polecić nawet początkującym kierowana jest raczej do zaawansowanych użytkowników PLD cieszy się uznaniem wśród administratorów serwerów. Niektórzy mówią też, że choć bazuje na Red Hacie, to wyszedł po prostu od niego lepszy! O projekcie PLD poczytamy pod www.pld-linux.org

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. 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. Dodatkowo dystrybucja znakomicie nadaje się do użytku domowego. I choć przeznaczona jest dla zaawansowanych, docenia ją nawet osoby raczkujące w Linuxie - 8710 pakietów sprawi, że do codziennej pracy zawsze znajdzie się odpowiedni program z CD, bez konieczności pobierania pakietów z sieci. Debian znajduje się pod: www.debian.org

Knoppix, Slackware Live, Aurox Live, WarLinux

To grupa dystrybucji uruchamialnych bezpośrednio z płyty CD, bez konieczności instalowania ich na dysku twardym komputera. Idealnie sprawdzają się one w sytuacji kiedy chcemy zobaczyć jak działa Linux, bądź tylko chwilowo z niego skorzystać na danym komputerze. Systemy te w chwili uruchomienia konfiguruje się prawie same i już po uruchomieniu są gotowe do wykonania większości codziennych zadań. Większość z bootowanych dystrybucji można także instalować na dysku.

Oprogramowanie Free Software a Open Source

(Atrykuł Richarda Stallman – założyciela Free Software Foundation)

Chociaż bez względu na nazwę wolne oprogramowanie oferuje tę samą wolność, nazwa jest bardzo ważna: różne słowa *przekazują różne znaczenia*.

W roku 1998 niektórzy programiści pracujący nad wolnym oprogramowaniem, zaczęli używać terminu „oprogramowanie open source” zamiast „wolne oprogramowanie”. Termin „open source” szybko zaczął być kojarzony z odmiennym podejściem, inną filozofią, innymi wartościami, a nawet z odmiennym kryterium, według którego akceptowane są licencje. Oba ruchy, Wolne Oprogramowanie i Open Source, są obecnie osobnymi ruchami, chociaż możemy i pracujemy razem nad niektórymi praktycznymi przedsięwzięciami.

Fundamentalna różnica między tymi dwoma ruchami leży w uznawanych przez nie wartościach, sposobach patrzenia na świat. Dla Open Source kwestia, czy oprogramowanie powinno mieć dostępne otwarte źródła to problem praktyczny, nie etyczny. Jak to ktoś ujął: „Open source to metodyka konstruowania, wolne oprogramowanie to ruch społeczny”. Dla ruchu Open Source oprogramowanie, które nie jest wolne to rozwiązanie gorsze niż optymalne. Dla ruchu Wolnego Oprogramowania programy, które nie są wolne to problem społeczny, którego rozwiązaniem jest wolne oprogramowanie.

Związek między ruchem Free Software a ruchem Open Source

Oba ruchy, Free Software i Open Source, są jak dwa skrzydła polityczne w naszej wspólnocie. Grupy radykalne znane są z tego, że tworzą frakcje: organizacje dzielą się z powodu niezgodności opinii na temat detali strategicznych, a następnie nienawidzą się nawzajem. Zgadzą się co do podstawowych zasad, lecz mają inne opinie w kwestiach praktycznych zaleceń, uważają jednak drugie skrzydło za przeciwników i zwalczają się jak mogą.

W przypadku ruchu Free Software i Open Source, wszystko jest na odwrót: nie zgadzamy się co do podstawowych zasad, ale jesteśmy mniej więcej zgodni jeśli chodzi o praktyczne zalecenia. Więc możemy współpracować i współpracujemy razem nad wieloma konkretnymi projektami. Nie uważamy ruchu Open Source za wroga. Wrogiem jest oprogramowanie prawnie zastrzeżone o restrykcyjnych licencjach.

Nie jesteśmy przeciwko ruchowi Open Source, ale nie chcemy by nas wrzucano do jednego worka. Uznajemy wkład, jakie wnieśli do naszej społeczności, lecz to my ją stworzyliśmy. Chcemy, by kojarzono nasze osiągnięcia z naszymi wartościami i naszą filozofią. Chcemy być słyszani, nie przysłonięci przez grupę o odmiennych poglądach. Wkładając wiele wysiłku w to, żeby nie

uważano nas za część tej grupy, wystrzegamy się stosowania słowa „otwarte” w odniesieniu do wolnego oprogramowania lub jego przeciwieństwa, „zamknięte”, gdy mowa o niewolnym oprogramowaniu.

Dlatego mówiąc o pracy, którą wykonaliśmy i oprogramowaniu, które zbudowaliśmy, takim jak system operacyjny GNU/Linux, wspomnijcie o ruchu Wolnego Oprogramowania.

Porównanie obu terminów

Pozostała część tego artykułu stanowi porównanie obu terminów: „wolne oprogramowanie” i „open source”. Pokazano w niej, dlaczego termin „open source” nie rozwiązuje żadnych problemów – wręcz przeciwnie: tworzy nowe.

Dwuznaczność

Termin „free software” niesie ze sobą problem dwuznaczności, bowiem niezamierzone znaczenie: „oprogramowanie, które możesz dostać za darmo” [w angielskim „free” znaczy zarówno „wolny” jak i „darmowy”] pasuje równie dobrze jak zamierzone znaczenie: „oprogramowanie, które daje użytkownikowi pewną wolność”. Staraliśmy się zaradzić temu problemowi publikując dokładniejszą definicję wolnego oprogramowania, nie jest to jednak doskonale rozwiązanie – nie wyeliminuje problemu całkowicie. Niedwuznaczny, prawidłowy termin byłby lepszy, przyjmując, że nie pociągnie za sobą innych problemów.

Na nieszczęście w języku angielskim wszystkie alternatywy wiążą się z kolejnymi problemami. Zbadaliśmy wiele z tych, które nam sugerowano, ale żadna nie jest na tyle „dobra”, by zamiana miała sens. Każdy proponowany zamiennik „free software” wiąże się z jakimś problemem semantycznym lub gorszej natury – do tej ostatniej kategorii należy „oprogramowanie open source”.

Oficjalna definicja „programowania open source”, opublikowana przez Open Source Initiative, jest bardzo zbliżona do naszej definicji wolnego oprogramowania. Jednak pod pewnymi względami jest nieco mniej rygorystyczna, przez co akceptują kilka licencji uważanych przez nas za niedopuszczalnie restrykcyjne wobec użytkowników.

Jednak oczywistym znaczeniem określenia „open source software” jest „można obejrzeć kod źródłowy”. Jest to o wiele słabsze kryterium niż wolne oprogramowanie; zawiera bowiem nie tylko wolne oprogramowanie, ale i półwolne programy, takie jak Xv, a nawet niektóre programy prawnie zastrzeżone [*proprietary* znaczy dokładniej „objęty prawem własności” – przyp. a.s.], włączając w to Qt za czasów, kiedy obowiązywała poprzednia licencja (przed QPL).

Oczywiste znaczenie „open source” nie jest tym, co mają na myśli ludzie, którzy je promują. Skutek jest taki, że większość osób je źle rozumie. Pisarz Neal Stephenson w ten oto sposób zdefiniował „open source”:

Linux jest oprogramowaniem „open source”, co po prostu oznacza, że każdy może uzyskać kopie jego kodów źródłowych.

Nie sądzę, by celowo próbował odrzucić czy spierać się z „oficjalną” definicją. Zastosował po prostu reguły języka angielskiego i wywnioskował z nich znaczenie terminu. Amerykański stan Kansas opublikował podobną definicję:

Stosowanie oprogramowania open source (open-source software, OSS). OSS to programy, których kod jest swobodnie i publicznie dostępny, choć konkretne umowy licencyjne mogą różnić się co do tego, co można zrobić z tym kodem.

Oczywiście można sobie z tym radzić publikując dokładną definicję terminu. Ludzie używający terminu „open source software” uczynili to, tak jak i my zrobiliśmy z „wolnym oprogramowaniem”.

Jednak objaśnienie terminu „wolne oprogramowanie” jest proste – ktoś, kto pojął, że mamy na myśli jedno znaczenie przymiotnika zamiast drugiego („free speech, not free beer” – „wolność słowa, nie darmowe piwo”) już nie będzie rozumiał go niewłaściwie. Nie ma natomiast podobnie zwięzłego wyjaśnienia oficjalnej definicji znaczenia terminu „open source”, które jasno wskazywałoby dlaczego naturalne znaczenie jest niepoprawne.

Strach przed wolnością

Głównym argumentem za „oprogramowaniem open source” jest twierdzenie, że „wolne oprogramowanie” jest dla ludzi niewygodne. To prawda: mówienie o wolności, o kwestiach etycznych, o odpowiedzialności i wygodzie wymaga od ludzi myślenia o rzeczach, które woleliby pominąć. Mogą one zasiać niepokój i niektórzy mogą je odrzucić z tego powodu. Nie wynika z tego, że społeczeństwu będzie lepiej, jeśli nie będziemy mówić o tych rzeczach.

Kilka lat temu osoby pracujące nad wolnym oprogramowaniem zauważyły tę reakcję – uczucie niepokoju, i niektórzy zaczęli szukać sposobu uniknięcia go. Zorientowali się, że nie mówią nic o etyce i wolności, lecz jedynie o bezpośrednim, praktycznym pożytku płynącym z używania niektórych wolnych programów, będą mogli w bardziej efektywny sposób „sprzedać” je pewnym użytkownikom, szczególnie w sferach biznesowych. Termin „open source” jest tego kontynuacją: sposobem na zdobycie „większej akceptacji w kręgach biznesowych”. Poglądy i wartości ruchu Open Source mają swoje źródło właśnie w tej decyzji.

Podejście to okazało się na swój sposób efektywne. Wiele osób zaczyna dziś używać wolnego oprogramowania z powodów czysto praktycznych. Jest to oczywiście dobre, ale to nie wszystko! Przyciąganie użytkowników do wolnego

oprogramowania nie jest całą pracą, a jedynie jej pierwszą fazą.

Prędzej czy później użytkownikom tym ponownie zaproponuje się, by używali oprogramowania o restrykcyjnej licencji, z powodu takich czy innych praktycznych zalet. Niezliczone firmy próbują zaoferować tego typu pokusę – dlaczego więc użytkownicy mieliby odrzucić ich ofertę? Może tak się stać tylko wtedy, jeśli uprzednio *poznali wartość wolności* wolnego oprogramowania – sami dla siebie. Od nas zależy, czy upowszechnimy tę ideę, zaś aby to uczynić, musimy mówić o wolności. Pewna doza „musimy być cicho” w podejściu do biznesu może być korzystna dla społeczności, ale musimy też mówić wiele o wolności.

Obecnie możemy zaobserwować sporo objawów podejścia „siedzimy cicho”, a za mało wypowiedzi o wolności. Większość ludzi zaangażowanych w rozwój wolnego oprogramowania nie wspomina wiele o wolności, czego powodem jest zwykle chęć „uzyskania większej akceptacji ze strony biznesu”. Schemat ten jest szczególnie wyraźny w przypadku dystrybutorów oprogramowania. Niektóre dystrybucje systemu GNU/Linux dodają pakiety o restrykcyjnych licencjach do podstawowej, wolnej wersji systemu i przedstawiają to użytkownikom jako zaletę, a nie jako cofnięcie się na ścieżce do wolności.

Nie nadążamy za napływem nowych użytkowników wolnego oprogramowania, trudno nam jest uczyć ich o wolności i naszej wspólnoty w takim tempie, w jakim przychodzą. Dlatego też niewolne oprogramowanie (takie jak Qt, kiedy po raz pierwszy stała się popularna) i częściowo niewolne dystrybucje systemów operacyjnych znajdują tak żyzne podłoże. Rezygnacja ze słowa „wolne” w tym momencie byłaby pomyłką: potrzebujemy więcej mówić o wolności, nie mniej.

Jeśli osoby używające terminu „open source” przyciągną do naszej społeczności więcej użytkowników, będzie to pewien wkład; jeśli jednak tak się stanie, reszta będzie musiała ciężiej pracować, by zwrócić uwagę nowych osób na kwestie wolnościowe. Częściej i głośniejsze niż kiedykolwiek musimy mówić: „To wolne oprogramowanie i daje wam wolność”.

Czy znak handlowy pomoże?

Osoby propagujące „oprogramowanie open source” próbowały zastrzec znak handlowy twierdząc, że dzięki temu będą mogli zapobiec niewłaściwemu używaniu tego terminu. Ta inicjatywa została potem zarzucona, określenie było zbyt opisowe, by nadawało się na znak handlowy. Tak więc prawny status „open source” jest taki sam jak „free software”: nie ma *prawnych* ograniczeń co do jego używania. Słyszałem o wielu firmach nazywających swoje programy „open source”, mimo że nie były zgodne z oficjalną definicją – sam byłem świadkiem kilku takich wypadków.

Czy jednak używanie zastrzeżonego terminu cokolwiek by zmieniło? Nie jestem przekonany.

Firmy formułują wiadomości o nowych produktach w ten sposób, by sprawić wrażenie, że program jest „oprogramowaniem open source”, nie

mówiąc o tym bezpośrednio. Np. pewne ogłoszenie IBM dotyczące programu, którego licencja nie była zgodna z oficjalną definicją, brzmiało tak:

Jak to jest w zwyczaju w społeczności open source, użytkownicy technologii... będą mogli również współpracować z IBM...

Nie mówiło ono, że program rzeczywiście *był* „open source”, jednak wielu czytelników nie zauważyło tego szczegółu. (Należy zauważyć, że IBM szczerze próbowało uczynić ten program wolnym oprogramowaniem i później wprowadziło nową licencję, dzięki której program stał się zarówno „wolnym oprogramowaniem” jak i „open source”; jednakże w chwili ogłoszenia nie był on ani jednym, ani drugim).

A oto jak Cygnus Solutions, utworzone jako firma zajmująca się wolnym oprogramowaniem, która następnie zmieniła kurs (by tak rzec) w kierunku oprogramowania prawnie zastrzeżonego, ogłosiło wprowadzenie na rynek takiego zastrzeżonego oprogramowania:

Cygnus Solutions jest liderem na rynku open source i właśnie udostępnił dwa nowe produkty współpracujące z Linuksem.

W przeciwieństwie do IBM, Cygnus nie próbował uczynić tych programów wolnym oprogramowaniem i nie zbliżyły się one do definicji. Jednakże Cygnus nie powiedział, że są one „oprogramowaniem open source”, sformułowali za to mglistą wypowiedź próbując uzyskać sprzyjające nastawienie związane z tym terminem.

Fakty te sugerują, że znak handlowy nie zapobiegałby naprawdę zamętowi, jaki towarzyszy terminowi „open source”.

Błędne zrozumienie(?) „Open Source”

„Open Source Definition” jest wystarczająco klarowna i jasne jest, że typowe niewolne oprogramowanie nie zalicza się do tej kategorii. Wydawać by się więc mogło, że „firma open source” to taka, której produkty są wolnym oprogramowaniem, prawda? Niestety, wiele firm próbuje nadać temu inne znaczenie.

Na spotkaniu „Open Source Developers Day” w sierpniu 1998 r., wielu komercyjnych producentów oznajmiło, że planują udostępnić jako wolne oprogramowanie (lub „open source”) tylko część swojej pracy. Ich celem jest praca nad dodatkami prawnie zastrzeżonymi, (takimi jak oprogramowanie czy podręczniki), by sprzedać je użytkownikom wolnego oprogramowania. Mówią nam, byśmy uważali to za właściwe, za część naszej społeczności, ponieważ część pieniędzy jest przekazywana na rozwój wolnego oprogramowania.

W rzeczywistości firmy te próbują uzyskać sprzyjający kwit „open source” dla swoich zastrzeżonych prawnie produktów – nawet jeśli te nie są oprogramowaniem „**open source**” – ponieważ mają jakiś związek z wolnym oprogramowaniem lub utrzymują jakieś wolne programy. (Założyciel pewnej firmy powiedział całkiem otwarcie, że w każdy wolny pakiet włożą tak niewiele własnej pracy, ile tylko wytrzyma społeczność).

Na przestrzeni ostatnich lat wiele firm dołożyło się do rozwoju wolnego

oprogramowania. Część z nich zajmuje się głównie rozwijaniem oprogramowania niewolnego, jednak te dwie aktywności były oddzielone, mogliśmy więc ignorować ich zastrzeżone prawnie produkty i pracować z nimi nad wolnymi projektami. Następnie mogliśmy im uczciwie podziękować za ich wkład w rozwój wolnego oprogramowania, nie wspominając pozostałej działalności.

Nie możemy postępować podobnie w stosunku do tych nowych firm, ponieważ nam na to nie pozwolą. Firmy te usilnie próbują tak się przedstawić, by inni postrzegali ich działania jako jedną całość; chcąc, byśmy postrzegali ich prawnie zastrzeżone oprogramowanie tak samo, jak wkład w rozwój wolnego oprogramowania – którym nie jest. Przedstawiają się jako „firmy open source” mając nadzieję na mgliste ciepłe odczucie z naszej strony, które będziemy żywić w stosunku do nich.

Ta manipulacja nie byłaby mniej groźna, gdyby wykorzystywali termin „wolne oprogramowanie”. Jednakże firmy nie używają terminu „wolne oprogramowanie” w taki sposób – być może idealistyczne skojarzenia sprawiają, że wydaje im się nieodpowiedni. Termin „open source” otworzył drzwi takim praktykom.

Pod koniec roku 1998, na targach poświęconych systemowi operacyjnemu, który często nazywa się „Linuksem”, jednym z głównych mówców był dyrektor znanej firmy programistycznej. Zaproszono go prawdopodobnie dlatego, że jego firma ogłosiła, iż będzie „wspierać” ten system. Niestety ich „wsparcie” polega na sprzedaży prawnie zastrzeżonego oprogramowania współpracującego z tym systemem; innymi słowy – wykorzystuje naszą społeczność zamiast wnosić wkład w jej rozwój.

Powiedział on: „Na pewno nie udostępniemy naszego projektu na zasadach open source, ale być może zrobimy z niego „wewnętrzne” open source. Jeśli pozwolimy naszemu działowi obsługi klienta na dostęp do kodu źródłowego, jego pracownicy będą mogli naprawiać błędy dla klientów; rezultatem będzie lepszy produkt, polepszy się też jakość usług”. (Nie jest to dokładny cytat, ponieważ nie zapisałem słów – zachowuje jednak sens wypowiedzi).

Niektórzy ze zgromadzonych ludzi powiedzieli mi potem: „Ależ on nic nie rozumie”. Czyżby? Czego tak naprawdę nie zrozumiał?

Zrozumiał to, o co właśnie chodzi w ruchu Open Source. Ten ruch nie mówi o tym, że użytkownicy powinni mieć jakąś wolność, a jedynie o tym, iż pozwalając większej ilości ludzi obejrzeć kod źródłowy i pomagać go ulepszać, szybciej się go rozwinie i efekt końcowy będzie lepszy. Dyrektor bardzo dobrze to rozumiał, a nie chcąc w pełni, łącznie z zaangażowaniem w to użytkowników, zastosować tego podejścia, rozważał jego częściowe wdrożenie – wewnątrz firmy. Nie uchwycił tylko tej kwestii, dla nieporuszenia której stworzono termin „open source”: użytkownicy *zasługują* na wolność.

Rozprzestrzenianie idei wolności to wielka praca – potrzebna jest wasza pomoc. Projekt GNU pozostanie przy terminie „wolne oprogramowanie”. Jeśli

uważacie, że wolność i wspólnota są same z siebie ważne, nie tylko ze względu na wygodę, którą ze sobą niosą, proszę dołączcie do nas, używając terminu „wolne oprogramowanie”.

Autor: Richard Stallman

Możliwości Linuksa

Wielozadaniowość (można uruchamiać wiele programów naraz),

Obsługa wielu użytkowników na tej samej maszynie w tym samym czasie (bez licencji wielostanowiskowych!),

Wieloplatfomowość (Linux działa pod wieloma platformami, nie tylko pod Intelowską),

Wieloprocessorowość (obsługa SMP (wieloprzetwarzania symetrycznego) na platformach Intel i SPARC (prace nad wsparciem dla kolejnych platform trwają); Linux używany jest w wymagających wieloprzetwarzania zastosowaniach, np. w systemach Beowulf i opartym na SPARC superkomputerze Fujitsu AP1000+),

Wielowątkowość (jądro obsługuje wiele niezależnych wątków w pojedynczej przestrzeni przydzielonej procesowi pamięci),

Działanie w trybie chronionym 386,

Międzyprocesowa ochrona pamięci (dzięki czemu niemożliwa jest sytuacja, w której jeden program powoduje pad całego systemu),

Ładowanie kodu wykonywalnego na żądanie (tzn. czytanie tylko tej części programu, która jest rzeczywiście wykorzystywana),

Współdzielenie pamięci między procesami (tzn. że wiele programów może korzystać z tego samego obszaru pamięci. Gdy dany program zapisuje coś w zapełnionej już stronie pamięci, zawartość tej strony (4 KB) kopiowana jest przedtem gdzie indziej. Proces ten nazywa się copy-on-write i ma dwie zalety: zwiększa szybkość i zmniejsza zużycie pamięci),

Pamięć wirtualna stronicująca (a nie składająca całe procesy) na dysk (na osobną partycję, do pliku, lub i tu i tu, z możliwością dynamicznego dodawania kolejnych obszarów wymiany (tak się to nadal nazywa). Jednocześnie w użyciu może być ogółem 16 128-megabajtowych (a w nowszych jądrach dwugigabajtowych) obszarów wymiany; w ten sposób teoretycznie można utworzyć funkcjonalny obszar wymiany do wielkości 2GB. Zwiększenie tego limitu jest proste -- sprowadza się do zmiany kilku linijek w kodzie źródłowym),

Zgrupowanie puli pamięci przydzielonej programom i dyskowej pamięci podręcznej (cache); dzięki czemu cała wolna pamięć może być użyta jako pamięć podręczna. Ilość pamięci podręcznej zmniejsza się podczas wykonywania dużych programów,

Dynamicznie zlinkowane biblioteki dzielone (DLL) oraz, oczywiście, biblioteki zlinkowane statycznie

Zrzuty pamięci (core dumps) umożliwiające „pośmiertną” analizę programów za pomocą debuggera,

Wysoka zgodność z POSIX, SYSV, BSD na poziomie źródeł,

Poprzez zgodny z iBCS2 moduł emulacji wysoka zgodność na poziomie binarnym z SCO, SVR3 i SVR4,

Dostępność całego kodu źródłowego, w tym całego jądra i wszystkich sterowników, narzędzi i programów; wszystko to wolno bez ograniczeń rozpowszechniać. Wiele komercyjnych programów dostępnych dla Linuksa dostarczanych jest bez kodu źródłowego, ale wszystko, co było wolnodostępne (w tym cały system operacyjny) wolnodostępne pozostało,

Zgodna z POSIX kontrola procesów,

Pseudoterminale (ptty),

Emulacja numerycznego koprocesora 387 w jądrze, co odciąża programy z konieczności wykonywania tej emulacji. W ten sposób wszystkie komputery pracujące pod kontrolą systemu Linux „mają” koprocesor. Oczywiście jeśli komputer naprawdę posiada koprocesor, to będzie on użyty; w takim przypadku można skompilować jądro wyłączając obsługę 387: zaowocuje to niewielkim zyskiem pamięci.

Wsparcie dla wielu narodowych i zindywidualizowanych klawiatur, łatwo dodawać nowe definicje klawiatury dynamicznie,

Do 64 dynamicznie tworzonych wirtualnych konsol, co oznacza wiele niezależnych sesji obsługiwanych jednocześnie przy tej samej fizycznej konsoli. Między sesjami można przełączać się za pomocą dowolnie zdefiniowanej kombinacji klawiszy (niezależnej od karty graficznej),

Wsparcie dla powszechnie używanych systemów plików, takich jak np. minix, Henix, wszystkie systemy V. Własny zaawansowany system plików (do 4 TB, nazwy plików do 255 znaków),

Bezproblemowy dostęp do partycji MS-DOS i OS/2. Dzięki specjalnemu systemowi plików użycie partycji MS-DOS nie wymaga żadnych dodatkowych sztuczek, dostęp doń nie różni się oprócz ograniczeń w

nazwach i prawach plików od dostępu do normalnego uniksowego systemu plików. Aktualnie nie są obsługiwane kompresowane partycje MS-DOS 6 (ale jest na to łąta). System VFAT (tj. Windows 9x/NT) obsługiwany jest od jądra 2.0 wzwyż.

Specjalny system plików (UMSDOS) umożliwiający instalację Linuksa na partycji DOS,

Wsparcie (tylko odczyt) systemu plików HPFS-2 z OS/2 2.1,

Obsługa HFS (systemu plików Macintosha) jako moduł,

Możliwość czytania wszystkich standardowych formatów CD-ROM-ów,

Wsparcie protokołu sieciowego TCP/IP, w tym ftp, telnet, NFS, itd.,

Serwer Appletalk,

Klient i serwer Netware,

Klient i serwer Lan Manager/Windows Native (SMB),

Wiele protokołów sieciowych: podstawowe protokoły dostępne w najnowszych rozwojowych wersjach jądra to: TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP (Appletalk), Netrom i inne. Stabilne wersje jądra zawierają m.in. protokoły TCP, IPv4, Ipv6, IPX, DDP i AX.25.

Instalacja systemu Linux

1. Zebranie informacji o sprzęcie
2. Podział dysku na partycje
3. Wybór nośnika
4. Wybór pakietów do zainstalowania
5. Konfiguracja użytkowników, sprzętu, sieci
6. Konfiguracja bootloadera
7. Przygotowanie dyskietki startowej

1. Zebranie informacji o sprzęcie

Należy zebrać info na temat procesora, karty graficznej, sieciowej, muzycznej, monitora, myszki, przerwań, typów dysków twardych (kontrolerów ATA/SATA, geometrii dysków). Uwaga !!! Niektóre starsze BIOS-y nie obsługują partycji bootującej powyżej 1024 cylindra, co jest ważne w przypadku kiedy instalujemy kilka systemów operacyjnych.

2. Podział dysku na partycje

Do poprawnego działania systemu Linux wystarcza tylko 2 partycje: native (czyli root) i partycja wymiany (swap), przy czym nie ma znaczenia jaki system plików wybierzemy (ext2,ext3,jfs). W systemach unixowych (jak FreeBSD) wystarczy jedna partycja, którą następnie dzielona jest na kilka mniejszych „plastrów” ("*slice*" - plaster) z plikiem wymiany. Do optymalnego działania Linuksa należy jednak podzielić dysk na większą ilość partycji. Do podziału można użyć różnych programów partycjonujących dysk jak.:

- fdisk – DOS-owy i Unixowy
- cfdisk, Disk Druid
- Ranish Partition Manager, fips – również DOS-owe
- Partition Magic – działający w środowisku Ms Windows

(Fips i Partition Magic mogą dzielić istniejącą partycję bez utraty danych)

```

    Device Boot          Start      End          Blocks      Id System
/dev/hda1             1          547          4393746     7  HPFS/NTFS
/dev/hda2             *          548          1520          7815622+    a5  FreeBSD
/dev/hda3             1521       7926          51456195     5  Extended
/dev/hda5             1521       2128          4883728+     7  HPFS/NTFS
/dev/hda6             2129       2979          6835626     7  HPFS/NTFS
/dev/hda7             2980       4681          13671283+    7  HPFS/NTFS
/dev/hda8             4682       5127          3582463+    83  Linux
/dev/hda9             5128       5637          4096543+    83  Linux
/dev/hda10            5638       5662           200781     83  Linux
/dev/hda11            5697       5747          409626     83  Linux
/dev/hda12            5748       5843          771088+     83  Linux
/dev/hda13            5844       5877          273073+     82  Linux swap
/dev/hda14            5878       5901          192748+     83  Linux
/dev/hda15            5902       5992          730926     83  Linux
/dev/hda16            5993       6418          3421813+    83  Linux
/dev/hda17            6419       6965          4393746     83  Linux
/dev/hda18            6966       7014          393561     83  Linux
/dev/hda19            7015       7026           96358+     83  Linux
/dev/hda20            7027       7148          979933+     83  Linux
/dev/hda21            7149       7391          1951866     83  Linux
/dev/hda22            7392       7695          2441848+    83  Linux
/dev/hda23            7696       7877          1461883+    83  Linux
/dev/hda24            7878       7926          393561     83  Linux
Command (m for help): █

```

Rys. Fdisk uruchomiony podczas instalacji systemu Red Hat Linux

```

          cfdisk 2.12

          Disk Drive: /dev/hda
          Size: 80026361856 bytes, 80.0 GB
          Heads: 255   Sectors per Track: 63   Cylinders: 9729

-----
Name      Flags      Part Type FS Type      [Label]      Size (MB)
-----
hda1             Primary NTFS              4499,23
hda2             Boot   Primary FreeBSD    8003,20
hda5             Logical NTFS              5000,98
hda6             Logical NTFS              6999,72
hda7             Logical NTFS              13999,43
hda8             Logical Linux ext2      3668,48
hda9             Logical Linux ext2      4194,90
hda10            Logical Linux ext2      205,64
hda11            Logical Free Space    279,66
hda12            Logical Linux ext2      419,49
hda13            Logical Linux ext2      789,63
hda14            Logical Linux swap    279,66
hda15            Logical Linux ext2      197,41
hda15            Logical Linux ext2      748,51

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ]   [ Type ]  [ Units ] [ Write ]

          Illegal command
          Toggle bootable flag of the current partition █

```

Rys. Cfdisk podczas instalacji Slackware Linux

Przykładowy podział dysku o wielkości 10 GB:

/	700 MB
/boot	100 MB
/home	2000 MB
/usr	4000 MB
/usr/local	2500 MB
/var	300 MB
SWAP	256 MB

Ilość miejsca przyznawana poszczególnym partycjom (oprócz swap) zależy od planowanego wykorzystania systemu jako serwera. Np. konfigurując serwer

SAMBA możliwe dużo miejsca należy przeznaczyć na /home, a w przypadku routera lub firewalla można utworzyć dodatkowa partycje /var/log na logi systemowe. Rozmiar partycji swap z reguły jest dwukrotnie wiekszy od zainstalowanej w komputerze pamięci RAM, jednak tak naprawdę system nie wykorzystuje więcej niż 256 MB nawet kiedy mamy 1GB RAM-u, więc nie ma potrzeby tworzyć większego niż w/w.

3. Wybór nośnika

Instalacje Linuxa lub Unixa można przeprowadzić zawsze z wielu źródeł:

- CDROM (najczęściej wybierany sposób)
- dyskietki startowe
- zamontowana partycja DOS-owa lub ext2/ext3
- serwer FTP
- serwer NFS (wybierany podczas instalacji w dużych sieciach)

4. Wybór pakietów do zainstalowania

Większość dystrybucji pozwala wybrać już na wstępie rodzaj instalacji który można łatwo dostosować do własnych doświadczeń z Linuxem. Nawet zupełnie początkujący użytkownik znajdzie tutaj odpowiedni dla siebie sposób np.: tryb „newbie” – w Slackware, czy „automagiczny Czarodziej” w PLD, Można wybrać instalację całych grup pakietów. Dla użytkownikoo zaawansowanych zawsze jest tryb instalacji eksperta gdzie można wybrać dowolne pakiety. Wybór pakietów zależny jest od przyszłego zastosowania serwera. Obowiązkowe sa pakiety takie jak *gcc*, *glibc*, pakiety obsługujące różne funkcje sieciowe (np.: *iptables*, lub *ipchains* w przypadku firewalli). Można wybrać opcje zainstalowania wszystkiego ale wymaga to poinstalacyjnego usuwania niepotrzebnych usług oraz wymaga więcej miejsca na dysku (około 3 GB)

5. Konfiguracja użytkowników, sprzętu, sieci

Na tym etapie instalator poprosi o podanie hasła administratora (roota) i opcjonalnie utworzenie użytkowników. Należy stworzyć przynajmniej jednego normalnego użytkownika - aby nie wszystkie operacje wykonywać z konta superużytkownika. Tworzenie użytkowników przez system polega na utworzeniu odpowiedniego katalogu w /home oraz skonfigurowanie ustawień środowiska pracy dla każdego z użytkowników przez utworzenie w jego katalogu domowym odpowiednich plików konfiguracyjnych oraz wpisu w /etc/passwd i przydzielenie użytkownika do właściwej grupy.

Po tym może nastąpić autodetekcja sprzętu i wstępna konfiguracja karty graficznej, muzycznej, środowiska graficznego. Etap ten w większości dystrybucji można pominąć i dokonać konfiguracji po właściwej instalacji. Na tym etapie z reguły instalator może pomóc skonfigurować ustawienia sieciowe.

Należy wskazać więc sposób połączenia (modem, karta sieciowa), sposób pozyskania IP (statycznie wpisane, z DHCP), serwer DNS. Programy instalacyjne niektórych dystrybucji (jak Slackware Linux) w tym miejscu próbują wykryć rodzaj karty sieciowej i przypisać jej właściwy sterownik. Również ustawienia sieci można pominąć na tym etapie i skonfigurować je po zakończeniu instalacji.

6. Konfiguracja bootloadera

Na tym etapie instalator zapyta o konfigurację bootloadera. W zależności od systemu jest kilka możliwości. Wspólną cechą wszystkich dystrybucji jest możliwość nie instalowania programu rozruchowego co ma sens kiedy jest już zainstalowany inny system z własnym bootloaderem do którego zostanie dodany aktualnie instalowany system. W przeciwnym wypadku należy skonfigurować program rozruchowy umieszczając najczęściej bo w MBR. Dystrybucje oparte na Red Hacie zazwyczaj dają do wyboru dwa tego typu programy LILO i GRUB.

7. Przygotowanie dyskietki startowej

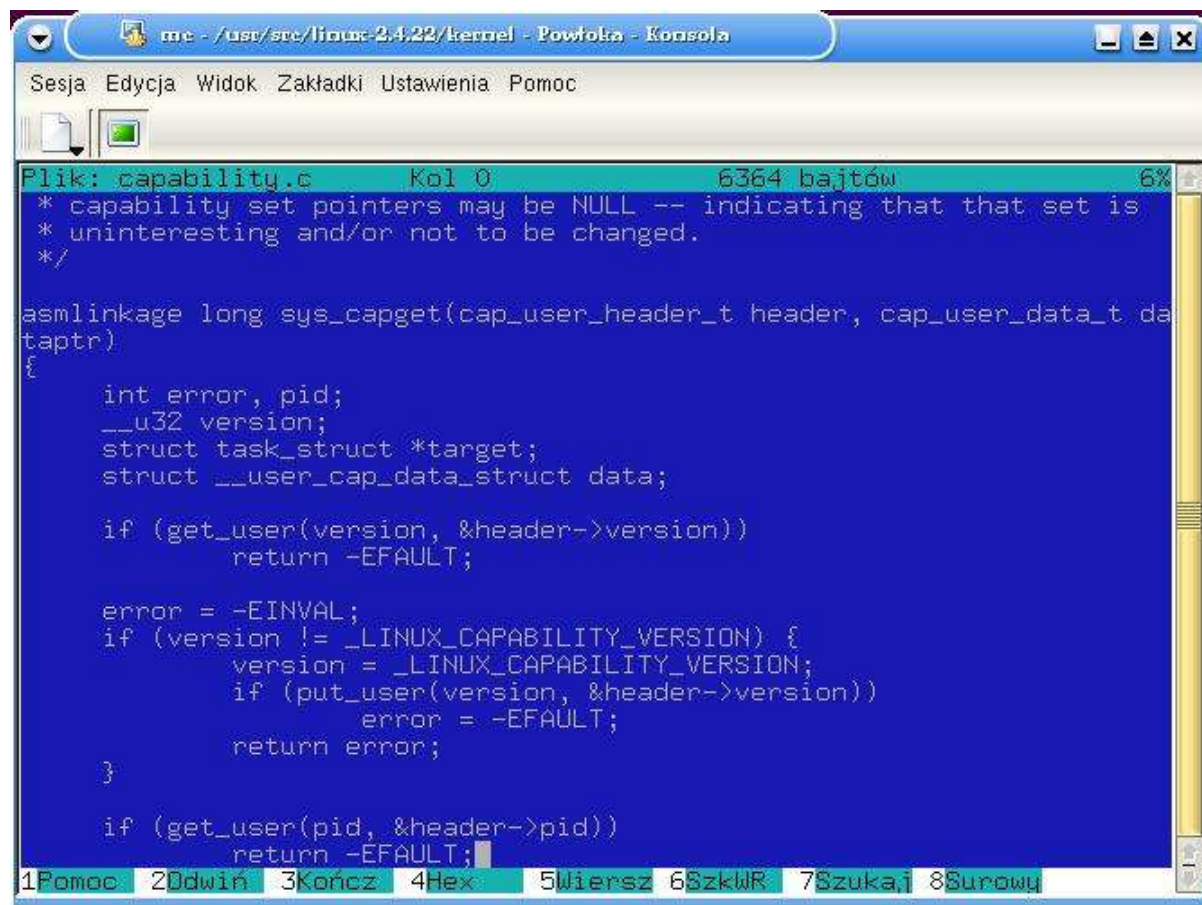
Z reguły przed zakończeniem instalacji program instalacyjny daje możliwość wykonania dyskietki startowej. Należy skorzystać z tego dobrodziejstwa gdyż czasem dyskietka startowa bywa jedynym sposobem na uruchomienie świeżo zainstalowanego systemu lub szybkie odtworzenie MBR-u po instalacji dodatkowego systemu (jak Windows) nadpisującego te sektory dysku .

Po zakończeniu instalacji należy wyjąć wszelki dyski wymienne i zrestartować świeżo zainstalowany system. Jeżeli przestawiony był poprzednio start z CD-ROM-u w BIOS-ie należy przywrócić bootowanie z dysku twardego.

Powyższe informacje nie odnoszą się do żadnej konkretnej dystrybucji – mają charakter czynności ogólnych które wykonują się niezależnie od instalowanej dystrybucji.

Jądro systemu Linux

Jądro (kernel) systemu to najważniejszy jego element. To kernel odpowiada za bezpośrednią komunikację ze sprzętem, przydział pamięci, uruchamianie i kończenie różnych procesów systemowych, operowanie sterownikami do urządzeń, wykonywanie poleceń użytkownika przesłanych do niego przez interpreter tych poleceń czyli **shell**. Właściwy Linux to tylko jądro systemu, skompilowany kod źródłowy napisany w ok. 90 % w języku C. Jądro jest więc rdzeniem systemu operacyjnego. Kernel wraz z dodatkowym oprogramowaniem takim jak powłoki użytkownika (shelle), kompilatory języków, biblioteki, środowiska graficzne i różne programy użytkowe tworzy kompletna dystrybucję.



The image shows a terminal window titled 'mc - /usr/src/linux-2.4.22/kernel - Powłoka - Konsola'. The window contains a text editor showing a snippet of C code for the `sys_capget` function. The code is as follows:

```
Plik: capability.c      Kol 0      6364 bajtów      6%
* capability set pointers may be NULL -- indicating that that set is
* uninteresting and/or not to be changed.
*/

asmlinkage long sys_capget(cap_user_header_t header, cap_user_data_t data,
taptr)
{
    int error, pid;
    __u32 version;
    struct task_struct *target;
    struct __user_cap_data_struct data;

    if (get_user(version, &header->version))
        return -EFAULT;

    error = -EINVAL;
    if (version != _LINUX_CAPABILITY_VERSION) {
        version = _LINUX_CAPABILITY_VERSION;
        if (put_user(version, &header->version))
            error = -EFAULT;
        return error;
    }

    if (get_user(pid, &header->pid))
        return -EFAULT;
}
```

At the bottom of the terminal, there is a menu with options: 1Pomoc, 2Dwini, 3Koncz, 4Hex, 5Wiersz, 6SzukwR, 7Szuka, 8Surowy.

Rys. Fragment kodu źródłowego jądra Linuksa napisanego w większości w języku C.

Jądro Linuksa ma charakter monolityczny. Oznacza to, że jest to całość zbudowana, z mniejszych dobrze przemyślanych komponentów. Linux jest kompilowany i łączony (linkowany) dynamicznie. Oznacza to, że 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**.

Komputera może pracować w dwóch trybach:

- tryb użytkownika (user mode)
- 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

[aplikacje]
TRYB UŻYTKOWNIKA

TRYB JĄDRA

[sprzęt]

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

Charakterystyczną rzeczą dla Linuksa jest jego numerowanie. Najnowszą stabilną wersją w chwili obecnej jest jądro 2.4.25. Pierwsze dwie cyfry opisują wersję kernela, ostatnia - jego wydanie. Dodatkowo można łatwo stwierdzić, czy jest to wydanie stabilne, czy testowe. Określa to druga cyfra - jeśli jest parzysta, oznacza stabilną wersję, nieparzysta - niestabilną. Np.:

- 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

Wersje stabilne pojawiają się raz na 2-3 miesiące, testowe częściej. Wersje rozwojowe po dokładnym przetestowaniu i poprawieniu błędów wypuszczane są jako wersje stabilne i umieszczane najczęściej na serwerach FTP z anonimowym dostępem (np.: ftp.kernel.org).

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.

Budowa systemu Linux

Rys.1 Struktura katalogów systemu Linux

Struktura katalogów systemu Linux i Unix ma postać drzewa. Nadrzędnym katalogiem jest / (root). Nie jest to jednak tożsame z katalogiem domowym administratora systemu (roota). Administrator ma swój katalog - /root. Nazwy wszystkich katalogów głównych (przedstawione powyżej) rozpoczynają się od „/” (/root, /home, /boot ... etc).

Zawartość poszczególnych katalogów:

- /dev** pliki urządzeń. W systemach uniksowych każde urządzenie czy napęd jest reprezentowane w postaci pliku. Te pliki przechowywane są w katalogu /dev.
- /bin** pliki wykonywalne poleceń wydawanych przez użytkownika w konsoli i narzędzia systemowe. Każde polecenie jest w rzeczywistości programem napisanym najczęściej w języku C.
- /lib** biblioteki systemowe.
- /mnt** to katalog w którym użytkownik tymczasowo montuje inne systemy plików (CD-ROM, dyskietka, partycje FAT). Po zamontowaniu zawartość ich będzie widoczna w katalogu /mnt.
- /usr** w tym katalogu system instaluje programy i aplikacje. Podkatalog /usr/local – jest najczęściej używany do instalacji dodatkowego oprogramowania przez użytkowników.
- /proc** to katalog gdzie przechowywane są podczas pracy systemu informacje o działających w systemie procesach.
- /root** katalog domowy administratora systemu.

/boot	jądro systemu i pliki niezbędne do uruchomienia systemu.
/etc	pliki konfiguracyjne całego systemu.
/home	katalog przechowujący katalogi domowe użytkowników. Po założeniu konta nowego użytkownika automatycznie tworzony jest dla niego katalog w /home o nazwie identycznej z loginem
/tmp	pliki tymczasowe, w tym internetowe
/var	różne pliki i logi systemowe. Tutaj przechowywana jest poczta userów (var/spool/mail) i kolejki wydruku

Pliki w Linuksie, inaczej niż w Windows, nie muszą mieć rozszerzeń. Mogą mieć natomiast różne atrybuty takie jak „read-only” czy „hidden”. Pliki ukryte z reguły zaczynają się od „.” Na początku nazwy (np.: `.profile`). Oprócz tego występują też pliki specjalne („*special files*”, które w rzeczywistość nie są plikami ale urządzeniami obsługującymi terminal, dyski itd. Każdy z tych plików można dowolnie przetwarzać za pomocą poleceń.

Dyski w Linuksie nazywane są inaczej niż w DOS-ie czy Windows 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.

<code>/dev/hda</code>	- dysk na pierwszym kanale IDE w pozycji Master
<code>/dev/hdb</code>	- dysk na pierwszym kanale IDE w pozycji Slave
<code>/dev/hdc</code>	- dysk na drugim kanale IDE w pozycji Master
<code>/dev/hdd</code>	- dysk na drugim kanale IDE w pozycji Slave
<code>/dev/hde</code>	- dysk na pierwszym kanale ATA w pozycji Master
<code>/dev/hdf</code>	- dysk na pierwszym kanale ATA w pozycji Slave
<code>/dev/hdg</code>	- dysk na drugim kanale ATA w pozycji Master
<code>/dev/hdh</code>	- dysk na drugim kanale ATA w pozycji Slave
<code>/dev/sda</code>	- dysk na pierwszym kanale SCSI

itd...

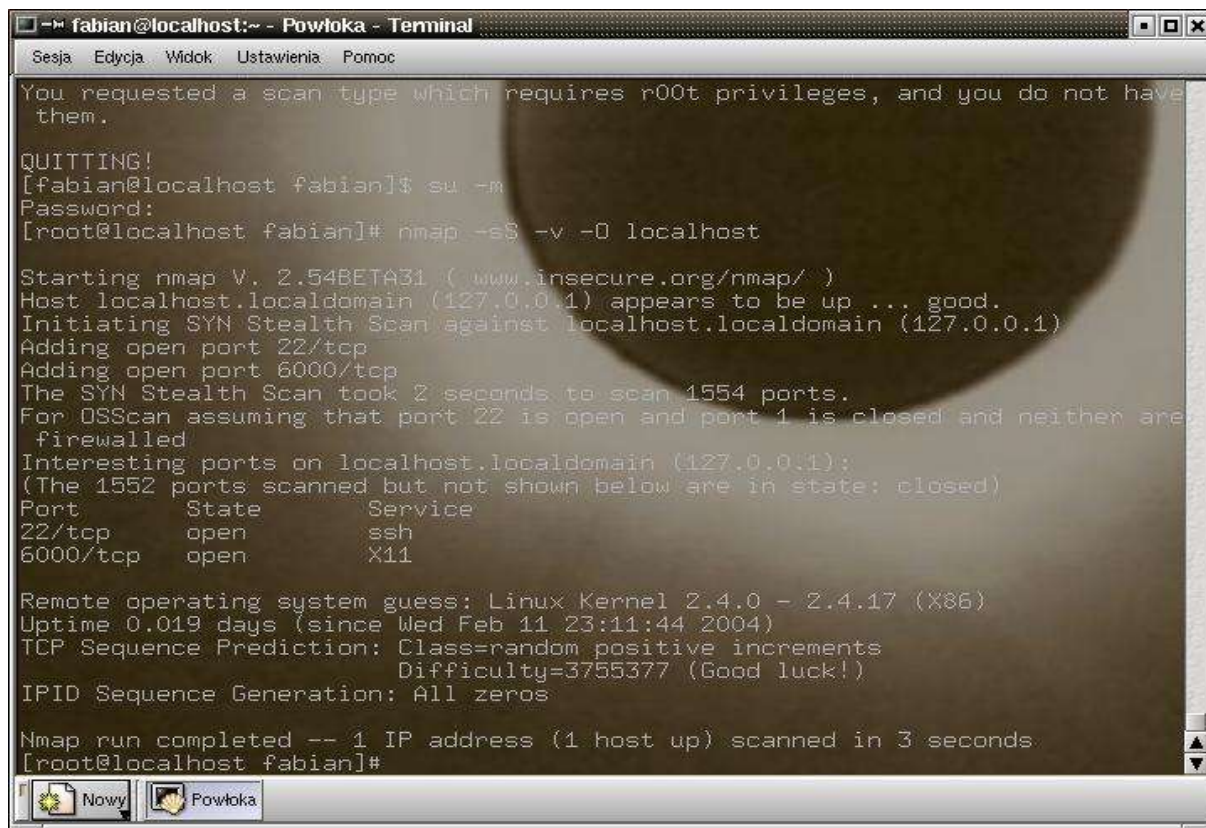
Zasada ta dotyczy również partycji. I tak: główna partycja na dysku pierwszym będzie nosiła nazwę 'hda1', a na dysku drugim - 'hdb1'. Stacje dyskietek to:

<code>/dev/fd0</code>	- pierwszy napęd dyskietek (czyli DOS-owe A:)
<code>/dev/fd1</code>	- drugi napęd dyskietek (czyli DOS-owe B:)

Konsola, konsole wirtualne

Linux domyślnie posiada siedem wirtualnych konsol, do których mamy dostęp za pomocą kombinacji klawiszy : ALT+F1-F7. Podczas uruchamiania systemu X Window ręcznie lub automatycznie, zablokowana zostaje konsola, z której zostały uruchomione X-y, a nasz menadżer wyświetlania dostępny jest na konsoli 7. Przejście z poziomu X-ów na pozostałe konsole odbywa się za pomocą następującej kombinacji klawiszy : CTRL+ALT+F1-F6

Na każdej z wirtualnych konsol możemy zalogować się, jako inny użytkownik. Takie rozwiązanie umożliwia nam uruchomienie na każdej z konsol jakiegoś zadania. Należy jednak pamiętać, że Linux to prawdziwy system wielozadaniowy, który umożliwia przeniesienie dowolnego procesu w tło, zwalniając terminal i ponowne przywrócenie go w dowolnym momencie. Poza tym Linux każdemu zadaniu (procesowi) przydziela zasób pamięci, w obrębie której może on działać. Gdyby program się „zawiesił” nie odbija się to negatywnie na innych uruchomionych programach. Co więcej można zalogować się na innej konsoli i „zabić” (o procesach więcej w dalszej części podręcznika) ów proces, zwalniając zajmowane przez niego miejsce w pamięci RAM.



```
fabian@localhost:~ - Powłoka - Terminal
Sesja  Edycja  Widok  Ustawienia  Pomoc
You requested a scan type which requires r00t privileges, and you do not have
them.
QUITTING!
[fabian@localhost fabian]$ su -m
Password:
[root@localhost fabian]# nmap -sS -v -O localhost

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Host localhost.localdomain (127.0.0.1) appears to be up ... good.
Initiating SYN Stealth Scan against localhost.localdomain (127.0.0.1)
Adding open port 22/tcp
Adding open port 6000/tcp
The SYN Stealth Scan took 2 seconds to scan 1554 ports.
For OSScan assuming that port 22 is open and port 1 is closed and neither are
firewalled
Interesting ports on localhost.localdomain (127.0.0.1):
(The 1552 ports scanned but not shown below are in state: closed)
Port      State      Service
22/tcp    open       ssh
6000/tcp   open       X11

Remote operating system guess: Linux Kernel 2.4.0 - 2.4.17 (X86)
Uptime 0.019 days (since Wed Feb 11 23:11:44 2004)
TCP Sequence Prediction: Class=random positive increments
                               Difficulty=3755377 (Good luck!)
IPID Sequence Generation: All zeros

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
[root@localhost fabian]#
```

Rys. Emulator konsoli **xterm** uruchomiony w KDE

W każdym środowisku graficznym jest też dostępny emulator konsoli, który

pozwała na wpisywanie poleceń tekstowych w taki sposób jak podczas normalnej pracy z Linuksem w trybie tekstowym. Najpopularniejszym programem tego typu jest **xterm**, który podobnie jak inne programy tego typu oferuje użytkownikowi wszelki dobrodziejstwa pracy z konsolą tekstową, jak edycja wiersza poleceń, historia wydawanych poleceń czy skróty klawiszowe.

Środowisko X-Window

X Window System jest to graficzny system dostępny w Linuksie i innych systemach Unixowych. Środowisko z wyglądu, zbliżone do tego jakie znamy z systemu MS Windows. Jedną z głównych różnic między XWindow System a systemem Windows jest to że XWindow jest zaprojektowany tak że działa niezależnie od typu systemu w architekturze klient-serwer.

Mamy więc serwer X-ów oraz program klient. Serwer odpowiedzialny jest za współpracę ze sprzętem tzn. kartą graficzną i monitorem a klient to różne programy które wyświetlają swoje dane poprzez ten serwer. Standardowy serwer XWindow pochodzi z projektu XFree86 i jest najczęściej dostarczany z popularnymi dystrybucjami. W skład jego wchodzi kilkanaście X-serwerów do obsługi najpopularniejszych kart graficznych. System Xwindow uruchamiany jest komendą "**startx**" lub automatycznie za pomocą programów takich jak xdm, kdm czy gdm. Za wygląd środowiska pracy odpowiadają klienci zwani zarządcami okien (window menedżerami). Po wpisaniu przez nas komendy **startx** zostaje uruchomiony zdefiniowany wcześniej przez nas menedżer okien. Najpopularniejsze obecnie menedżery a właściwie desktopty to:

KDE

Gnome

Enlightenment

Fvwm2

WindowMaker

System XWindow i menedżery okien są to składniki zapewniające graficzny interfejs i komfortową pracę z Linuksem. Dzięki środowiskom graficznym praca w Linuksie może być łatwa nawet dla osób początkujących.

Rys.2 Schemat graficznego interfejsu użytkownika

Rys. 3. Uruchomione środowisko KDE

KDE

KDE - K Desktop Environment jest obecnie bardzo zaawansowanym zbiorem narzędzi, który rozwijany przez lata stał się wysoką pozycją. Środowisko posiada własny menedżer plików - Konqueror. KDE posiada Centrum Sterowania - program, w którym dokonać można zmian, począwszy do ustawień graficznych, kończąc na siecia nawet konfigurować serwery. Informacje odnośnie KDE znajdują się oczywiście na oficjalnej stronie - www.kde.org.

GNOME

Gnome jest drugim najbardziej rozwiniętym środowiskiem pracy. Wraz z KDE jest najbardziej rozwiniętym środowiskiem okienkowym. Gnome korzysta z przenośnej biblioteki GTK+. Jako menedżera plików Gnome używa SawFish oraz świetnego Nautilusa. Środowisko posiada własne Centrum Sterowania Gnome. Informacje odnośnie środowiska znajdziemy pod oficjalnym adresem - www.gnome.org.

Konfiguracja środowiska graficznego

W systemie Linux mamy możliwość pracy w trybie graficznym podobnie jak

w systemach MS Windows, dlatego też najczęściej oczywiście z tego też korzystamy wynosząc te przyzwyczajenie często z systemu windows. Dlaczego by właściwie nie korzystać z możliwości jaką nam daje nowoczesny system operacyjny - wbrew szamanom tego systemu preferującym tryb tekstowy. W pierwszej kolejności oczywiście musimy sprawdzić czy mamy zainstalowane odpowiednie oprogramowanie.

Najłatwiej jest przeprowadzić autokonfigurację X-ów przez wydanie polecenia:

```
XFree86 -configure
```

Jeżeli wszystko pójdzie dobrze to konfiguracja zapisze się w /
etc/X11/XF86Config i możliwe będzie już używanie środowiska okienkowego

Standardową konfigurację okienek przeprowadzamy w zależności od posiadanej dystrybucji programami **Xconfigurator** w Red Hacie lub **xf86config**, który dostępny jest w większości dystrybucji Linuksa a nawet w niektórych Uniksach (jak FreeBSD) Program działa w trybie tekstowym; kropk po kroku wybieramy spośród dostępnej listy nasz sprzęt jaki mamy zainstalowany w komputerze oraz jego parametry. Będą potrzebne głównie takie parametry jak: Rodzaj karty graficznej, mysz, dwu lub trzy przyciskowa oraz jakiego producenta, monitor i jego dokładne parametry,

W niektórych systemach znajdziemy osobne programy do obsługi myszy jak np. w Red Hat - "**mouseconfig**".

Po ukończeniu i zapisaniu konfiguracji najczęściej w pliku /
etc/X11/XF86config, spróbujemy uruchomić system w trybie graficznym. Sposobem na uruchomienie tego trybu jest wpisanie polecenia "**startx**". Jeśli zobaczymy na ekranie jakieś niepokojące symptomy, jak dziwne śmieci lub migotanie itp natychmiast zatrzymujemy serwer wciskając klawisze **CTRL + ALT + Backspace**. Następnie ponownie uruchamiamy program konfiguracyjny i szukamy w którym miejscu popełniliśmy błąd. Czasami po takiej konfiguracji następuje zwis komputera, więc nie pozostaje nam nic innego jak restart i ponowna konfiguracja i tak do skutku. Jest też możliwe że system nie rozpoznając naszej karty uruchomi się nam w rozdzielczości 320x200 też nic ciekawego, nawet nie ma jak się rozglądać po systemie nie mówiąc już o normalnej pracy. Więc także restart i wszystko od nowa. Po prawidłowej konfiguracji i wpisaniu "**startx**" powinniśmy zobaczyć szary ekran w siatkę i po chwili powinien wystartować domyślny menedżer okien .

W większości dystrybucji można z poziomu okienek specjalnymi programami zdefiniować automatyczny start XWindow i ulubionego menedżera KDE lub Gnome.

Przeglądarki stron internetowych

Większość dystrybucji systemu Linux zaraz po zainstalowaniu daje użytkownikowi gotowe narzędzia internetowe jak przeglądarki stron www, programy pocztowe, komunikatory. Programów tego typu jest całe mnóstwo i nie sposób opisać opniżej wszystkich. Z tego powodu wybrane zostały tylko najpopularniejsze z nich.

Mozilla M18

Wszystkie przeglądarki oparte na kodzie Netscape wyświetlają na platformie Win32 wyraźnie mniejsze fonty niż IE 5.5, ale ciężko to uznać za wadę -- można nawet uznać, że jest w tym lepsza, bo *xx-small* są naprawdę małe. Przeglądarka ta nie honorowała atrybutu *text-align: right* dla znacznika ``, ale poza tym jest godna polecenia ze względu na swoje bardzo rozbudowane możliwości.

Konqueror

Konqueror to przeglądarka pochodząca z desktop-manager'a KDE dla środowiska X-Window. Layout strony oddany dość wiernie, czcionki przy odpowiednich ustawieniach (nieingerujących w ich wielkość) wyświetlane poprawnie. Nie obsługuje *text-align* dla `` i nie parsuje entities *—*, *–*, *•* i -- co najdziwniejsze -- *©*. Najnowsze wersje Konquerora posiadają możliwość translacji fragmentów tekstu pobranych ze strony z jednego języka na inny; obsługiwane są m.in.: angielski, francuski, hiszpański, koreański niemiecki.

Opera

Kolejne przeglądarka jest jedną z najlepszych. Layout oddany wiernie jeżeli nie liczyć marginesu wokół strony (brak obsługi marginesów strony w stylach). Wszystkie encje obsługiwane poprawnie, łącznie z najbardziej problematycznymi. Brak polskich liter na screenshot'cie jest normalny, ale da się go ominąć za pomocą dostępnego na sieci dodatku. Bez problemu można również znaleźć polski język interface'u.

Lynx, Links, W3M

Przeglądarki tekstowe to dość szczególny przypadek. Nie obsługują styli, lecz oprócz *text-align* i *bold* nie mają właściwie takiej możliwości. W większości (tzn. dwie na trzy:) obsługują wszystkie entities (na swój dosyć czytelny i sensowny sposób) i polskie literki. Wygląd strony (jeżeli chodzi o entites) jest mocno zależny od ustawionego stylu kodowania. Ich wyraźnym brakiem jest niemożliwość obsługi styli

Programy biurowe

System spod znaku pingwina zdobywa również rzeszę nowych użytkowników jako komputer na biurko. Ogrom aplikacji dostępnych pod Linuksa powoduje, że przestaje to być system serwerowy, a staje się w pełni użytecznym systemem do zastosowań domowych. Z powodzeniem można w Linuksie tworzyć strony www, słuchać muzyki we wszystkich znanych formatach muzycznych, oglądać filmy, grać (nawet w gry dostępne dotychczas tylko na platformę DOS czy Windows dzięki emulatorom tych systemów), pisać programy w niemal dowolnym języku (C, C++, Java, Fortran, Perl) itp. Od strony graficznej Linux oferuje więcej niż można się spodziewać - liczba menedżerów okien, odpowiedzialnych za pracę w środowisku graficznym, nie pozwala wszystkich tu opisać. Jeżeli doliczyć do tego ilość tematów graficznych, które można do nich zastosować to pojawia się wielkość liczona w tysiącach. Taka różnorodność powoduje, że każdy może mieć na swoim komputerze swój własny, indywidualny system - dopasowany do własnych potrzeb, lub wręcz upodobań. Najbardziej rozbudowana dystrybucja – Debian, oferuje użytkownikowi prawie 9000 pakietów!

Wiodącymi środowiskami graficznymi są KDE i Gnome charakteryzujące się bardzo estetycznym i wygodnym interfejsem graficznym oraz mnogością oprogramowania współpracującego z tymi środowiskami (programy pocztowe, pakiety biurowe, przeglądarki www itp).

Warto tu wymienić kilka najpopularniejszych programów, które powoduje, że praca z Linuksem staje się przyjemna i wydajna:

pakiey biurowe:

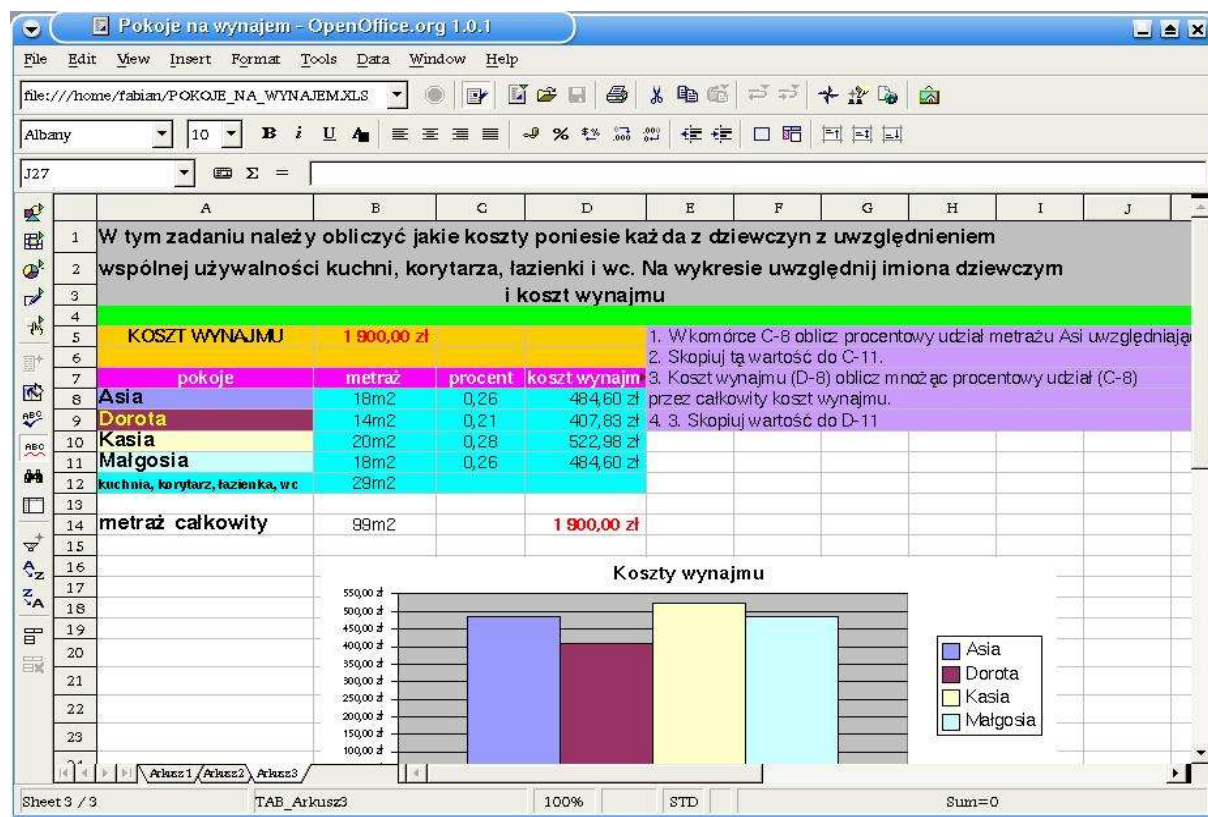
- KOffice - pakiet biurowy dla środowiska KDE
- StarOffice - zaawansowany, obecnie komercyjny pakiet biurowy
- OpenOffice - darmowy pakiet biurowy, bardzo rozbudowany i zaawansowany

oprogramowanie biurowe wchodzące w skład środowiska Gnome - Abiword, Gnumeric, Evolution, Galeon, Gimp itp.

Na linuksowe programy biurowe można się z powodzeniem przesiąść z komercyjnych programów tego typu znanych z platformy Windows, gdyż obsługa ich jest bardzo zbliżona. Poniższa tabela przedstawia programy pakietu Ms Office i ich odpowiedniki dostępne na platformę Linux.

Ms Office	KOffice	OpenOffice	Inne
Ms Word	KWord	Writer	Abiword
Ms Excel	KSpread	Calc	Gnumeric

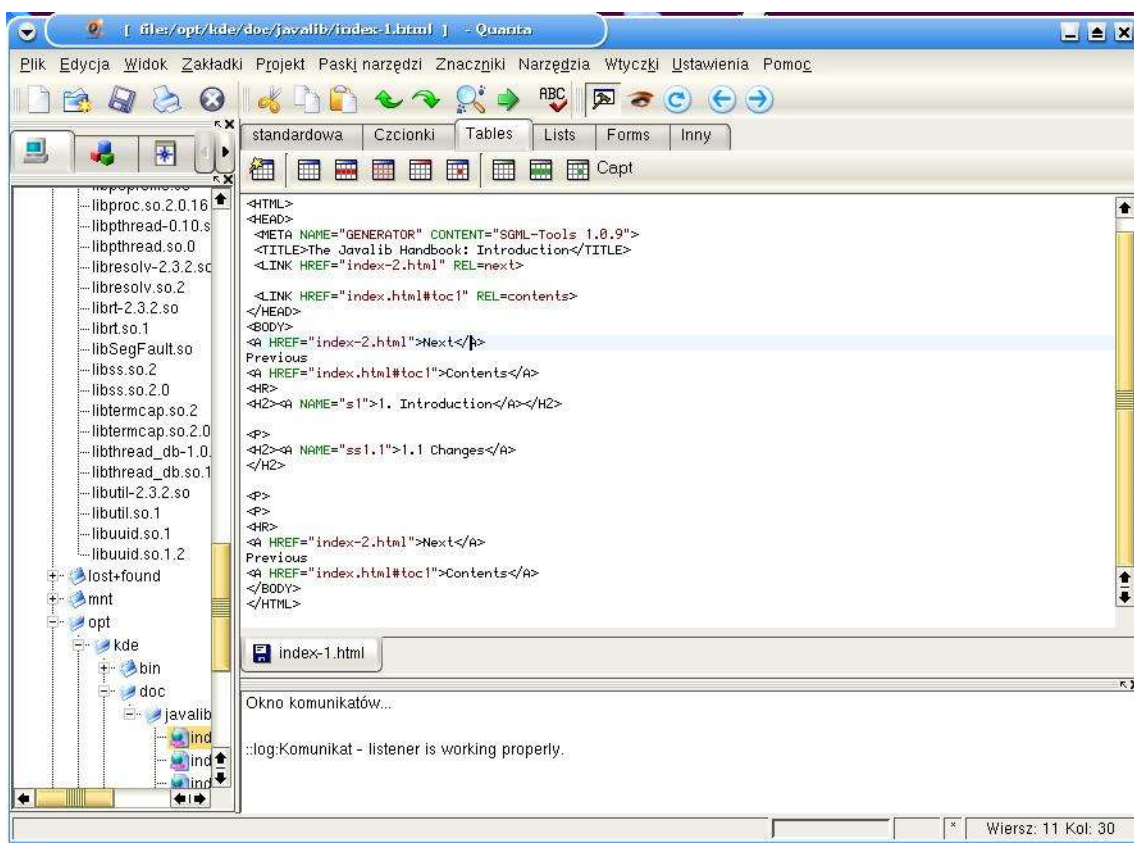
Ms PowerPoint	KPresenter	Impress	
Ms FrontPage			Balsa, Bluefish
Ms Outlook			Evolution
Ms Equation		Math	



Rys. Arkusz kalkulacyjny Calc – składnik pakietu OpenOffice w akcji

Programy internetowe i pocztowe

Każda dystrybucja Linuksa (poza minidystrybucjami oczywiście) już na etapie instalacji pozwala wybrać wiele programów użytkowych, także programów internetowych. Ich obsługa jest stosunkowo łatwa, nie odbiegają trudnością od swoich komercyjnych odpowiedników z platformy Windows czy MacOS, w związku z czym w niniejszym podrozdziale zostaną one wymienione jedynie z nazwy i bardzo krótko opisane.



Rys. Edytor Quanta podczas pracy

Przeglądarki internetowe

- Mozilla - konkurent IE, oferuje mnóstwo funkcji, jest stabilna... prawie doskonała Galeon - przeglądarka opierające się na "silniku" Mozilli
- Opera - świetna i szybka przeglądarka
- Skipstone - kolejny projekt oparty na "silniku" Mozilli
- Konqueror - przeglądarka www stworzona dla KDE
- Lynx i Links - szybkie przeglądarki tekstowe

Programy pocztowe

- Balsa - szybki i użyteczny klient pocztowy pod Gnome i KDE

- Evolution - zaawansowany pakiet pocztowy, zawierający również organizier – odpowiednik Ms Outlook z Ms Windows
- KMail - klient pocztowy dla KDE
- Mutt - przez niektórych uważany za najlepszego klienta pod konsolę
- Pine - również świetny

Czytniki grup dyskusyjnych

- Knode - czytnik news pod KDE
- Pan - doskonały klient news pod Gnome
- Sln - według wielu najlepszy czytnik news (pod konsolę)

Edytory HTML i PHP

- Bluefish - zaawansowany program do tworzenia stron www
- Screem - bardzo ciekawy edytor HTML
- Quanta Plus - również zaawansowany edytor

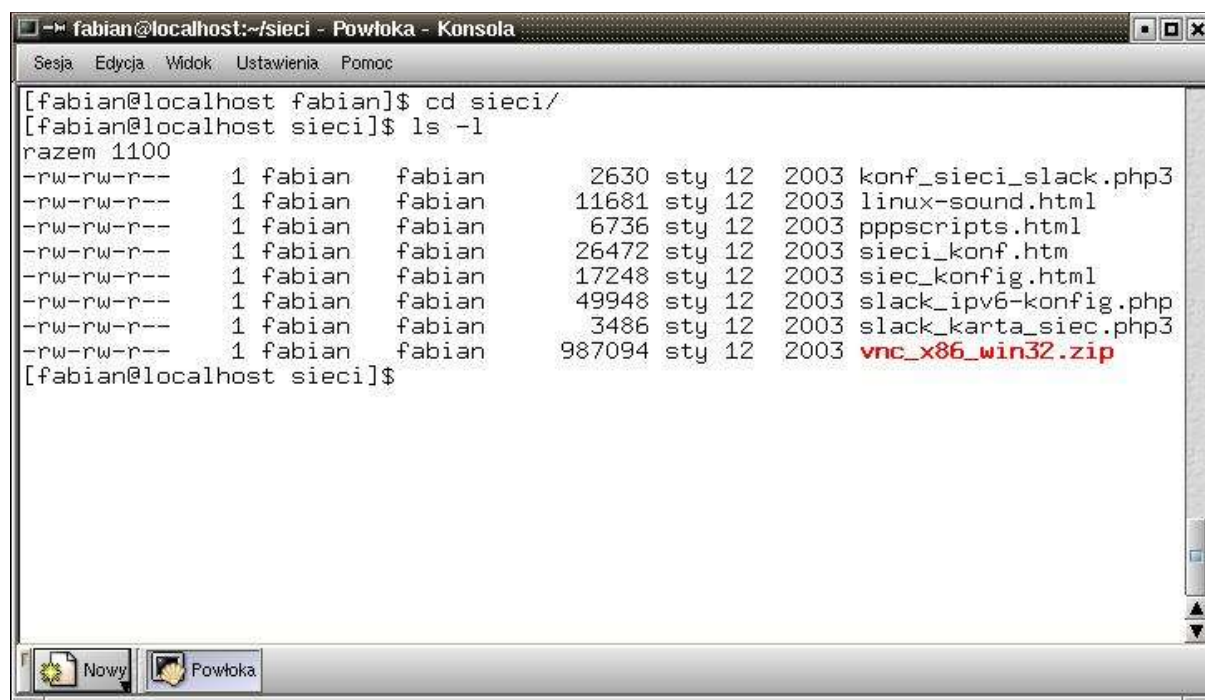
Logowanie do systemu

Pierwszym krokiem, zanim można korzystać z systemu, jest otworzenie sesji czyli zalogowanie w systemie. System prosi o login i hasło:

```
login: user
password: ****
```

Hasło podczas wpisywania nie jest widoczne na ekranie. Hasło może wygasać po określonym czasie (zależy to od konfiguracji podczas zakładania konta nowego użytkownika), i wtedy użytkownik zostanie poinformowany o tym że jego hasło wygasło i mu podać nowe. Należy często zmieniane aby zapobiec próbom nieautoryzowanego dostępu do systemu. W większości systemów uniksowych użytkownik nie posiadający praw administratora może zmieniać własne hasło. Wybierając hasło dobrze jest utworzyć kombinacje małych i dużych liter (systemy uniksowe – w przeciwieństwie do DOS-a i Windowsa, rozróżniają duże i małe litery) i cyfr, aby hasło było możliwe trudne do odgadnięcia i złamania jakimś programem do łamania haseł.

Zmiana hasła następuje po wydaniu polecenia: `passwd`



```
fabian@localhost:~/sieci - Powłoka - Konsola
Sesja  Edycja  Widok  Ustawienia  Pomoc
[fabian@localhost fabian]$ cd sieci/
[fabian@localhost sieci]$ ls -l
razem 1100
-rw-rw-r--  1 fabian  fabian    2630 sty 12  2003  konf_sieci_slack.php3
-rw-rw-r--  1 fabian  fabian   11681 sty 12  2003  linux-sound.html
-rw-rw-r--  1 fabian  fabian    6736 sty 12  2003  pppscripts.html
-rw-rw-r--  1 fabian  fabian   26472 sty 12  2003  sieci_konf.htm
-rw-rw-r--  1 fabian  fabian   17248 sty 12  2003  siec_konfig.html
-rw-rw-r--  1 fabian  fabian   49948 sty 12  2003  slack_ipv6-konfig.php
-rw-rw-r--  1 fabian  fabian    3486 sty 12  2003  slack_karta_siec.php3
-rw-rw-r--  1 fabian  fabian  987094 sty 12  2003  vnc_x86_win32.zip
[fabian@localhost sieci]$
```

Aby zakończyć sesję należy się wylogować. Polecenie wylogowujące użytkownika to :

```
logout
```

Po wpisaniu `logout` system łagodnie zamyka wszystkie procesy i kończy sesję danego użytkownika. Można też wydać polecenie:

`exit`

co powoduje szybkie wyjście z sesji z jednoczesnym przerwaniem działających procesów.

Zamykanie systemu.

Zatrzymanie systemu powoduje przerwanie wszystkich otwartych sesji, zamknięcie procesów systemowych, odmontowanie systemu plików i zatrzymanie systemu. Można to zrobić wydając polecenie:

```
shutdown -h now           //zatrzymanie systemu  
shutdown -r now         //restart systemu
```

przy czym należy zaznaczyć że w niektórych systemach użytkownik nie dysponujący prawami roota nie może zamknąć systemu ani wykonać restartu. Dodatkowo polecenie: shutdown pozwala zaprogramować opóźnienie wykonania polecenia np.:

```
shutdown -h 10 „Wyłączam serwer bo idę na kawę“
```

zatrzymuje system po 10 minutach i nie pozwala się już nikomu zalogować, wysyłając cyklicznie do wszystkich zalogowanych użytkowników podany komunikat Aby zrestartować system można użyć też polecenia :

reboot

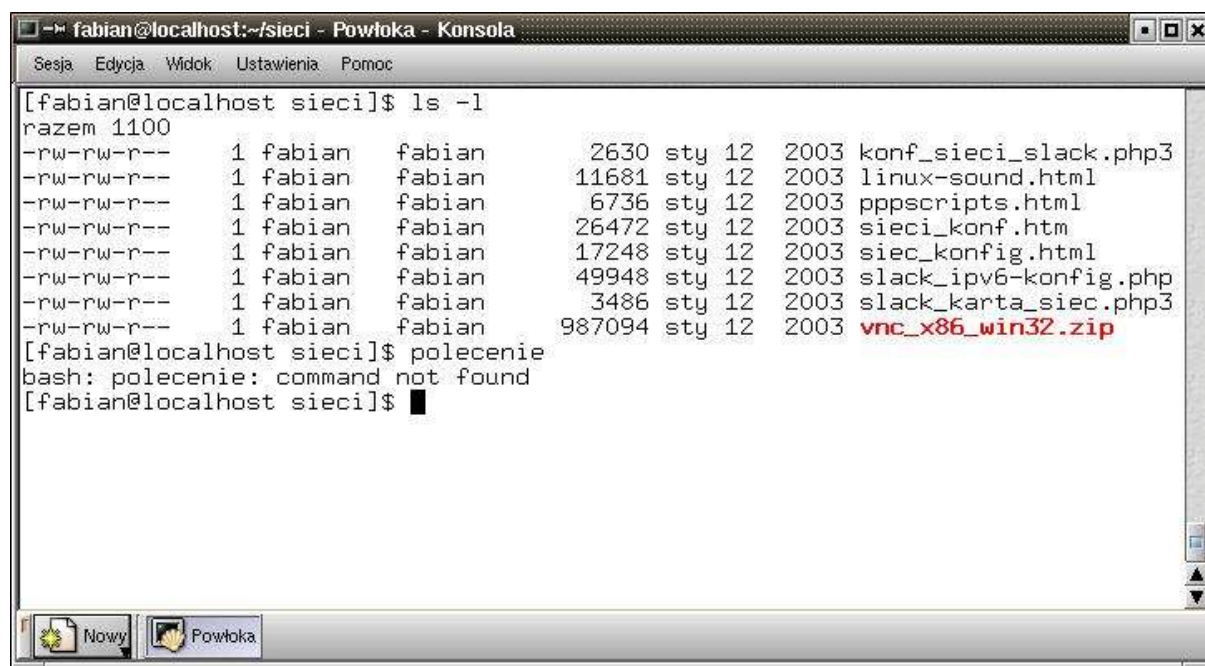
W nagłych wypadkach kiedy trzeba natychmiast zamknąć system (gdy grozi poważna awaria albo stwierdzony zostanie atak z sieci) można użyć polecenia:

halt

halt nie zamyka tak czysto systemu jak shutdown, ale robi to szybciej. Nie zamyka czysto usług i nie odmontowuje dysków.

Podstawowe polecenia i ich opcje

Każde polecenie wydane w shellu ma określoną składnię. Wpisanie wyrażenia które nie jest rozpoznawanym przez system poleceniem kończy się wyświetleniem komunikatu:



```
fabian@localhost:~/sieci - Powłoka - Konsola
Sesja Edycja Widok Ustawienia Pomoc
[fabian@localhost sieci]$ ls -l
razem 1100
-rw-rw-r-- 1 fabian fabian 2630 sty 12 2003 konf_sieci_slack.php3
-rw-rw-r-- 1 fabian fabian 11681 sty 12 2003 linux-sound.html
-rw-rw-r-- 1 fabian fabian 6736 sty 12 2003 pppscripts.html
-rw-rw-r-- 1 fabian fabian 26472 sty 12 2003 sieci_konf.htm
-rw-rw-r-- 1 fabian fabian 17248 sty 12 2003 siec_konfig.html
-rw-rw-r-- 1 fabian fabian 49948 sty 12 2003 slack_ipv6-konfig.php
-rw-rw-r-- 1 fabian fabian 3486 sty 12 2003 slack_karta_siec.php3
-rw-rw-r-- 1 fabian fabian 987094 sty 12 2003 vnc_x86_win32.zip
[fabian@localhost sieci]$ polecenie
bash: polecenie: command not found
[fabian@localhost sieci]$
```

Polecenia shella działają na plikach katalogach i urządzeniach (dyskietki, taśmy, drukarki). Może być też wykonywalnym programem napisanym przy użyciu innych poleceń shella. Podstawowa składnia polecenia wygląda następująco:

polecenie -opcje argumenty

Opcje modyfikują działanie polecenia. Większość ich to pojedyncze liczby lub litery poprzedzone znakiem minus (-).

Podstawowe polecenia Unixa/Linuxa:

ls katalog	Listuje zawartość katalogu.
ls -l	Listuje zawartość katalogu podając szczegóły dotyczące plików i katalogów.
ls -a	Wyświetla wszystkie pliki i katalogi, w tym ukryte.

cd	Przechodzi do wskazanego katalogu.
cd -	Przechodzi do poprzednio użytego katalogu.
cd ..	Przechodzi do katalogu stojącego powyżej bieżącego w strukturze katalogów.
cd	Wydane bez żadnego parametru przechodzi od katalogu zalogowanego domowego użytkownika.
logout	Wylogowuje użytkownika.
exit	Wyjście z powłoki danego użytkownika.
pwd	Wyświetla ścieżkę bieżącego katalogu roboczego.
startx	Uruchamia środowisko graficzne X-Windows.
who	Pokazuje zalogowanym w systemie użytkowników.
date	Wyświetla bieżącą datę i godzinę.
cat plik	Wyświetla zawartość pliku tekstowego.
more plik	Wyświetla zawartość pliku tekstowego ze stronicowaniem.
less plik	Wyświetla zawartość pliku tekstowego ze stronicowaniem.

```

fabian@localhost:~/sieci - Powłoka - Konsola
Sesja  Edycja  Widok  Ustawienia  Pomoc
[fabian@localhost fabian]$ cd sieci/
[fabian@localhost sieci]$ ls -l
razem 1100
-rw-rw-r--  1 fabian  fabian      2630 sty 12  2003  konf_sieci_slack.php3
-rw-rw-r--  1 fabian  fabian     11681 sty 12  2003  linux-sound.html
-rw-rw-r--  1 fabian  fabian     6736 sty 12  2003  pppscripts.html
-rw-rw-r--  1 fabian  fabian    26472 sty 12  2003  sieci_konf.htm
-rw-rw-r--  1 fabian  fabian    17248 sty 12  2003  siec_konfig.html
-rw-rw-r--  1 fabian  fabian    49948 sty 12  2003  slack_ipv6-konfig.php
-rw-rw-r--  1 fabian  fabian     3486 sty 12  2003  slack_karta_siec.php3
-rw-rw-r--  1 fabian  fabian   987094 sty 12  2003  vnc_x86_win32.zip
[fabian@localhost sieci]$

```

Rys. Działanie poleceń: **cd** oraz **ls -l**

Ćwiczenia:

Zalogować się do systemu i sprawdzić działanie polecenia „who”

1. Sprawdzić polecenie „whoami” a następnie „uname -a”
2. Poleceniem „ls” wyświetlić zawartość swojego katalogu
 - a. To samo tylko „ls -a”
 - b. ls -la
 - c. ls -Rla
3. cd / aby przejść do katalogu /
4. Używając „cd” przejść po kolei do wszystkich katalogów systemowych, wyświetlić zawartość każdego z nich przez „ls -l”
5. Wypróbować działanie „cd - „i „cd ..”
6. Poleceniem „touch” w swoim katalogu domowym utworzyć pliki:
 - a. Imie
 - b. System
 - c. Kernel
 - d. Procesor
 - e. Partycje
7. Odnaleźć pliki :
/var/spool/mail
/bin/mount
/usr/src/linux
/etc/lilo.conf
i podejrzeć ich zawartość.
9. Utworzyć katalogi:
/home/kurs/katalog1
/home/kurs/katalog1/podkatalog1
/home/kurs/katalog2
10. Skopiować:
/var/spool/mail/kurs do /home/kurs/katalog1
/home/kurs/katalog1 do /home/kurs/katalog2
/home/kurs/plik1 do /home/kurs/katalog1/podkatalog1
/var/spool/mail/kurs do /home/kurs
11. Używając „mcedit” wpisać zawartość do plików utworzonych poprzednio

przez „touch”

12. *Poleceniem „cat” wyświetlić ich zawartość w konsoli.*

13. *Wypróbować „pwd”*

14. *Porównać działanie „logout” i „exit”*

15. *Uruchomić X-y, odpalić xterm i wylogować się z X-ów*

16. *Porównać działanie „shutdown” z różnymi parametrami i „reboot” oraz „halt”*

Gdzie szukać pomocy ?

Każde z poleceń systemu Linux jest dobrze opisane w podręczniku systemowym – manualu. Dostęp do niego uzyskuje się wydając komendę :

man

Np.:

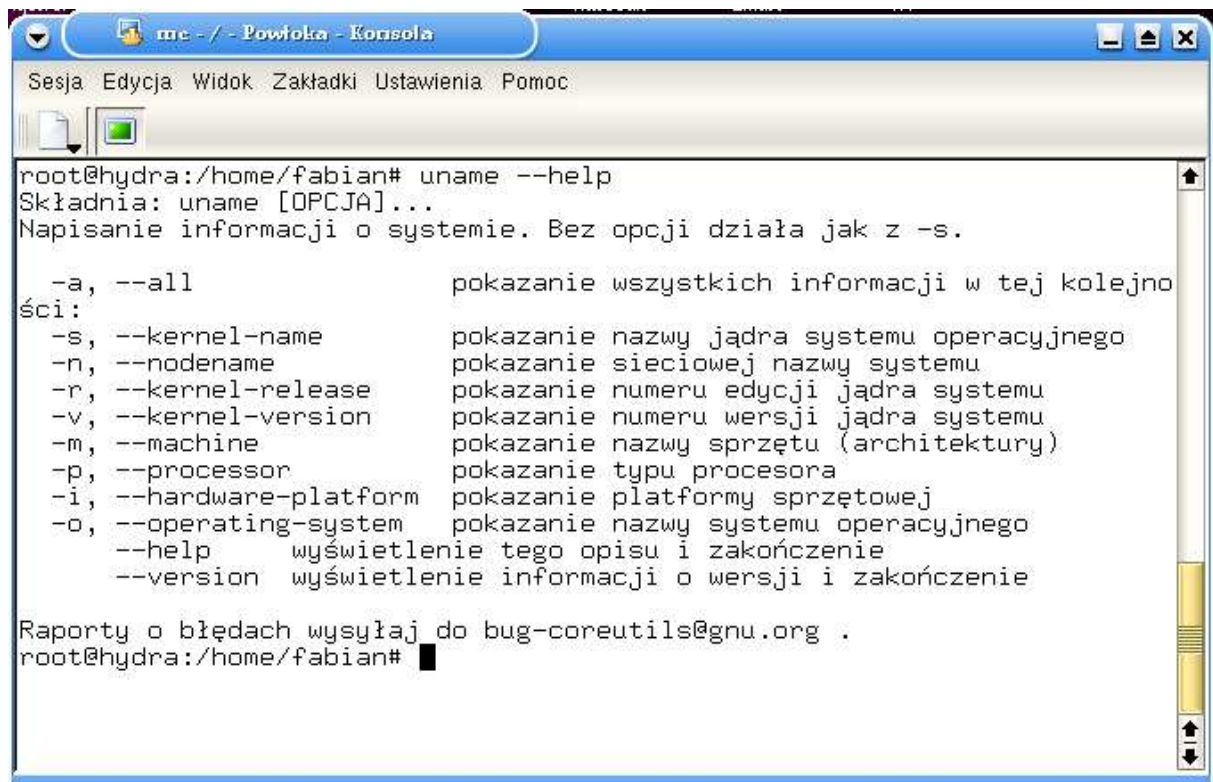
man cat

Wyświetli pomoc dotyczącą polecenia cat. Aby wyjść z manuala wystarczy wcisnąć **q**. Oprócz dokładnego opisu w manualu dostępna jest też pomoc dotyczącą poleceń, zawierająca składnię polecenia i opis najpowszechniejszych parametrów. Należy wpisać w konsoli:

polecenie --help (help po dwóch minusach!)

np.:

cat --help



```
root@hydra:/home/fabian# uname --help
Składnia: uname [OPCJA]...
Napisanie informacji o systemie. Bez opcji działa jak z -s.

  -a, --all                pokazanie wszystkich informacji w tej kolejno
ści:
  -s, --kernel-name       pokazanie nazwy jądra systemu operacyjnego
  -n, --nodename           pokazanie sieciowej nazwy systemu
  -r, --kernel-release    pokazanie numeru edycji jądra systemu
  -v, --kernel-version    pokazanie numeru wersji jądra systemu
  -m, --machine            pokazanie nazwy sprzętu (architektury)
  -p, --processor         pokazanie typu procesora
  -i, --hardware-platform pokazanie platformy sprzętowej
  -o, --operating-system  pokazanie nazwy systemu operacyjnego
  --help                  wyświetlenie tego opisu i zakończenie
  --version               wyświetlenie informacji o wersji i zakończenie

Raporty o błędach wysyłaj do bug-coreutils@gnu.org .
root@hydra:/home/fabian#
```

Rys. Przykład polecenia `uname --help`.

Podobną funkcję pełni pomoc "`info`", do której można odwołać się podobnie jak do manuala systemowego poprzez:

info polecenie

np.:

```
info ls
```

Nie zawsze jednak użytkownik zna nazwę polecenia realizującego poszukiwaną funkcję. W sytuacji takiej można przeszukać pomoc kontekstową pod kątem określonych wyrazów. Dokonać tego można pisząc:

```
apropos szukane wyrazy
```

W przypadku podania więcej niż jednego wyrazu, odszukane zostaną strony pomocy zawierające w nagłówku dowolny z podanych wyrazów. Dla przykładu - by przypomnieć sobie polecenie służące do wyświetlania zawartości katalogów należy wydać polecenie:

```
apropos directory
```

Dodatkowo zauważyć należy, że wyświetlona jest informacja o numerze rozdziału w którym znajduje się dane polecenie.

Innym narzędziem do przeglądania systemu pomocy jest komenda "**whatis**". W wyniku jej działania wyświetlony zostaje krótki opis poszukiwanej komendy. Przykład:

```
whatis ls
```

Argumentem komendy "**whatis**" jest nazwa polecenia. Jeżeli podana zostanie nieprawidłowa nazwa polecenia, wówczas system poinformuje o tym stosownym komunikatem.

Skróty klawiszowe w konsoli

Stosując skróty klawiszowe można znacznie usprawnić pracę w shellu Linuksa. Np. W shellu BASH (domyślnym shellu Linuksa) uderzenie w klawisz TAB uzupełnia nazwy plików i katalogów po wpisaniu ich pierwszych liter:

jeśli w bieżącym katalogu jest plik `moj_plik`

to wpisanie:

```
more mo
```

i uderzenie w TAB uzupełni nazwę do:

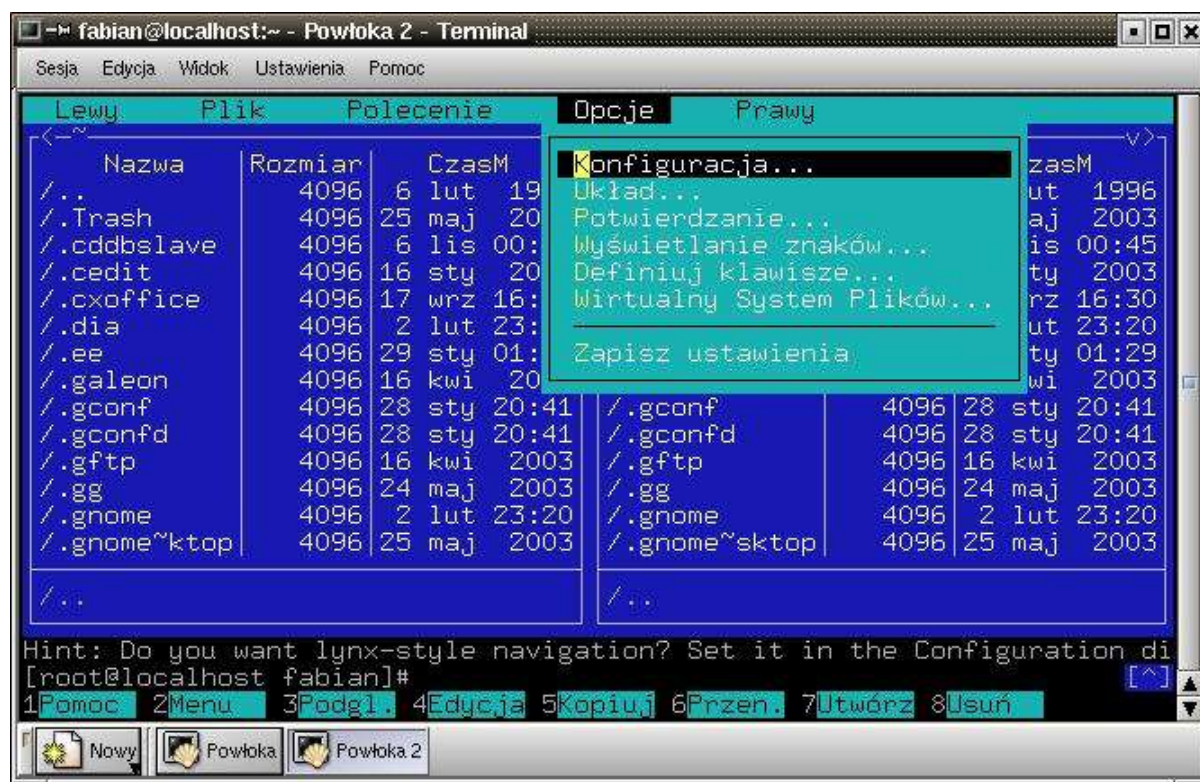
```
more moj_plik.
```

Teraz uderzenie w ENTER wykona to polecenie, czyli `more moj_plik`.
Poniżej przedstawione są podstawowe skróty klawiszowe używane w konsoli.

Tab	uzupełnia nazwę pliku/katalogu w bieżącym katalogu roboczym.
PageUp/PageDown	W Bashu przewija polecenia zaczerpnięte z bash_history .
Shift+PageUp/PageDown.	przewijanie odczytanego tekstu
Ctrl+D	przewinięcie na koniec.
Ctrl+Z	zawieszenie procesu.
Ctrl+C	przerwanie procesu
Ctrl+Alt+F1 do F6	uruchomienie kolejnego terminala tty.
Ctrl+Alt+Num- /Num+ -	przewijanie kolejnych terminali.
Ctrl+Alt+F7 -	uruchomienie X-ów.
q	wyjście.

Midnight Commander

Midnight Commander jest podstawowym i najczęściej używanym menedżerem plików dostępnym w shellu. Jest dostępny w każdej dystrybucji Linuksa. Poza podstawowymi operacjami na plikach i katalogach może pełnić funkcje klienta FTP, edytora tekstu. Pozwala także na kopiowanie plików między hostami w sieci. Obsługa jest bardzo zbliżona do DOS-owego Norton Commandera



Rys. Midnight Commander

Polecenia uruchamiające:

- mc** – uruchomienie bez parametru
- mc -a** – tryb podstawowy
- mc -b** – tryb czarno biały
- mc -c** – tryb kolorowy
- mc -d** – wyłączenie myszy
- man mc** – manual dla mc

Klawisze nawigacyjne:

F1 – menu
F2 – zapis
F3 – podgląd
F4 – edycja
F5 - kopiuj
F6 - przeniesienie
F7 – usuń
F8 – utwórz katalog
F10 – wyjście

Aby MC wyświetlał polskie znaki konieczne jest przełączenie wyświetlania na „pełne 8 bitowe wyjście” :

F9 -> Opcje -> Wyświetlanie znaków -> Pełne 8 bitowe wyjście.

Aby ustawienia zostały zachowane należy je zapisać:

F9 -> Opcje -> Zapisz ustawienia

Jeśli układ klawiatury nie jest przełączany automatycznie podczas startu systemu, należy przełączyć ją na układ „polski-programisty”, wydając polecenie:

loadkeys pl

Konfiguracja MC jest zapisana w pliku **.mc** w katalogu domowym każdego użytkownika.

Ćwiczenia:

*1. Uruchomić **mc** po kolei na różnych terminalach ze wszystkimi parametrami oraz w X-ach. Przełączyć się na polski układ klawiatury i ustawić wyświetlanie polskich znaków w mc. Zapisać ustawienia.*

Przejsć do /home a potem spróbować przejść do katalogów innych użytkowników.

Podejrzeć pliki w katalogu /home/Desktop. Zamknąć edycję lub podgląd. Uruchomić edycję a potem podgląd wybranych plików w swoim katalogu domowym.

2. Odnaleźć pliki :

/var/spool/mail

/bin/mount

*/usr/src/linux
/etc/lilo.conf
i podejrzeć ich zawartość.*

3. Utworzyć katalogi:

*/home/kurs/katalog1
/home/kurs/katalog1/podkatalog1
/home/kurs/katalog2*

4. Skopiować:

*/var/spool/mail/kurs do /home/kurs/katalog1
/home/kurs/katalog1 do /home/kurs/katalog2
/home/kurs/plik1 do /home/kurs/katalog1/podkatalog1
/var/spool/mail/kurs do /home/kurs*

5. Otworzyć plik /home/kurs/katalog1/podkatalog1/plik1 do edycji i dopisać na końcu swoje imię i nazwisko, dzisiejszą datę i nazwę roboczej dystrybucji OS. Skopiować w/w plik do /home/kurs/katalog2, usunąć swoje dane z końca pliku i umieścić je na początku i zapisać.

6. Utworzyć katalog ftp-download w swoim katalogu domowym

7. Otworzyć sesję FTP na wskazanym serwerze, zalogować się jako: anonymous podając jako hasło swój e-mail (lub dowolny). Skopiować wskazane przez prowadzącego pliki.

8. Usunąć wszelkie założone katalogi

9. Wykonać kopie pliku .bash_history.

10. Uruchomić edytor kopie .bash_history w mcedit i wykasować 10 ostatnich wierszy z pliku.

11. Zamknąć mc przez F10 i exit na wszystkich terminalach i w X-ach

Podstawy edytora vi

vi (pisany małymi literami) jest edytorem tekstów umożliwiającym zapis i modyfikację plików tekstowych. Istnieje kilka powodów dla których warto poznać ten mało przyjazny z początku edytor:

- vi jest jedynym edytorem tekstowym, który na pewno znajdzie się na każdym komputerze z systemem UNIX/Linux;
- vi umożliwia edycję bardzo wielkich plików
- vi posiada wyjątkowi duże możliwości
- vi pozwala na szybkie i efektywne wykonanie wielu złożonych operacji.

vi działa w **trybie poleceń** i **trybie wstawiania**:

W trybie poleceń wydajesz polecenia mówiące, co mamy zrobić a w trybie wstawiania wpisuje się tekst do pliku, który jest aktualnie edytowany. Po uruchomieniu programu aktywny jest tryb poleceń. Przejście do trybu wstawiania uzyskuje się przez klawisz INSERT, powrót po trybu poleceń przez ESC.

i Tekst (insert)	wstawia łańcuch znaków "Tekst" zaczynając od pozycji kursora
a Tekst (append)	dodaje łańcuch znaków "Tekst"
o Tekst (open)	otwiera nowa pustą linię poniżej kursora i wstawia "Tekst"
O Tekst (open)	otwiera nowa pustą linię powyżej kursora i wstawia "Tekst"
J (join)	łączy aktualną linię z następującą po niej
x	usuwa znak, na którym znajduje się kursor
u (undo)	cofa ostatnią modyfikację...
.	powtarza ostatnią modyfikację
:w (write)	zapisuje aktualnie otwarty plik
:w innyplik	zapisuje zawartość do nowego pliku o nazwie „inny plik"
:e innyplik	wczytuje do edycji plik "innyplik"
:q (quit)	wyjście; jeśli nie zapisano zmian program poprosi o potwierdzenie
:q! (quit!)	wyjście bez zapisywania ostatnich modyfikacji
:wq (write quit)	zapisz i wyjdź
:7	przemieszcza kursor do początku siódmej linii
G	przemieszcza kursor do końca pliku
dd (delete)	usuwa aktualną linię
x	Kasuje znak na którym stoi kursor
/tekst	Szuka tekstu „tekst"
:set list	wyświetla znaki ukryte (w trybie poleceń)
:set number	włącza numerację wierszy
:set nonumber	wyłącza numerację wierszy

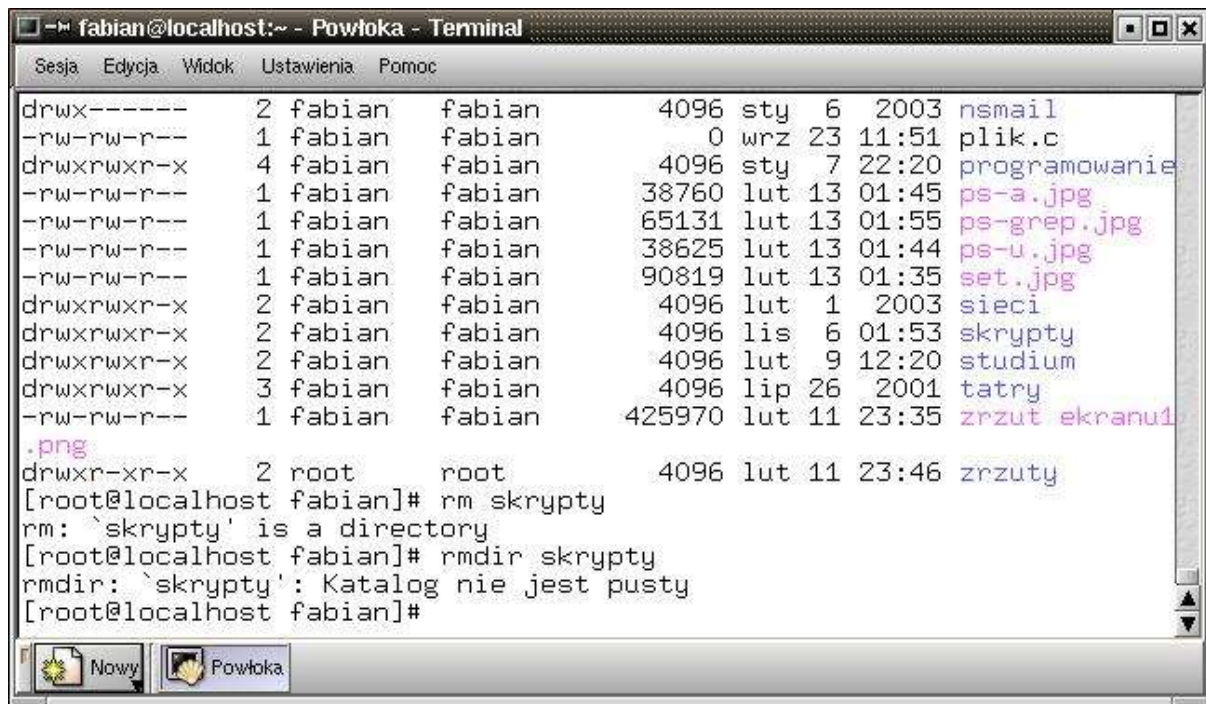
Przesuwanie kursora się po edytowanym pliku obsługują klawisze:

- h** przesuwają kursor w lewo
- j** przesuwają kursor w dół
- k** przesuwają kursor do góry
- l** przesuwają kursor w prawo

Pełna lista opcji edytora vi jest oczywiście dłuższa. Więcej informacji na ten temat zawarte jest w podręczniku systemowym **man**

Kopiowanie, przenoszenie, usuwanie plików i katalogów

<code>mkdir katalog</code>	utworzenie katalogu.
<code>cat plik</code>	wyświetlenie pliku.
<code>cat>plik</code>	utworzenie pliku tekstowego
<code>touch plik</code>	utworzenie pustego pliku.
<code>more plik</code>	Stronicowanie pliku tekstowego.
<code>rm plik</code>	usuniecie pliku i katalogu.
<code>rm -R</code>	kasowanie rekurencyjne – usuwa całe poddrzewa katalogów.
<code>rm -dR</code>	kasowanie katalogu wraz z zawartością – z zejściem włąb katalogu do podkatalogów.
<code>rm -f</code>	kasowanie „force” – bez pytania.
<code>rmdir katalog</code>	usuniecie pustego katalogu.
<code>mv plik plik1</code>	zmiana nazwy pliku/katalogu lub przeniesienie.
<code>mv plik ~/kat/plik</code>	przeniesienie pliku .
<code>cp plik plik2</code>	kopiowanie plików i katalogów.
<code>cp -R katalog ~/kat/katalog</code>	kopiowanie rekursywne – kopiuje całe poddrzewa katalogów.
<code>cp -f</code>	kopiowanie z nadpisaniem plików istniejące o takich samych nazwach.
<code>cp -i</code>	kopiowanie interaktywne - pyta przed nadpisaniem.
<code>cp -v</code>	verbose –tryb „gadatliwy”.
<code>cp -l</code>	Tworzy link zamiast kopiować.



```
fabian@localhost:~ - Powłoka - Terminal
Sesja  Edycja  Widok  Ustawienia  Pomoc
drwx-----  2 fabian  fabian      4096 sty  6  2003  nsmail
-rw-rw-r--  1 fabian  fabian        0 wrz  23  11:51  plik.c
drwxrwxr-x  4 fabian  fabian      4096 sty  7  22:20  programowanie
-rw-rw-r--  1 fabian  fabian     38760 lut  13  01:45  ps-a.jpg
-rw-rw-r--  1 fabian  fabian     65131 lut  13  01:55  ps-grep.jpg
-rw-rw-r--  1 fabian  fabian     38625 lut  13  01:44  ps-u.jpg
-rw-rw-r--  1 fabian  fabian     90819 lut  13  01:35  set.jpg
drwxrwxr-x  2 fabian  fabian      4096 lut  1  2003  sieci
drwxrwxr-x  2 fabian  fabian      4096 lis  6  01:53  skrypty
drwxrwxr-x  2 fabian  fabian      4096 lut  9  12:20  studium
drwxrwxr-x  3 fabian  fabian      4096 lip  26  2001  tatry
-rw-rw-r--  1 fabian  fabian    425970 lut  11  23:35  zrzut_ekranu1
.png
drwxr-xr-x  2 root    root        4096 lut  11  23:46  zrzuty
[root@localhost fabian]# rm skrypty
rm: `skrypty' is a directory
[root@localhost fabian]# rmdir skrypty
rmdir: `skrypty': Katalog nie jest pusty
[root@localhost fabian]#
```

Rys. Próba usunięcia niepełnego katalogu poleceniem `rmdir`

Ćwiczenia:

1. Utworzyć katalog *Kat1* w swoim katalogu domowym, a w nim trzy pliki tekstowe zawierające: imię i nazwisko, nazwę systemu operacyjnego, i dowolny tekst.
2. Wyświetlić bieżący katalog roboczy (`pwd`) zawartość katalogu i plików tekstowych.
3. Wypróbować programy `more` i `less` przy wyświetlaniu zawartości katalogu z plikami ukrytymi (`ls -al`).
4. Skopiować istniejące pliki bez zmiany katalogu, zmieniając jedynie ich nazwy (`cp`)
5. Utworzyć nowe dwa katalogi *Kat2* i *Kat3* w swoim katalogu domowym.
6. Skopiować do nich istniejące pliki oraz pliki:
 - `/var/spool/mail/`
 - `/home/kurs/bash.history`
 - `/home kurs/Desktop`.
7. Zmienić nazwy katalogów *Kat2* na *katalog2* i *Kat3* na *katalog3* (`mv`).

8. *Utworzyć katalog katalog4 i przenieść do niego wszystkie założone katalogi (mv).*
9. *Utworzyć puste pliki tekstowe. Przekierowaniem strumienia danych zapisać w nich:*
 - *bieżąca datę*
 - *zawartość swojego katalogu domowego*
 - *zawartość utworzonych wcześniej plików*
 - *wydane dotychczas polecenia*
 - *wpis dotyczący danego użytkownika z pliku etc/passwd*
10. *Usunąć wszystkie istniejące katalogi (rmdir -dr).*

Montowanie dyskietki, CD-ROMu i innych systemów plików

Aby partycja z innym systemem, lub inny dysk oraz CDROM były widziane przez Linuksa należy je zamontować. W tym celu w katalogu `/mnt/` tworzy się katalogi które będą odnosiły się do dysków i partycji innych systemów operacyjnych. Katalog może mieć dowolną nazwę np: dla CDROM-u można utworzyć `/mnt/cdrom`, a dla partycji z systemem Windows `/mnt/winda`. W katalogu `/mnt` z reguły są już katalogi `cdrom`, `floppy` i niekiedy `hd`. Po zamontowaniu dany system plików będzie widoczny w wybranym podkatalogu katalogu `/mnt`.

Inne systemy plików, jeśli tylko pozwala nam na to nasz system można montować w trybie do odczytu, zapisu lub z oboma wymienionymi atrybutami. Po zakończeniu pracy z danym systemem plików należy pamiętać aby go odmontować.

Polecenie montowania ma składnię:

```
mount [paramerty] co gdzie np.:
```

```
mount -t vfat /dev/hda1 /mnt/windows
```

Powyższe polecenie zamontuje windowsowy dysk C: z systemem plików FAT w katalogu `/mnt/windows`

Aby móc skorzystać z dyskietki lub CD-ROM należy więc wydać polecenie:

```
mount /dev/fd0 /mnt/floppy lub
mount /mnt/floppy
mount /dev/cdrom /mnt/cdrom
monut /mnt/cdrom
```

-t vfat	Montuje system plików FAT (DOS, Windows 9x/Me/XP)
-t ntfs	Montuje system plików NTFS (Windows NT/2000/XP)
-t ufs	Montuje system plików UFS (UNIX)
-o remount	przemontowanie zamontowanego już systemu plików
-o ro	montuje system plików w tryb tylko do odczytu
-o rw	montuje system plików w tryb tylko do odczytu i zapisu
-o nouser	zabrania zwykłemu użytkownikowi montowania systemu plików

Do odmontowania plików służy polecenie `umount` o składni:

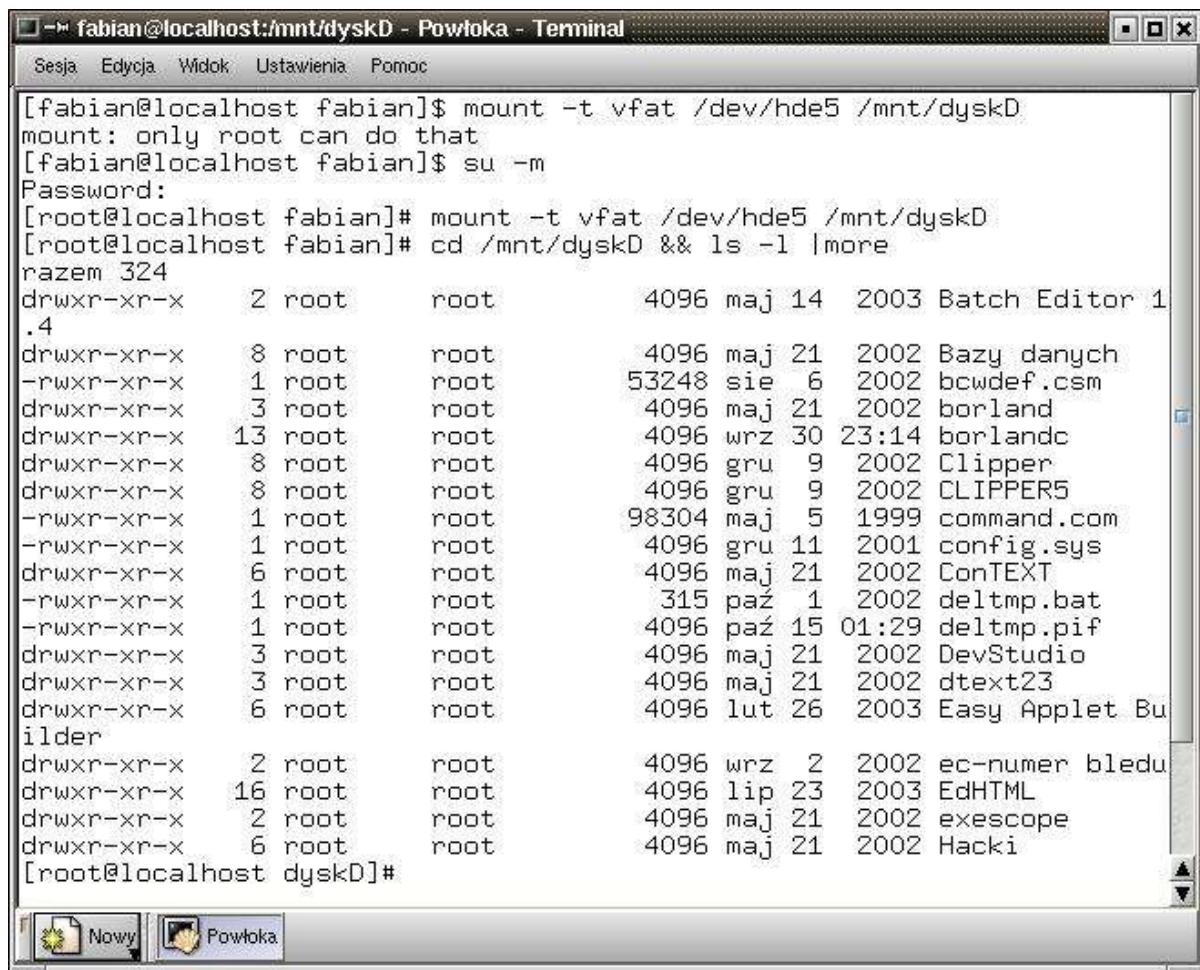
```
umount /mnt/cdrom
```

Które w powyższym przypadku odmontowuje CD-ROM. Inne przykłady:

```
umount /mnt/floppy
```

```
umount /mnt/hd
```

```
umount /home/andrzej/udostepnione
```



```
fabian@localhost:~/mnt/dyskD - Powłoka - Terminal
Sesja  Edycja  Widok  Ustawienia  Pomoc
[fabian@localhost fabian]$ mount -t vfat /dev/hde5 /mnt/dyskD
mount: only root can do that
[fabian@localhost fabian]$ su -m
Password:
[root@localhost fabian]# mount -t vfat /dev/hde5 /mnt/dyskD
[root@localhost fabian]# cd /mnt/dyskD && ls -l |more
razem 324
drwxr-xr-x    2 root    root      4096 maj 14  2003 Batch Editor 1
.4
drwxr-xr-x    8 root    root      4096 maj 21  2002 Bazy danych
-rwxr-xr-x    1 root    root     53248 sie  6  2002 bcwdef.csm
drwxr-xr-x    3 root    root      4096 maj 21  2002 borland
drwxr-xr-x   13 root    root      4096 wrz 30  2002 23:14 borlandc
drwxr-xr-x    8 root    root      4096 gru  9  2002 Clipper
drwxr-xr-x    8 root    root      4096 gru  9  2002 CLIPPER5
-rwxr-xr-x    1 root    root     98304 maj  5  1999 command.com
-rwxr-xr-x    1 root    root      4096 gru 11  2001 config.sys
drwxr-xr-x    6 root    root      4096 maj 21  2002 ConTEXT
-rwxr-xr-x    1 root    root      315 paź  1  2002 deltmp.bat
-rwxr-xr-x    1 root    root      4096 paź 15  2002 01:29 deltmp.pif
drwxr-xr-x    3 root    root      4096 maj 21  2002 DevStudio
drwxr-xr-x    3 root    root      4096 maj 21  2002 dtext23
drwxr-xr-x    6 root    root      4096 lut 26  2003 Easy Applet Bu
ilder
drwxr-xr-x    2 root    root      4096 wrz  2  2002 ec-numer bledu
drwxr-xr-x   16 root    root      4096 lip 23  2003 EdHTML
drwxr-xr-x    2 root    root      4096 maj 21  2002 exescope
drwxr-xr-x    6 root    root      4096 maj 21  2002 Hacki
[root@localhost dyskD]#
```

Rys. Polecenie **mount** i zamontowana partycja FAT z systemem Windows.

Domyślnie tylko superużytkownik czyli root ma prawo montować inne systemy plików, co widać na powyższej ilustracji.

Zawsze przed odmontowaniem należy być poza katalogiem który jest odmontowywany. W przeciwnym wypadku system wyświetli komunikat błędu, w stylu „device is busy” czyli, że urządzenie jest zajęte.

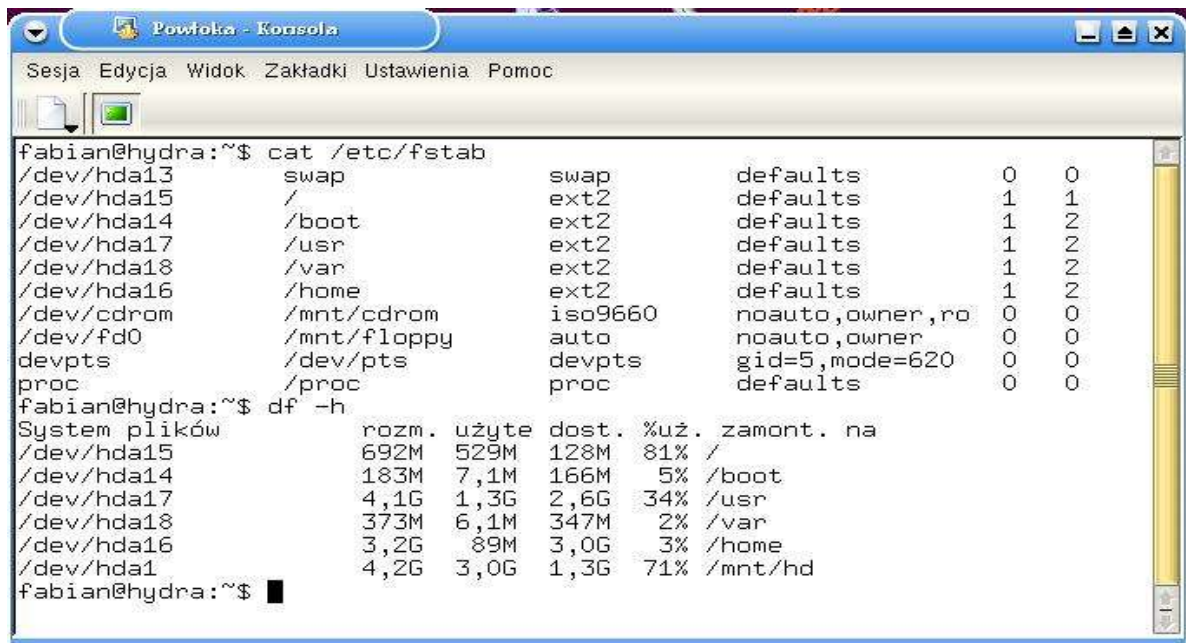
Ćwiczenia:

1. *Utworzyć w swoim katalogu domowym 3 katalogi.*
2. *Zamontować CD-ROM i skopiować kilka wskazanych plików do utworzonych wcześniej katalogów.*
3. *Zamontować dyskietkę, usunąć z niej wszystkie pliki jedną komendą **rm**. Następnie skopiować na dyskietkę kilka wybranych plików i katalogów z CD-ROMu i z katalogu domowego.*
4. *Skopiować z CD-ROMu na dyskietkę kilka plików.*
5. *Odmontować dyskietkę i CD-ROM i zamontować następnie wybraną partycję w trybie „tylko do odczytu” a następnie spróbować usunąć z niej wybrany plik lub katalog.*
6. *Przetestować działanie opcji parametru **-t**.*

Konfiguracja /etc/fstab

Metoda ręcznego montowania plików jest niewygodna przypadku częstego korzystania z danego systemu plików. Istnieje możliwość automatycznego montowania plików. Aby Linux automatycznie zamontował twoją partycję, musisz dodać jej nazwę w pliku **/etc/fstab**.

Plik **/etc/fstab** jest zwykłym plikiem tekstowym, w którym umieszczona jest tabela informacji o dostępnych w systemie systemach plikowych - partycje linuxowe, windowsowe, cdrom`y i inne.



```
fabian@hydra:~$ cat /etc/fstab
/dev/hda13      swap          swap          defaults      0      0
/dev/hda15      /             ext2          defaults      1      1
/dev/hda14      /boot        ext2          defaults      1      2
/dev/hda17      /usr         ext2          defaults      1      2
/dev/hda18      /var         ext2          defaults      1      2
/dev/hda16      /home        ext2          defaults      1      2
/dev/cdrom      /mnt/cdrom   iso9660       noauto,owner,ro 0      0
/dev/fd0        /mnt/floppy  auto          noauto,owner   0      0
devpts         /dev/pts     devpts        gid=5,mode=620 0      0
proc           /proc        proc          defaults      0      0
fabian@hydra:~$ df -h
System plików    rozmi. użyte dost.  %uż. zamont. na
/dev/hda15       692M  529M  128M  81% /
/dev/hda14       183M   7,1M  166M   5% /boot
/dev/hda17       4,1G   1,3G  2,6G  34% /usr
/dev/hda18       373M   6,1M  347M   2% /var
/dev/hda16       3,2G    89M   3,0G   3% /home
/dev/hda1        4,2G   3,0G   1,3G  71% /mnt/hd
fabian@hydra:~$
```

Rys. Zawartość pliku **/etc/fstab** i sprawdzenie wykorzystania dysków przez **df -h**.

Plik **/etc/fstab** zawiera kilka pól oddzielonych tabulatorem lub spacją:

- Pierwsze pole jest nazwą systemu plików do zamontowania np.: **/dev/hda1** oznacza pierwszą partycję dysku twardego.
- Drugie pole to katalog w systemie plików Linuksa, tam zostanie zamontowany system plików danego urządzenia np: **/mnt/winda**
- Trzecie pole oznacza typ montowanego systemu np: **vfat**
- Następne pole to opcje montowania np: **b defaults**

- **defaults**, oznacza że urządzenie jest do odczytu i zapisu, jest asynchroniczne , blokowe, zwykły użytkownik nie może montować na nim innych systemów plików, mogą na nim być wykonywane programy

- **ro**, oznacza system tylko do odczytu
 - **noauto**, system nie jest automatycznie montowany. Opcja ta wykorzystywana jest z CDROM i napędami dyskietek, ponieważ nie wiadomo czy w czasie uruchamiania systemu, będziesz miał cokolwiek w napędach.
 - **async**, wszystkie operacje we/wy związane z systemem plików powinny być wykonywane asynchronicznie
 - **auto**, system plików ma być zamontowany automatycznie.
 - **exec**, oznacza pozwolenie na wykonywanie programów. - **nouser**, zabrania zwykłemu użytkownikowi montowania systemu plików.
 - **rw**, do odczytu i zapisu.
 - **sync**, wszystkie operacje we/wy powinny być wykonywane synchronicznie.
 - **user**, zezwala na montowanie plików zwykłemu użytkownikowi.
- Piąte pole to wartość wykorzystywana przez polecenie **dump** w celu określenia, czy ma zostać wykonana kopia zapasowa.
 - Ostatnie pole ustawia wartość dla polecenia **fsck**
 - 1 oznacza sprawdzenie systemu
 - 0 oznacza, że system plików nie będzie sprawdzany

Kilka przykładów:

```
/dev/hda15          /          ext2 defaults 1 1
```

to jest główna partycja linuksowa, zamontowana w punkcie root /, system plików linuxa - ext2, opcje domyślne.

```
/dev/hda16          /home     ext2 defaults 1 2
```

oddzielna partycja /dev/hda16 zamontowana jako /home, opcje domyślne.

```
/dev/hda7          swap      swap defaults 0 0
```

partycja wymiany z systemem plików SWAP. Dwa zera na końcu oznaczają że nie będzie ona sprawdzana przez **fsck** ani archiwizowana przez **dump**.

<code>/dev/cdrom</code>	<code>/cdrom</code>	<code>iso9660 noauto,owner,ro 0</code>
-------------------------	---------------------	--

CD-ROM widoczny w katalogu /cdrom, opcja **owner** mówi, że tylko właściciel (root) może ją montować. Aby mógł to robić każdy, wystarczy opcję **owner** zamienić na **user**. Podobnie jest w przypadku montowania dyskietek. Opcja **noauto** nie pozwala automatycznie zamontować CD-ROMu podczas startu systemu ponieważ nie wiadomo czy w czasie uruchamiania systemu, będzie cokolwiek w napędach.

Poza tym są tam też wpisy odnoszące się do pseudofilesystemów (/proc) co zostanie szerzej omówione w podręczniku opisującym administrację systemem Linux.

Shelle i zmienne środowiskowe

Powłoka (shell) jest podstawowym programem, interpretującym polecenia i rozkazy w wierszu poleceń, i decydujący o ich wykonaniu. Służy do komunikacji między użytkownikiem a jądrem systemu. Jest odpowiednikiem pliku **command.com** znanego z systemów DOS/Windows. Istnieje kilka rodzajów shelli, najbardziej popularnym jest powłoka **BASH** (Bourne Again Shell), i ona jest domyślnym shellem w Linuksie. Inne najbardziej znane powłoki to **PDKSH** (Public Domain Corn Shell), **SH** (shell Bourne'a) i **TCSH**. Aby zmienić powłokę, należy napisać w linii poleceń:

```
tcsh lub ksh
```

i naciśnij ENTER. Aby wrócić z powrotem do powłoki domyślnej wystarczy wyjść z poprzedniej powłoki przez wydanie polecenia:

```
exit lub kombinację klawiszy CTRL+d.
```

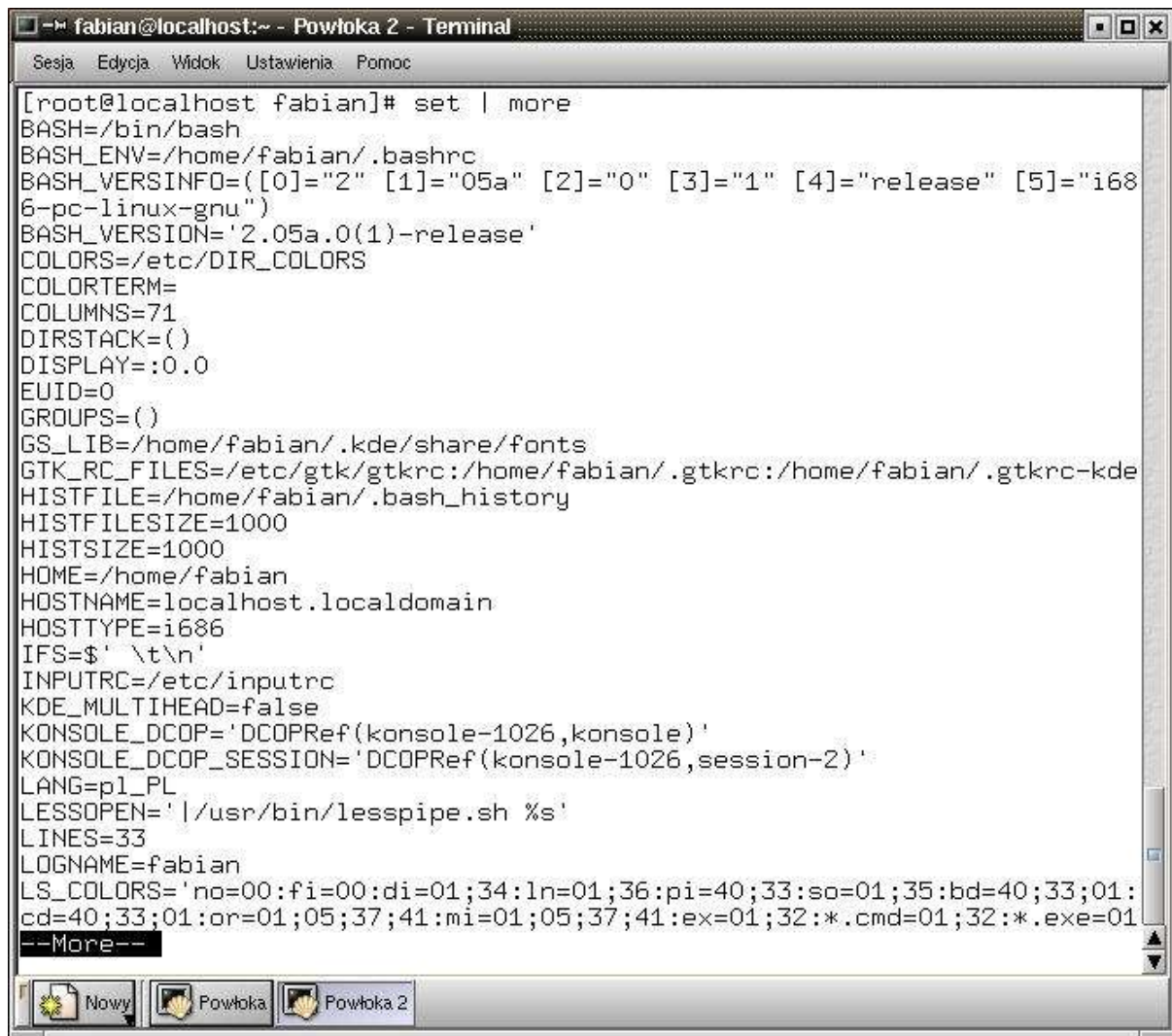
Rodzaj shella dla użytkownika, określony jest w ostatniej linii pliku z hasłami / **etc/passwd**.

Po zalogowaniu się na swoje konto uruchomiona zostaje powłoka logowania zwana inaczej powłoką użytkownika, następnie powłoka sprawdza parametry, obowiązujące wszystkich użytkowników w pliku /**etc/profile**. Potem zostają uruchomione pliki w katalogu użytkownika:

```
.bash_profile  
.bash_login  
.profile
```

Jeżeli w katalogu domowym nie ma tych plików, w większości dystrybucji należy je skopiować z katalogu /**etc/skel/**. Plik **.bash_profile** zawiera polecenia powłoki, zawierające zmienne środowiskowe. Na przykład gdy użytkownik loguje się, shell musi wiedzieć w jakich katalogach znajdują się polecenia i programy Linuksa, określa to zmienna **PATH**. W pliku tym znajduje się polecenie **export**, udostępniające zmienne specjalne w każdej powłoce. Można wyeksportować kilka zmiennych używając jednego polecenia **export**.

Plik **.bash_profile** jest plikiem tekstowym i można go konfigurować według swoich potrzeb, np. dodać katalogi do zmiennej **PATH**, dodać nowe definicje zmiennych. Wprowadzone zmiany uaktywnią się po wylogowaniu się i ponownym zalogowaniu, lub poprzez polecenie **.bash_profile**. Zmienne specjalne w **.bash_profile** są ważniejsze od tych w pliku **.profile**.



```
[root@localhost fabian]# set | more
BASH=/bin/bash
BASH_ENV=/home/fabian/.bashrc
BASH_VERSINFO=( [0]="2" [1]="05a" [2]="0" [3]="1" [4]="release" [5]="i686-pc-linux-gnu" )
BASH_VERSION='2.05a.0(1)-release'
COLORS=/etc/DIR_COLORS
COLORTERM=
COLUMNS=71
DIRSTACK=()
DISPLAY=:0.0
EUID=0
GROUPS=()
GS_LIB=/home/fabian/.kde/share/fonts
GTK_RC_FILES=/etc/gtk/gtkrc:/home/fabian/.gtkrc:/home/fabian/.gtkrc-kde
HISTFILE=/home/fabian/.bash_history
HISTFILESIZE=1000
HISTSIZE=1000
HOME=/home/fabian
HOSTNAME=localhost.localdomain
HOSTTYPE=i686
IFS=$' \t\n'
INPUTRC=/etc/inputrc
KDE_MULTIHEAD=false
KONSOLE_DCOP='DCOPRef(konsole-1026, konsole) '
KONSOLE_DCOP_SESSION='DCOPRef(konsole-1026, session-2) '
LANG=pl_PL
LESSOPEN='|/usr/bin/lesspipe.sh %s'
LINES=33
LOGNAME=fabian
LS_COLORS='no=00:fi=00:di=01;34:ln=01;36:pi=40;33:so=01;35:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05;37;41:ex=01;32:*:cmd=01;32:*:exe=01'
--More--
```

Rys. Zdefiniowane zmienne systemowe – efekt działania polecenia **set**.

Poza zmiennymi systemowymi istnieją też zmienne globalne. **Zmienne globalne** są to zmienne widoczne w każdym podshellu. Aby ustalić nową zmienną lub przypisać inną wartość starej należy użyć polecenia :

```
export EDITOR=mcedit
```

Ważniejsze zmienne to :

BASH=/bin/bash - nazwa powłoki użytkownika

BASH_VERSION=1.14.7(1) - wersja powłoki

COLUMNS=80 - liczba kolumn znaków na ekranie

LINES=25 - liczba linii na ekranie

HOME=/home/fabian - katalog domowy zalogowanego użytkownika

LOGNAME=fabian – login użytkownika

OSTYPE=Linux - typ systemu

PATH=/usr/bin:/sbin:/bin:/usr/sbin - domyślna ścieżka dostępu

PS1=[\u@\h \W]\\$ - ustawienie znaku zachęty

PWD=/home/fabian - aktualny katalog roboczy

SHELL=/bin/bash - nazwa powłoki

USERNAME=fabian - nazwa użytkownika, który jest aktualnie zalogowany w systemie

EDITOR=vim - nazwa domyślnego edytora tekstu

HISTFILE=/home/fabian/.bash_history - plik zawierający historię poleceń

MAIL=/var/spool/mail/\$USER - określa plik, do którego dopisywane są listy przychodzące na konto zalogowanego użytkownika.

Shell posiada pamięć ostatnio wykonywanych poleceń, historię. Aby zobaczyć ostatnio wykonywane polecenia należy użyć polecenia:

history

Narzędzie historii pozwala na odwoływanie się do poleceń poprzednio wykonanych i umieszczenie ich w wierszu poleceń oraz wykonanie. Najprostszym sposobem jest użycie kursora w górę lub w dół. Można również odwołać się do historii za pomocą polecenia ze znakiem **!**, za znakiem wpisując tekst rozpoczynający polecenie lub numer z listy historii. Ilość poleceń do zapamiętania określona jest w pliku **.bash_history**, za pomocą wartości nadanej zmiennej **HISTSIZE**.

Ćwiczenia:

1. Wyświetlić ustawienia środowiska i odnaleźć zmienne odpowiedzialne za:
 - a. login użytkownika
 - b. domyślny shell
 - c. katalog domowy
 - d. lokalizację pliku z historią wydanych poleceń
 - e. ilość zapamiętanych poleceń
 - f. nazwę systemu operacyjnego
2. Posługując się poznanymi poleceniami zapisać plik zawierający historię poleceń do pliku „polecenia”.
3. Wyświetlić tylko zmienne odpowiedzialne za katalog domowy użytkownika.
4. Zmienić domyślny edytor na **Pico** i powłokę logowania na **sh**.
5. Wyświetlić tylko 10 ostatnich poleceń historii używając poznanych poleceń.
6. Wykonać ostatecznie 3 polecenia z historii odwołując się do ich numerów.

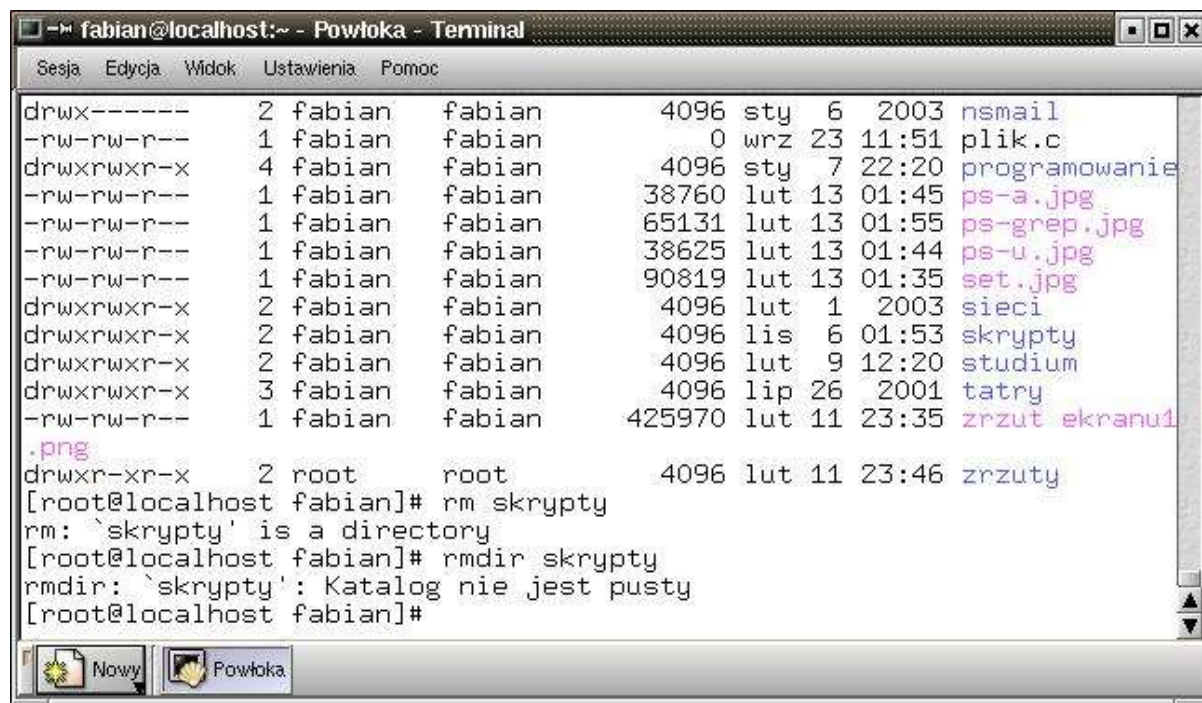
Prawa dostępu do plików i katalogów

W systemie operacyjnym Linux każdy plik i katalog posiada zestaw praw określający, kto ma dostęp do pliku i jakie ma prawa. Każdy plik lub katalog może mieć prawo czytania (read), pisania (write), i wykonywania. Każde z tych praw dostępu reprezentowane jest odpowiednią literą i posiada przypisany odpowiedni parametr cyfrowy.

Znaczenie	Litera	Liczba
Prawo odczytu	r	4
prawo zapisu	w	2
prawo uruchomienia	x	1
brak praw dostępu	-	0

Do każdego pliku lub katalogu możemy wyszczególnić trzy zestawy takich praw:

- prawa właściciela (u - user)
- prawa grupy (g - group)
- prawa pozostałych użytkowników (o - others)



```
fabian@localhost:~ - Powłoka - Terminal
Sesja  Edycja  Widok  Ustawienia  Pomoc
drwx----- 2 fabian fabian 4096 sty 6 2003 nsmail
-rw-rw-r-- 1 fabian fabian 0 wrz 23 11:51 plik.c
drwxrwxr-x 4 fabian fabian 4096 sty 7 22:20 programowanie
-rw-rw-r-- 1 fabian fabian 38760 lut 13 01:45 ps-a.jpg
-rw-rw-r-- 1 fabian fabian 65131 lut 13 01:55 ps-grep.jpg
-rw-rw-r-- 1 fabian fabian 38625 lut 13 01:44 ps-u.jpg
-rw-rw-r-- 1 fabian fabian 90819 lut 13 01:35 set.jpg
drwxrwxr-x 2 fabian fabian 4096 lut 1 2003 sieci
drwxrwxr-x 2 fabian fabian 4096 lis 6 01:53 skrypty
drwxrwxr-x 2 fabian fabian 4096 lut 9 12:20 studium
drwxrwxr-x 3 fabian fabian 4096 lip 26 2001 tatry
-rw-rw-r-- 1 fabian fabian 425970 lut 11 23:35 zrzut_ekranu1
.png
drwxr-xr-x 2 root root 4096 lut 11 23:46 zrzuty
[root@localhost fabian]# rm skrypty
rm: `skrypty' is a directory
[root@localhost fabian]# rmdir skrypty
rmdir: `skrypty': Katalog nie jest pusty
[root@localhost fabian]#
```

Rys. Widoczne jako pierwsze z lewej uprawnienia do każdego pliku po ls -l

Na powyższym zrzucie ekranu widać jako pierwsze z lewej w każdym wierszu

przypisane uprawnienia do każdego pliku i katalogu. Pierwsza litera z lewej określa czy jest to katalog (d) czy dowiązanie (l) czy też zwykły plik (-). Kolejne 3 literki określają uprawnienia dla właściciela pliku (odpowiednik u), kolejne 3 – grupy (g) i ostatnie wszystkich pozostałych (o)

Poprzez dodanie do siebie odpowiednich parametrów, zestaw trzech praw możemy przedstawić za pomocą jednej cyfry, np.:

Prawa	Wartość	Znaczenie
---	0	brak praw
r--	4	prawo do odczytu
rw-	6	prawa do odczytu i zapisu
rwX	7	prawa do odczytu zapisu i uruchomienia
r-x	5	prawa do odczytu i uruchomienia
--x	1	prawo do uruchomienia

`rxrx-rx-x 755`

Powyższy zapis oznacza prawa do odczytu, zapisu i uruchomienia dla właściciela pliku oraz odczytu i uruchomienia dla wszystkich innych użytkowników

Do zmiany praw dostępu do plików w systemie Linux służy polecenie **chmod**. Składnia polecenia przy wykorzystaniu metody cyfrowej wygląda tak:

```
chmod uprawnienia plik
```

np.:

```
chmod 600 /etc/lilo.conf  
chmod 750 /home/fabian
```

lub

```
chmod u+x ~/plik  
chmod g-w /etc/lilo.conf  
chmod o=rwx /home/fabian/udostepnione  
chmod -R ~/dla_wszystkich
```

(-R powoduje zmianę rekursywną uprawnień do katalogu wraz z podkatalogami)

Ćwiczenia:

1. Utworzyć w swoim katalogu domowym 3 katalogi i po 2 pliki w każdym z

nich

2. Używając edytora **mcedit** lub **vi** zapisać w tych plikach dane dotyczące: nazwy systemu operacyjnego, powłoki, katalogu domowego (dane na podstawie ustawień środowiska – polecenie **SET**).

3. Plikom w katalogu pierwszym nadać prawa tylko do odczytu i spróbować je usunąć.

4. Plikom w katalogu drugim odebrać wszystkie prawa i spróbować je odczytać

5. Posługując się parametrami polecenia **chmod** zmienić uprawnienia do katalogu 3 wraz z całą zawartością.

7. Wybrany plikom nadać uprawnienia:

- a. wszystkie dla wszystkich
- b. zapis i odczyt dla właściciela, pozostali tylko odczyt
- c. odczyt dla właściciela, zapis i odczyt dla grupy, brak dla wszystkich
- d. tylko odczyt dla właściciela, grupy i wszystkich pozostałych
- e. uprawnienia dla grupy takie jak dla właściciela

7. Wszystkie wybrane polecenia wykonać na dwa sposoby: przez zapis literowy i ósemkowy

FTP – File Transfer Protocol

FTP jest protokołem sieciowym służącym do przesyłania plików. Większość oprogramowania i plików dostępnych w Internecie zamieszczania jest właśnie na serwerach FTP. Linux oferuje wsparcie dla protokołu FTP i umożliwia wydawanie poleceń kierowanych bezpośrednio do serwera FTP. Umożliwia przesyłanie plików, zmiany ich nazw na serwerach oraz kasowanie i przenoszenie

Sposób transmisji określany jest przez trzy parametry:

1. rodzaj pliku (*ascii, binary*)
2. tryb transferu (*stream*)
3. struktura pliku (*file*)

Otwarcie sesji na serwerze FTP umożliwia program ftp, który wywołuję się poleceniem:

```
ftp host
```

np.:

```
ftp ftp.tu.kielce.pl
```

Po załogowaniu się można zacząć pracę. Duża część serwerów FTP umożliwia anonimowe logowanie się:

```
user: anonymous
```

```
password: anonymous@com.pl
```

Polecenia protokołu FTP :

! [polecenie]	odnosi się do lokalnego komputera
cd [nazwa katalogu]	zmiana katalogu na odległej maszynie.
lcd [nazwa katalogu]	j.w. ale dla lokalnego komputera
ls lub dir [nazwa katalogu [nazwa pliku]	zapisanie info o zawartości katalogu na odległej maszynie do wskazanego pliku. Jeśli brak wskazanego pliku to zawartość wyświetla się na ekranie.
delete nazwa pliku	skasowanie pliku na odległym hoście.

mdelete [lista plików]	skasowanie listy plików.
ascii	ustanowienie rodzaju pliku do transmisji ASCII (.txt).
binary	ustanowienie rodzaju pliku do transmisji binarnej (programy, grafika).
glob	ustanowienie rodzaju przenoszenia nazwy dla <code>mget</code> , <code>mput</code> , <code>mdelete</code> . Jeśli <code>glob</code> jest włączony, to występujące w nazwach <code>*</code> i <code>?</code> traktowane są jako znaki zastępcze, jeśli wyłączony to jako znaki literalne.
open [host]	otwarcie połączenia z wybranym serwerem
get plik [plik]	przekopiowanie pliku do wskazanego pliku na lokalnym komputerze zgodnie z ustalonym wcześniej sposobem transmisji.
put plik [plik1]	j.w ale w kierunku odwrotnym z ew. zmianą nazwy na plik1.
prompt	włączenie lub wyłączenie trybu zapytań przy operacjach na plikach.
? komenda	wyświetlenie info na temat wyświetlonej komendy.
pwd	zwraca nazwę bieżącego katalogu roboczego.
bye, quit	zakończenie programu ftp.

Aliasy

Aliasy są to skrótowe nazwy często używanych poleceń w Linuksie. Polecenie to jest wbudowane w powłokę **Bash**. W systemach uniksowych występują 2 rodzaje dowiązań:

- dowiązania symboliczne
- dowiązania sztywne

Symboliczne jedynie wskazują na dany plik i ich usunięcie nie powoduje usunięcia pliku na który wskazują. Dowiązania sztywne mają taki sam rozmiar jak wskazywane pliki; usunięcie jednego z dowiązań sztywnych nie powoduje usunięcia pliku, jedynie usunięcie ostatniego linku sztywnego skończy się skasowanie oryginalnego pliku

Za pomocą **aliasów** można zastąpić długie polecenia, krótkimi i łatwymi do zapamiętania skrótami.

Aby utworzyć **alias** trzeba wydać polecenie o następującej składni:

```
alias moj_skrót=`polecenie`
```

Na przykład - często używanym poleceniem w celu zamontowania dyskietki jest polecenie:

```
mount /dev/fd0 /mnt/floppy
```

po utworzeniu np. takiego aliasu:

```
alias dys=`mount /dev/fd0 /mnt/floppy`
```

będzie odtąd można używać skrótu **dys** który zamontuje dyskietkę.

W celu zlikwidowania zdefiniowanego aliasu, musisz wydać polecenie:

```
unalias moj_skrót
```

czyli np.:

```
unalias dys
```

Polecenie **alias** bez jakichkolwiek opcji wyświetla listę aktualnie zdefiniowanych skrótów.

Globalne definicje aliasów znajdują się w pliku **/etc/bashrc** i mogą być modyfikowane tylko przez administratora systemu. Użytkownicy, definicje swoich własnych aliasów mogą zapisywać do pliku konfiguracyjnego **.bashrc** w swoim katalogu domowym.

Ćwiczenia:

1. Wydać polecenie *alias* aby sprawdzić zdefiniowanie przez system aliasy
Otworzyć plik *.bashrc* dowolnym edytorem i dopisać aliasy do poleceń:

- a. *startx*
- b. */bin/mouseconfig*
- c. *ls -la | more*
- d. *cd /var/log && ls -l*
- e. *mount /mnt/cdrom*
- f. *mount /mnt/floppy*
- g. *mount -t vfat /dev/fd0 /mnt/floppy*
- h. *umount /mnt/floppy*
- i. *umount /mnt/cdrom*
- j. *shutdown -h now*
- k. *shutdown -r now*
- l. *logout*

2. Przetestować działanie aliasów.

3. Usunąć wszystkie wcześniej utworzone

Dowiązania - linki

Dowiązania to inaczej odnośniki lub łączniki. Dowiązania (linki, odnośniki, łączniki) umożliwiają w systemie Linux nadać plikowi więcej niż jedną nazwę. Można odwoływać się do pliku używając innej nazwy, plik znajdujący się w jednym katalogu może być dostępny w innym katalogu. Podczas instalacji Linuksa automatycznie utworzonych zostaje wiele dowiązań. Dowiązanie przechowuje informację gdzie znajduje się określony plik.

W systemach uniksowych występują 2 rodzaje dowiązań:

- dowiązania symboliczne
- dowiązania sztywne

Symboliczne jedynie wskazują na dany plik i ich usunięcie nie powoduje usunięcia pliku na który wskazują. Mogą też wskazywać katalog. Usunięcie pliku powoduje zaburzenie działania takiego linku – jest on wtedy bezużyteczny (tzw. *broken link*).

Linki symboliczne powstają po wydaniu polecenia :

```
ln -s nazwa_oryginalna_pliku nowa_nazwa_pliku
```

np.:

```
ln -s /opt/OpenOffice/soffice ~/soffice
```

Dowiązania sztywne mają taki sam rozmiar jak wskazywane pliki; usunięcie jednego z dowiązań sztywnych nie powoduje usunięcia pliku, jedynie usunięcie ostatniego linku sztywnego skończy się skasowanie oryginalnego pliku. Utworzenie linku sztywnego powstaje po wydaniu polecenia:

```
ln nazwa_oryginalna nowa_nazwa
```

np.:

```
ln /opt/OpenOffice/sadmin ~/admin
```

Dowiązania sztywne nie powinno się tworzyć między plikami leżącymi na różnych partycjach

ln -b	tworzy kopie zapasowe plików przed ich usunięciem lub nadpisaniem
ln -f	nadpisuje istniejący plik o takiej samej nazwie
ln -i	oczekuje na potwierdzenie przed usunięciem
ln -s	tworzy dowiązania symboliczne zamiast stałych

Ćwiczenia:

1. *Utworzyć w swoim katalogu domowym katalog „linki”.*
2. *Wewnątrz katalogu utworzyć 3 pliki: imie, system, kernel*
3. *Używając edytora vi lub mcedit zapisać w nich informacje adekwatne do tytułów.*
4. *Utworzyć po 2 linki symboliczne do każdego z plików.*
5. *Usunąć jeden z plików na które wskazują dowiązania i sprawdzić działania linku połączonego z tym usuniętym plikiem.*
6. *Sprawdzić prawa dostępu do utworzonych poprzednio linków.*
7. *Utworzyć podobny katalog o nazwie „linki_sz” zawierający kopie plików z katalogu „linki”.*
8. *Utworzyć po 3 linki sztywne do tych plików, sprawdzić ich prawa dostępu.*
9. *Sprawdzić działanie linków sztywnych.*
10. *Spróbować utworzyć dowolny link symboliczny między katalogami /home i /var.*
11. *Sprawdzić działanie dowiązania sztywnego po usunięciu pliku na który dowiązanie wskazywało.*
12. *Usunąć wszystkie stworzone przez siebie dowiązania*

Wyszukiwanie i sortowanie

Często występuje potrzeba odnalezienia konkretnego pliku, lub pliku spełniającego określone kryteria. Polecenie "find" stanowi silne narzędzie do wyszukiwania plików. Składnia tego polecenia to:

find ścieżka wyrażenie

gdzie:

ścieżka - katalog w którym "find" ma rozpocząć wyszukiwanie,

wyrażenie - składa się z OPCJI, TESTÓW oraz AKCJI,

opcje - wpływają na sposób działania polecenia "find".

Najczęściej stosuje się:

- ▣ **follow** - wymusza wchodzenie do katalogów, które są linkami symbolicznymi,
- ▣ **maxdepth POZIOM** - określa poziom katalogów do którego "find" realizował będzie poszukiwania,
- ▣ **mindepth POZIOM** - określa poziom katalogów poniżej którego "find" realizował będzie poszukiwania,
- ▣ **mount** - zawęża poszukiwania tylko do jednego systemu plików.

TESTY - w zależności od wyniku porównania z testem, znaleziony plik będzie poddawany AKCJI bądź będzie ignorowany. Wartości numeryczne mogą być specyfikowane jako "+n" co oznacza większe od n, "-n" oznacza mniejsze od n lub "n" - równe n. Najczęściej wykonywane testy to:

- ▣ **group** - należy do określonej grupy,
- ▣ **links** - posiada odpowiednią liczbę dowiązań,
- ▣ **mtime** - był modyfikowany podaną ilość dni temu,
- ▣ **name** - posiada podaną nazwę,
- ▣ **perm** - posiada prawa dostępu zgodne z podanymi,
- ▣ **size** - ma rozmiar zgodny/większy/mniejszy niż podany,
- ▣ **type** - jest określonego typu (np. "f" dla plików, "d" dla katalogów),
- ▣ **user** - jest własnością określonego użytkownika

AKCJA - domyślną akcją jest wypisanie nazwy pliku. "find"

oferuje dodatkowo m.in. następujące akcje:

- ▣ **exec komenda** ; - wykonanie zewnętrznej komendy. "{}" zastępuje nazwę pliku,
- ▣ **printf format** - wypisanie nazwy pliku odpowiednio sformatowanej,
- ▣ **ls** - wykonanie komendy **ls -dils** na pliku.

Przykłady użycia komendy find:

Poszukiwanie pliku o nazwie "motd:

```
find / -name motd
```

Poszukiwanie podkatalogów w katalogu /ftp należących do użytkownika fabian

```
find /ftp/ -user fabian -type d
```

Kasowanie plików w katalogu domowym, posiadających w nazwie "bak" i nie modyfikowanych od minimum 7 dni:

```
find ~/ -type f -name "*bak*" -mtime +7 -exec rm {} ;
```

Bardzo rozbudowanym narzędziem jest też program grep. Program pozwala przeszukiwać zawartość plików na okoliczność wystąpienia danego ciągu znaków oraz świetnie sprawdza się jako filtr dla innego polecenia. Składnia jest następująca:

```
grep [-civ] wzorzec [plik]
```

np.:

```
ls -l /home/fabian/muzyka/ | grep mp3
```

wyświetli nazwy wszystkich plików o rozszerzeniu mp3 i inne zawierające w nazwie ciąg znaków "mp3"

```
ls -l /home/fabian/muzyka/ | grep -c mp3
```

zliczy ilość plików mp3 (ilość wierszy zawierających ciąg znaków mp3)

```
cat /etc/passwd | grep -v root
```

wyświetli wszystkie linie (użytkowników) z pliku /etc/passwd poza rootem

```
ps -aux | grep -v root
```

wyświetli wszystkie procesy poza procesami roota

```

root@hydra:/home/fabian# ls -l /var/log/packages/ | grep gcc
-rw-r--r--  1 root    root      9737 2004-03-05 13:19 gcc-3.2.3-i486-2
-rw-r--r--  1 root    root     8634 2004-03-05 13:19 gcc-g++-3.2.3-i486-1
root@hydra:/home/fabian# ps -aux | grep cron
root    1284  0.0  0.1  1500  592 ?        S    16:43   0:00 /usr/sbin/crond -11
0
root@hydra:/home/fabian# cat /etc/lilo.conf | grep image
image = /boot/vmlinuz
image = /boot/vmlinuz-2.2.22-6
#image = /boot/vmlinuz-2.4.18-3
image = /boot/bzImage
root@hydra:/home/fabian# ps -fa | grep -v root
UID      PID  PPID  C  STIME TTY          TIME CMD
fabian   1475  1450  0  17:02 tty2        00:00:00 /bin/sh /usr/X11R6/bin/startx
fabian   1486  1475  0  17:02 tty2        00:00:00 xinit /usr/X11R6/lib/X11/xinit/xin
itrc --
fabian   1491  1486  0  17:02 tty2        00:00:00 /bin/sh /usr/X11R6/lib/X11/xinit/x
initrc
fabian   1493  1491  0  17:02 tty2        00:00:00 /bin/sh /opt/kde/bin/startkde
fabian   1528  1493  0  17:02 tty2        00:00:00 kwrapper ksmsserver
root@hydra:/home/fabian# █

```

Rys. Różne sposoby użycia **grep**.

Najczęściej stosowane parametry polecenia **grep**

-c	podaje liczbę wierszy w których znaleziono pasujący wzorzec
-i	pomija różnicę w wielkości liter
-v	wyświetla wiersze, w których nie ma podanego wzorca
-plik	nazwa pliku, w którym następuje przeszukanie

Archiwizacja plików i katalogów

W systemie Linux podobnie jak w innych systemach Unixowych jest wiele narzędzi służących do archiwizacji i kopii zapasowych plików i katalogów. Najpopularniejsze z nich to:

- tar
- cpio
- dd
- dump

W tym podręczniku omówione zostanie tylko pierwsze z wymienionych. Pozostałe 3 przedstawione zostanie w dalszej części kursu.

Program tar jest używany zarówno do przygotowywania kopii zapasowych danych jak i do przygotowywania archiwów zawierających pliki programów do dystrybucji w Internecie. Znajomość tego programu i jego parametrów jest niezbędna przy instalacji programów ze źródeł, gdyż pliki tych programów są prawie zawsze rozprowadzane w postaci archiwum **tar**. Składnie polecenia tar jest następująca:

```
tar -parametr plik
```

```
np.: tar -zxvf archwium.tar.gz
```

rozpakowuje spakowane programem **gzip** archiwum do katalogu archiwum, natomiast

```
tar -cf /root/archiwum.tar /home/pliki/
```

utworzy archiwum o nazwie archiwum.tar zawierające pliki z katalogu /home/pliki.

Tworzenie archiwum

Składnia polecenia jest następująca:

```
tar -parametry nazwa_tworzonego_archiwum plik1 [plik2 plik3]
```

```
np.:
```

```
tar -cvf arch.tar /var/log/
```

```
tar -cvf /root/arch.tar ~/plik1 ~/plik2 ~/plik3
```

```
tar -cvf /dev/hda1 /home/fabian /root
```

Utworzone już archiwum można dodatkowo skompresować programami **gzip**

lub **bzip2**, które jako argument przyjmują nazwę archiwum, np.:

```
bzip2 archiwum.tar  
gzip archiwum.tar
```

Co w rezultacie daje archiwum **tar.gz** lub **tar.bz2**:

```
archiwum tar.gz  
archiwum.tar.bz2
```

Dekompresja archiwum

Składnia polecenia jest następująca:

```
tar -paramentry archiwum.tar.gz
```

np.:

```
tar -xvf arch.tar  
tar -xzvf arch.tar.gz  
tar -xlvf arch.tar.b2
```

Można też tylko użyć programu **bzip2** lub **gzip** do dekompresji archiwum:

```
bunzip2 archiwum.tar.bz2  
gunzip archiwum.tar.gz
```

co spowoduje iż plik archiwum będzie miał postać:

```
archiwum.tar
```

Najczęściej używane parametry programu tar

-c	Tworzy nowe archiwum
-r	Dodaje nowe pliki do istniejącego archiwum
-u	Upgrade zmodyfikowane pliki do archiwum
-t	Wyświetla zawartość (listę plików) archiwum
-x	De kompresuje pliki z archiwum
--delete	Usuwa wskazane pliki z archiwum
-z	Rozpakowuje archiwa spakowane przez gzip

-l	Rozpakowuje archiwa spakowane przez bzip2
-v	Wyświetla listę przetwarzanych plików podczas kompresji lub dekompresji

Ćwiczenia:

- 1.** Wybrane pliki i katalogi ze swojego katalogu domowego spakować programem *tar*
- 2.** Przetestować działanie parametrów *-v -z -l*.
- 3.** Utworzyć kolejne archiwa nie stosując parametrów *-z -l* i spakować je programami **gzip** i **bzip2**
- 4.** Posługując się programem *Midnight Commander* spakować i rozpakować wybrane pliki i katalogi
- 5.** Skopiowane z płyty CD archiwa rozpakować stosując poznanie parametry i programy

„Spolszczenie” Linuksa

System Linux posiada uniwersalny mechanizm lokalizacji różnych wersji językowych tzw. *locales*. *Locales* jest to zbiór danych i programów wspomagających tłumaczenie komunikatów wypisywanych przez programy. W katalogu `/usr/share/locale/` znajdują się podkatalogi z plikami dla różnych wersji językowych, każdy plik odpowiada za komunikację z jednym programem. Niektóre aplikacje dostosowane są do komunikowania się z użytkownikiem w języku polskim. Pliki dla języka polskiego znajdują się w katalogu:

`/usr/share/locale/pl/LC_MESSAGES/` lub `/usr/share/locale/pl_PL/LC_MESSAGES/`.

Aby poinformować system w jakim języku ma się komunikować z użytkownikiem, należy dokonać odpowiedniej konfiguracji **locale** w pliku:

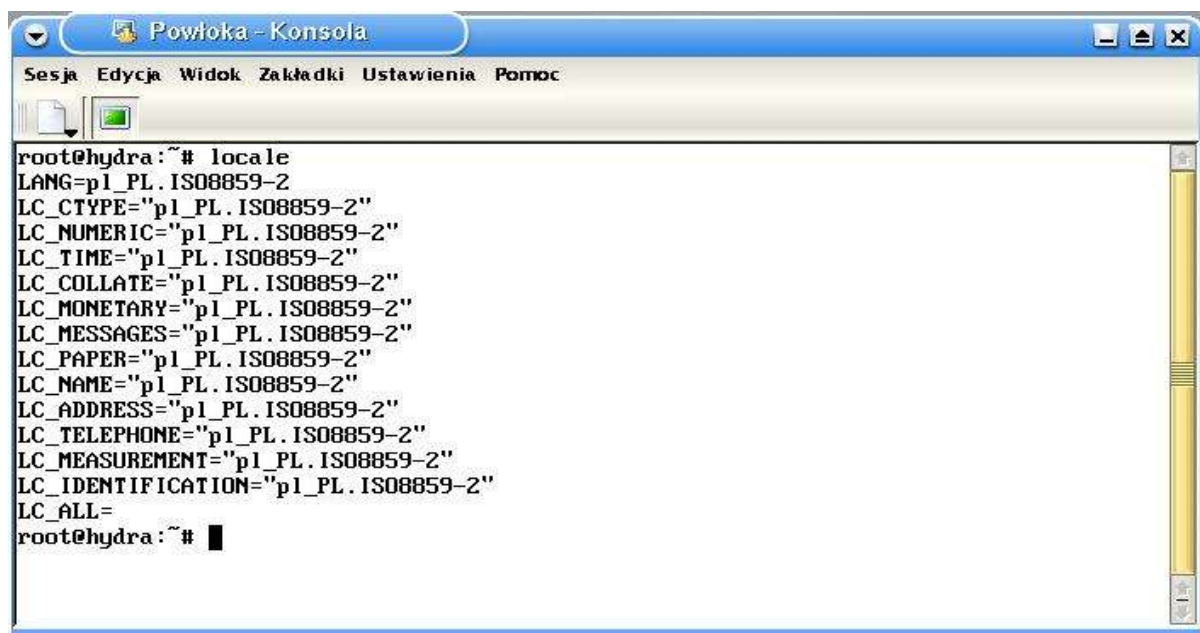
`/etc/sysconfig/i18n`.

W pliku tym ustawić można język komunikatów systemu i wszystkie specyficzne dla danego kraju, formaty zapisu: daty, czasu, itp.

Aby sprawdzić jaka jest konfiguracja **locale** należy wpisać w konsoli :

locale

Efekt powinien być taki jak na poniższej ilustracji.



```
root@hydra:~# locale
LANG=p1_PL.ISO8859-2
LC_CTYPE="p1_PL.ISO8859-2"
LC_NUMERIC="p1_PL.ISO8859-2"
LC_TIME="p1_PL.ISO8859-2"
LC_COLLATE="p1_PL.ISO8859-2"
LC_MONETARY="p1_PL.ISO8859-2"
LC_MESSAGES="p1_PL.ISO8859-2"
LC_PAPER="p1_PL.ISO8859-2"
LC_NAME="p1_PL.ISO8859-2"
LC_ADDRESS="p1_PL.ISO8859-2"
LC_TELEPHONE="p1_PL.ISO8859-2"
LC_MEASUREMENT="p1_PL.ISO8859-2"
LC_IDENTIFICATION="p1_PL.ISO8859-2"
LC_ALL=
root@hydra:~#
```

Aby mieć w konsoli znakowej polskie czcionki i pisać po polsku, należy mieć zainstalowany pakiet **console tools**. Aby to sprawdzić należy wpisać:

- w Red Hat Linux: `rpm -q console-tools`

- w Slackware Linux: `ls -l /var/log/packages | grep console-tools`

W katalogu `/etc/sysconfig/console/` powinny znajdować się czcionki typu iso2. Następnie w pliku `/etc/sysconfig/i18n` dopisać należy następujące wiersze:

```
SYSFONT=lat2u-16 UNIMAP=lat2u  
SYSFONTACM=iso02
```

a pliku `/etc/sysconfig/keyboard` musi być:

```
KEYTABLE=pl
```

Na koniec należy załadować polską klawiaturę poleceniem:

```
loadkeys pl
```

Ćwiczenia :

- 1. Sprawdzić ustawienia pliku locale.*
- 2. Załadować klawiaturę o polskim układzie znaków.*
- 3. Dokonać polonizacji systemu i sprawdzić czy wszystkie ustawienia działają.*

Przekierowania i potoki

Większość komend wyniki działania (wyjście) kieruje na ekran a dane pobiera z klawiatury (wejście), ale w Linuksie jest możliwe skierowanie rezultatów działania do pliku lub wprowadzanie danych z pliku. Na przykład polecenie:

```
ls
```

wyświetla wyniki na ekranie; aby skierować wyjście tego polecenia do pliku możemy napisać tak :

```
ls > nazwa_pliku.
```

To znaczy że wysyłamy rezultat działania polecenia ls do pliku (nazwa_pliku). To są trzy główne symbole przekierowania >, >>, <

Symbol przekierowania >

```
polecenie > nazwa_pliku
```

Przekierowuje rezultat działania polecenie-Linuksa do pliku (nazwa_pliku). Jeżeli plik już istnieje, to jest nadpisywany w przeciwnym wypadku tworzony jest nowy. Przykład : aby wysłać rezultat działania polecenie ls do pliku musimy napisać:

```
ls > mojplik
```

Teraz jeżeli plik '**mojplik**' istnieje w aktualnym katalogu zostanie nadpisany bez wyświetlania żadnych ostrzeżeń. (A co jeżeli chcemy wysłać wyjście do pliku, który już istnieje i jednocześnie chcemy zachować informacje już istniejące w danym pliku? Przeczytaj o następnym symbolu przekierowania)

Symbol przekierowania >>

```
polecenie >> nazwa_pliku
```

Przekierowuje rezultat działania polecenie Linuksa zapisuje na końcu pliku (nazwa_pliku). Jeżeli plik istnieje , zostanie otwarty i nowe informacje/dane zostaną zapisane na końcu tego pliku, bez utraty poprzednich informacji danych. A jeżeli plik nie istnieje, zostanie stworzony nowy. Przykład : przekierujemy wyjście danych polecenia date do już istniejącego pliku

```
date >> mojplik
```

Symbol przekierowania <

```
polecenie < nazwa_pliku
```

Wejściem (wprowadzeniem danych) polecenie-Linuxa jest plik (nazwa_pliku).
Przykład : aby określić źródło danych dla polecenia cat wpisujemy

```
cat < myfiles
```

Potoki

Potok to sposób połączenia wyjścia jednego programu z wejściem innego programu bez pliku pośredniego (tymczasowego).

Potok jest miejscem tymczasowego przechowania rezultatu jednego polecenia które podane jest jako wejście dla drugiego polecenie. Potoki używane są do uruchamiania przynajmniej dwóch poleceń w tym samym wierszu poleceń.

Składnia: polecenie1 | polecenie2

Polecenie z użyciem potoku	Znaczenie użytego potoku
<pre>\$ ls more</pre>	Tutaj wyjście polecenia ls jest podane jako wejście dla komendy more Wynik jest wyświetlony częściami podzielony na ekrany.
<pre>\$ who sort</pre>	Tutaj wyjście komendy who podane jest na wejście komendy sort Efektem jest wyświetlenie posortowanej listy użytkowników
<pre>\$ who wc -l</pre>	Tutaj wyjście komendy who jest podane jako wejście komendy wc Zwrócona zostanie liczba użytkowników aktualnie zalogowanych do systemu
<pre>\$ ls -l wc -l</pre>	Wyjście komendy ls podane jako wejście dla komendy wc . Wyświetli liczbę plików w aktualnym katalogu.
<pre>\$ who grep uzytkownik</pre>	Tutaj wyjście polecenia who podane jest na wejście polecenia grep Zostanie wyświetlona linia z nazwa podanego użytkownika jeżeli jest on zalogowany do systemu lub nic nie zostanie wyświetlone

Filtry

Jeżeli polecenie Linuksa przyjmuje dane wejściowe ze standardowego wejścia i przekształca je produkując wynik na standardowym wyjściu nazwane jest filtrem. Na przykład : Przypuśćmy że mamy plik nazwany 'hotel.txt' zawierający 100 linii danych, z pliku 'hotel.txt' chcemy pobrać dane od linii numeru 20 do linii numer 30 i zachować rezultat w pliku nazwanym 'hlist' więc wydajemy polecenie

```
tail +20 < hotel.txt | head -n30 >hlist
```

Tutaj **head** jest filtrem który bierze wejście z polecenia **tail** (polecenie **tail** zaczyna wyświetlać dane od linii numer 20 podanego pliku, tutaj hotel.txt) i posyła pobrane linie do polecenia **head**, którego wyjście jest

Instalacja pakietów RPM

Programy dla Linuksa są rozprowadzane najczęściej w postaci gotowych pakietów binarnych (rpm, deb, pkg, tgz) lub jako źródła do kompilacji w postaci archiwum tar skompresowanym programem **gzip** lub **bzip2**, zdarzają się też programy posiadające indywidualny program instalacyjny.

RPM - RedHat Package Manager to format wykorzystywany jest w większości dystrybucji Linuksa, między innymi Red Hat, Mandrake, Turbo Linux, SuSE, Aurox, PLD i wiele innych. Pakiety oprogramowania dzielą się na: binarne, które zostały już wcześniej skompilowane i od razu nadają się do uruchomienia, oraz źródłowe, które sami musimy skompilować, by potem móc odpowiednio wygenerowane programy zainstalować na dysku. Najprostszą i najłatwiejszą metodą instalacji jest skorzystanie z pakietów binarnych **rpm**, **deb (Debian Linux)** lub **tgz (Slackware Linux)**

Menadżer pakietów **rpm** służy do instalowania, usuwania, uaktualniania, przepytania, testowania i tworzenia pakietów. Typowy pakiet utworzony za pomocą **rpm** składa się z zarchiwizowanego pliku oraz szeregu przypisanych do niego informacji. Podstawowe informacje to nazwa programu, numer wersji, opis.

Nazwa pakietu składa się: nazwy pakietu, typu i daty oraz oznaczenia typu.

Można spotkać następujące typy pakietów:

- zakończone **.i386.rpm** przeznaczone są dla procesorów 386 lub nowszych
- **.i586.rpm** dla procesorów Pentium i lepszych
- **.noarch.rpm** do zastosowania niezależnie od typu procesora
- **.src.rpm** pakiet zawiera źródła programu, do kompilacji.

Instalacja pakietu dokonuje się po wydaniu polecenia:

```
rpm -i nazwa_pakietu.rpm
```

Dodanie opcji **-v** podaje nazwę instalowanego aktualnie pakietu, co w przypadku instalacji tylko jednego jest zbędne, natomiast **-h** za pomocą znaczka **#** wskazuje nam postęp instalacji. Opcje **-v** i **-h** powinny być wydawane razem z **-i** w postaci:

```
rpm -ivh nazwa_pakietu.rpm
```

Upgrade zainstalowanego pakietu:

```
rpm -U nazwa_pakietu.rpm
```

Można dodać także parametry **-v -h** które działają analogicznie jak w przypadku normalnej instalacji, np.:

```
rpm -Uvh lilo-2.23-i386.rpm
```

Aby usunąć pakiet:

```
rpm -e nazwa_pakietu.rpm
```

co czasem nie wystarcza, gdyż dana paczka ma powiązania z innymi i program **rpm** nie może usunąć paczki dopóki istnieją zależności z innymi. Pomocne jest wtedy dodanie parametru **-nodeps** (po 2-óch minusach), który usuwa pakiet bez szukania jego zależności.

```
rpm -e -nodeps nazwa_pakietu.rpm
```

Parametr **-nodeps** można równie dobrze stosować do instalacji bez szukania zależności.

Aby dowiedzieć się czy pakiet jest zainstalowany można użyć parametrów: **-q, -a, -i**, np.:

```
rpm -qi nazwa_pakietu.rpm
```

wyświetli szczegółowe dane na temat zainstalowanego pakietu, a

```
rm -qa nazwa_pakietu.rpm
```

wyświetli nazwy wszystkich zainstalowanych pakietów.

Zestawienie najczęstszych opcji programu rpm:

-i	Instaluje pakiet
-v	wyświetla na ekranie informacje o działaniu rpm
-h	wyświetla postęp instalacji w postaci znaków #, po skończeniu instalacji widać 50 takich znaków.
-q	sprawdza czy pakiet jest zainstalowany
-qs	wyświetla bieżący status plików wchodzących w skład pakietu
-ql	pokazuje listę wszystkich plików należących do pakietu.
-U	uaktualnia pakiet
-e	usuwa wybrany pakiet

<code>--percent</code>	informuje ile procent pakietu zostało zainstalowane
<code>--test</code>	sprawdzenie czy instalacja pakietu nie spowoduje konfliktów w systemie (symulacja instalacji)
<code>--nodeps</code>	nie wykonuje testów zależności przed instalacją
<code>--replacefiles</code>	pozwała zastępować istniejące pliki, plikami z instalowanego pakietu
<code>--replacepkgs</code>	pozwała na zastępowanie już zainstalowanych pakietów, nowymi pakietami instalowanymi.
<code>--oldpackage</code>	pozwole na zastąpienie starszej wersji pakietu wersją nowszą
<code>--force</code>	wymusza bezwzględną instalację pakietu.

Ćwiczenia:

- 1.** *Skopiować do swojego katalogu domowego wskazane pakiety .rpm*
- 2.** *Przed zainstalowaniem sprawdzić obecność tych pakietów w systemie*
- 3.** *Zainstalować je stosując po kolei różne opcje i parametry*
- 4.** *Zaktualizować wybrane pakiety*
- 5.** *Usunąć pakiety zainstalowane stosując opcje `-e` i `--nodeps`.*

Instalacja programów ze źródeł

Programy dla Linuksa są rozprowadzane w postaci gotowych pakietów binarnych (rpm, deb) lub jako źródła do kompilacji w postaci archiwum tar skompresowanym programem **gzip** lub **bzip2**. Metoda instalacji ze źródłem wymaga trochę większych umiejętności od użytkownika. Bardzo często wyskakują komunikaty o błędzie, ale nie z winy programistów lecz z winy użytkowników, którzy zazwyczaj nie posiadają odpowiednich bibliotek oraz nie czytają zbyt dokładnie dokumentacji dostarczanej wraz z programem. Pierwszym krokiem jest rozpakowanie archiwum zawierającego pliki programu. Aby to zrobić należy wydać polecenie:

```
tar -zxvf nazwa_archiwum.tar.gz
```

lub w przypadku archiwum **bz2**:

```
tar -xlvf nazwa_archiwum.tar.bz2
```

Do rozpakowania można użyć też **Midnight Commandera**, w którym po naciśnięciu **F2** znajdziemy odpowiednie opcje stosowne do każdego programu archiwizującego.

Po rozpakowaniu należy wejść do powstałego katalogu o takiej samej nazwie jak archiwum (tylko bez `.tar.gz` lub `.tar.bz2` na końcu nazwy), odszukać pliki **README** i **INSTALL** i przeczytać. Pliki te mogą zawierać dodatkowe wskazówki co do instalacji i konfiguracji programu. Standardowa instalacja rozpoczyna się od wydania polecenia:

```
./configure
```

którego zadaniem jest wstępna konfiguracja, sprawdzenie wszystkich składników potrzebnych do instalacji programu. Często skrypt **configure** używany jest z opcją `--prefix` zmieniającą domyślny katalog do instalacji plików programu. Na przykład polecenie:

```
./configure --prefix=/usr,
```

ustala że pliki wykonywalne programu zainstalowane zostaną w katalogu `/usr/bin`, biblioteki w `/usr/lib` itd. Po wykonaniu `./configure` powinny zostać utworzone pliki *Makefile*, w przypadku niepowodzenia wstępnej konfiguracji, zostaną wyświetlone odpowiednie komunikaty, a więcej szczegółów szukać należy w pliku *config.log*. Jeśli na tym etapie nie pojawiły się żadne błędy można przejść do właściwej kompilacji programu przez wydanie polecenia:

make

Program **make** posilkuje się domyślnym kompilatorem (zazwyczaj jest to jakaś wersja **gcc**) odczytuje kolejne polecenia z pliku **Makefile** i wykonuje je. Czas kompilacji uzależniony jest od wielkości programu, możliwości komputera. Również na tym etapie nie powinno być żadnych błędów.

Ostateczna instalacja skompilowanego programu następuje po wydaniu polecenia:

make install

co trzeba zrobić będąc zalogowanym jako root. Pliki programu przeniesione zostają do odpowiednich katalogów a pliki wykonywalny lokują się z reguły w jakimś katalogu **bin**.

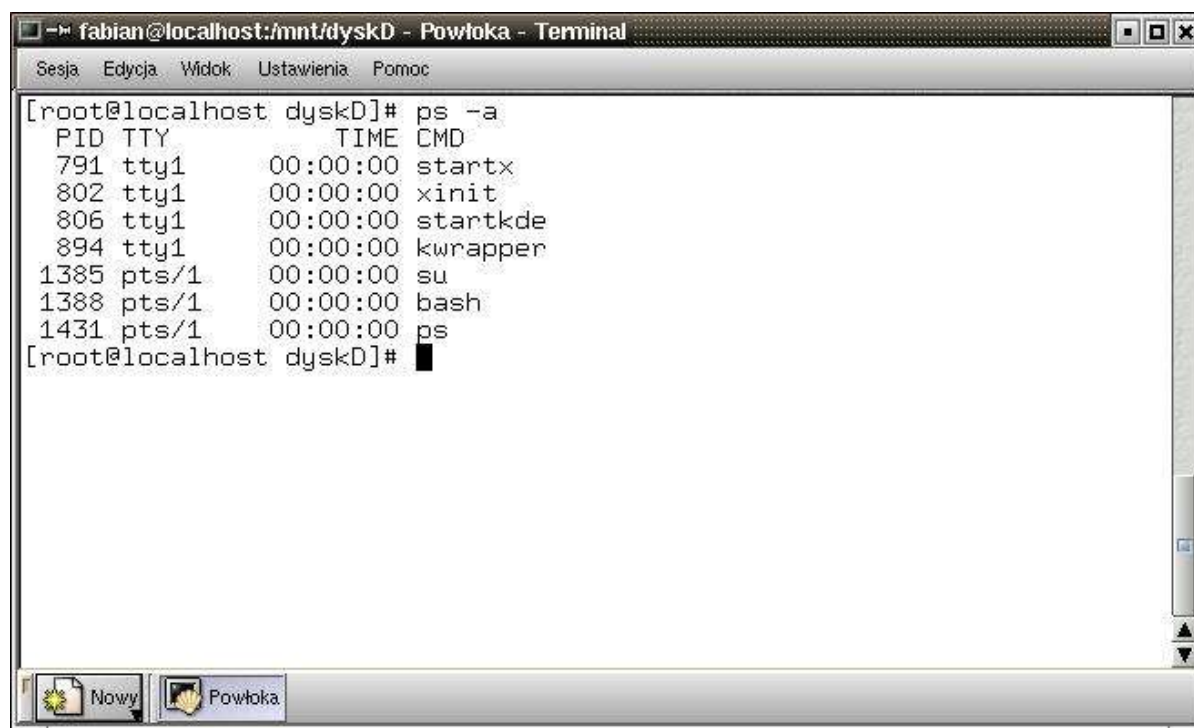
Bardzo często zdarza się, że pojawia się błąd, że plik jakistam.h nie znaleziony. Należy wtedy sprawdzić czy na pewno są zainstalowane pliki nagłówkowe kernela oraz jego źródła! Jeśli to nie pomoże należy jeszcze raz przeczytać dostarczone z programem opisy instalacji, w ostateczności można napisać do autora programu.

Ćwiczenia:

1. Skopiować z płyty CD następujące programy:
 - a. Open Office 1.0
 - b. Midnight Commander
2. Rozpakować archiwa przez program **tar**.
3. Zapoznać się z Plikami **README** i **INSTALL** instalowanych programów
4. Dokonać konfiguracji i zainstalować programy (pakiet Open Office w katalogu **/usr** lub **/opt**)
5. Utworzyć linki do plików wykonywalnych zainstalowanych programów.
6. Uruchomić programy i przetestować ich działanie.

Procesy

Proces jest rodzajem programu lub zadania działającego w naszym komputerze. Dla przykładu `ls -lR`, jest poleceniem lub inaczej, żądaniem wyświetlenia listy plików w naszym aktualnym katalogu wraz ze wszystkimi podkatalogami. To jest proces. Proces jest programem (poleceniem wydanym przez użytkownika) wykonującym jakąś pracę. W Linuksie kiedy startuje proces, nadawany mu jest numer (zwany PID lub process-id), PID może mieć wartość od 0 do 65535.



```
[root@localhost dyskD]# ps -a
PID TTY          TIME CMD
 791 tty1         00:00:00 startx
 802 tty1         00:00:00 xinit
 806 tty1         00:00:00 startkde
 894 tty1         00:00:00 kwrapper
1385 pts/1       00:00:00 su
1388 pts/1       00:00:00 bash
1431 pts/1       00:00:00 ps
[root@localhost dyskD]#
```

rys. Komenda `ps -a` wyświetlająca wszystkie procesy innych użytkowników

Linux jest wieloużytkownikowym, wielozadaniowym systemem. To znaczy że jeżeli potrzeba, można uruchomić co najmniej dwa procesy równocześnie. Na przykład, aby dowiedzieć się jak wiele plików mamy w naszym systemie wystarczy wydać następujące polecenie

```
ls / -R | wc -l
```

To zadanie zajmie trochę czasu przeszukując nasz system. Można uruchomić to polecenie w tle jednocześnie wydając poniższą komendę:

```
ls / -R | wc -l &
```

Znak `'&'` na końcu polecenia przekazuje dla powłoki (shell), aby

polecenie zostało uruchomione (**ls / -R | wc -l**) w tle przyjmując natychmiast następne zadanie. Uruchomiana w przykładownie komenda jest procesem a numer wyświetlony przez powłokę jest identyfikatorem procesu (PID), ten PID może być użyty do odwołania do uruchomionego procesu. W tym samym czasie może być uruchomionych wiele procesów na kilku konsolach jednocześnie. Np.: na 1-szej kompilować jądro, na drugiej kopiować jakiś program z serwera FTP a na kolejnej uruchomić X-Widow i oglądać film na DVD.

```

fabian@localhost:/mnt/dyskD - Powłoka - Terminal
Sesja  Edycja  Widok  Ustawienia  Pomoc
[root@localhost dyskD]# ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1385  0.0  0.1  2308   972 pts/1    S    01:41   0:00 su -m
root      1388  0.0  0.2  2540  1356 pts/1    S    01:41   0:00 bash
root      1429  0.0  0.1  2672   756 pts/1    R    01:44   0:00 ps -u
[root@localhost dyskD]#

```

Rys. Wyświetlenie nazwy użytkownika i czasu startu.

Polecenia Linuksa związane z procesami

Użycie komendy	Przykład	Do czego służy
ps	\$ ps	Wyświetla aktualnie uruchomione procesy
ps -a	\$ ps -a	Wszystkie procesy pozostałych użytkowników
ps -u	\$ ps -u	format użytkownika - podaje nazwę użytkownika i czas startu
ps -x	\$ ps -x	pokaż procesy bez kontrolującego terminala
kill {PID}	\$ kill 1012	Zatrzymanie procesu zwane też zabiciem procesu

ps -ag	\$ ps -ag	Wyświetla informacje o wszystkich uruchomionych procesach
kill 0	\$ kill 0	Zatrzymuje wszystkie procesy poza powłoką zalogowanego użytkownika
komenda &	\$ ls / -R wc -l &	Przeniesienie procesu w tło (Używając &, przenosi konkretną komendą lub program w tło)
jobs	\$ jobs	listuje aktywne programy
fg nr_procesu	\$ fg 1400	ustawienie procesu na front
bg nr_procesu	\$ bg 1544	przeniesienie procesu w tło

Użytkownik może zabijać tylko te procesy które zostały stworzone przez niego. Administrator może zabić niemal 95-98% procesów. Lecz niektóre procesy nie mogą zostać zabite, takie jak procesy VDU.

Narzędzia dyskowe: **fsck** i **fdformat**

W czasie ładowania systemu Linux uruchamiany jest program **fsck**, zadaniem jego jest wychwycenie uszkodzonych plików i ich ewentualne naprawienie. W zależności od typu systemu plików, program uruchamia odpowiednie polecenia do przeprowadzenia sprawdzania i naprawy. Część danych normalnie przechowywanych na dysku jest okresowo przechowywana w pamięci komputera, gdy system nie będzie miał możliwości zapisania tych danych na dysku, system plików może zostać uszkodzony.

W większości przypadków uszkodzenie systemu plików następuje gdy:

- praca systemu została nagle przerwana w wyniku zaniku lub spadku napięcia w sieci.
- wyłączenie sytemu nastąpiło przyciskiem reset lub wyłącznikiem zasilania .

Na ogół **fsck** potrafi automatycznie naprawić wszystkie błędy, bez angażowania użytkownika i system zostanie naprawiony oraz pomyślnie załadowany.

Gdyby okazało się jednak że system plików jest poważnie uszkodzony, automatyczny proces naprawy zostanie zatrzymany i zostaje uruchomiony *shell* (powłoka), dzięki czemu jako *root* będzie możliwe ręczne uruchomienie **fsck** i naprawa systemu. Pojawi się taki oto komunikat:

```
*** An error occured during the filesystem check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for mainteance
(or type Control-D for normal startup):
```

Należy podać teraz hasło `root`a` i nacisnąć **[ENTER]**. System zgłosi się następującym komunikatem:

```
Repair filesystem #
```

Jeżeli przedtem na ekranie pojawiła się informacja jaki system plików jest uszkodzony np: `hda3` w konsoli należy wpisać:

```
fsck -r /dev/hda3
```

W przypadku gdy nie wiadomo który system plików został uszkodzony pozostaje sprawdzić każdy system plików poleceniem:

```
fsck -A -V ; echo == $? ==
```

Podobnie jak w pierwszym przypadku system zadaje pytania czy ma daną czynność naprawiającą wykonać. Jeżeli za bardzo nie wiemy o co chodzi z tymi pytaniami, najlepiej naciskać po prostu **y** (yes).

fsck w komunikatach wyświetlanych między znakami równości, będzie wyświetlał cyfry. Jeżeli wartość tej cyfry będzie mniejsza od 4, to oznacza że wszystko jest w porządku. Poniżej znaczenie poszczególnych cyfr:

0	brak błędów.
1	błędy zostały naprawione
2	system powinien zostać przeładowany
4	błędy nie zostały naprawione
8	błąd operacyjny (<i>operational error</i>)
16	błąd użycia (składni)
128	błąd wspólnej biblioteki

Opis wszystkich parametrów znajduje się w podręczniku man

Innym użytecznym narzędziem dyskowym jest **fdformat** który to program służy do formatowania dyskietek. Składnia jest prosta:

```
fdformat [parametry] urządzenie
```

```
fdformat /dev/fd0
```

Niskopoziomowy format dyskietki

Opcja **-n** - wyłącza weryfikację sformatowanej dyskietki.

Dodatek A: Indeks podstawowych poleceń systemu Linux.

cd ścieżka_dostępu

Opis : zmiana bieżącego katalogu

touch [-acm] [MMDDhhmm] lista_plików

Opis : modyfikacja daty i czasu utworzonego pliku, tworzenie nowego pliku

Opcje:

-a - zmienia tylko czas ostatniego dostępu do pliku

-c - jeśli plik nie istnieje nie zostanie utworzony

-m - zmienia czas ostatniej modyfikacji pliku

MMDDhhmm - czas w postaci : miesiąc dzień godzina minuta (np.: 10122122)

rm [-fr] nazwa_pliku

Opis : kasowanie pliku

Opcje :

-f - pomija konieczność potwierdzenia pytania o usuwanie plików

-r - usuwa zawartość katalogu wraz z podkatalogami

cp [-flpRs] plik plik_docelowy_lub_katalog

Opis : kopiowanie plików i katalogów

Opcje :

-f - powoduje, że program nie uprzedza o nadpisaniu pliku docelowego o tej samej nazwie

-l - tworzy dowiązanie zamiast kopiować plik

-p - opcja ta powoduje, że kopiowane pliki zachowują w miarę możliwości swoje atrybuty

-R - kopiuje zawartość katalogów i podkatalogów

-s - tworzy dowiązanie symboliczne zamiast kopiować plik

mv [-fu] plik plik_docelowy_lub_katalog

Opis : zmiana nazwy lub przenoszenie pliku i katalogu

Opcje :

-f - powoduje, że program nie uprzedza o nadpisaniu pliku docelowego o tej samej nazwie

-u - przenosi pliki tylko wówczas, gdy istniejące pliki docelowe są starsze

chmod [-R][ugoa][+ -=][rwx] plik_lub_katalog

Opis : zmiana atrybutów pliku

Opcje :

-R - zmiana praw dla plików i katalogów wraz z podkatalogami

-u (user) - właściciel pliku lub katalogu
-g (group) - grupa, do której należy dany plik lub katalog
-o (other) - wszyscy pozostali
+ - dodaje prawa
- - odbiera prawa
= - sprawia, że plik lub katalog będzie miał prawa tylko takie, jakie zostaną podane po tej opcji
-r - prawo czytania danych
-w - prawo zapisu (edycji)
-x - prawo uruchamiania

chown [-Rf] właściciel[:grupa] plik_lub_katalog

Opis : zmiana właściciela pliku lub katalogu

Opcje :

-R - zmiana właściciela i/lub grupy plików i katalogów wraz z podkatalogami
-f - zmienia właściciela i/lub grupę tylko tym plikom, które należą do podanego właściciela i/lub grupy

mkdir [-mp] katalog

Opis : tworzenie katalogów

Opcje :

-m=prawa - nadaje tworzonym katalogom podane prawa
-p - nie ostrzega, jeśli katalog o podanej nazwie już istnieje

rmdir [-p] katalog

Opis : kasowanie katalogów

Opcje :

-p - usuwa katalog wraz z katalogami nadrzędnymi

ls [-al] plik_lub_katalog

Opis : wyświetla informacje o plikach i katalogach

Opcje :

-a - wyświetla wszystkie pliki w katalogu, wraz z plikami ukrytymi
-l - listuje w długim formacie pełną informację o plikach i katalogach

ln [-dfs] plik_lub-katalog link

Opis : tworzenie dowiązań

Opcje :

-d - tworzy dowiązanie twarde do katalogów
-f - nie informuje o nadpisywaniu istniejącego pliku
-s - tworzy dowiązanie symboliczne

less nazwa_pliku

Opis : podgląd plików tekstowych

cat nazwa_pliku

Opis : przesyła plik na standardowe wejście

df [-ah] urządzenie

Opis : sprawdzenie ilości wolnego miejsca na dysku

Opcje :

-a - wyświetla informacje o każdym systemie plików

-h - wyświetlane informacje są podawane w łatwych do odczytania jednostkach

-t=typ - wyświetla informacje tylko o systemie danego typu

mkfs [-t] [-c] urządzenie_blokowe

Opis : tworzenie systemu plików

Opcje :

-t - określa typ tworzonego systemu plików, domyślnie ext2

-c - sprawdza uszkodzone bloki urządzenia przed utworzeniem systemu plików

mount [-t] [-o] urządzenie_lub_katalog

Opis : montowanie systemu plików

Opcje :

-t - typ montowanego systemu plików

-o - po tej opcji następuje podanie oddzielonych przecinkami wartości :

remount - przemontowanie zamontowanego już systemu plików

ro - montuje system plików w tryb tylko do odczytu

rw - montuje system plików w tryb tylko do odczytu i zapisu

nouser - zabrania zwykłemu użytkownikowi montowania systemu plików

umount urządzenie_lub_katalog

Opis : odmontowanie systemu plików

e2fsck [-pcf] [-b] urządzenie

Opis : testowanie systemu plików

Opcje :

-p - napraw automatycznie system plików bez żadnych pytań

-c - uruchom program badblocks w celu znalezienia uszkodzonych bloków

-f - wymuś sprawdzanie, nawet jeśli system plików wydaje się być czysty

-b superblok - użyj podanego superbloku

fdisk [-lv] [-s] urządzenie

Opis : obsługa partycji

Opcje :

-l - drukuje tablice partycji

-v - drukuje numer wersji fdiska

-s partycja - jest partycją wybranego dysku
urządzenie - jest plikiem z katalogu /dev wskazującym na dyski (np.: hda)

fdformat [-n] urządzenie

Opis : niskopoziomowy format dyskietki

Opcje :

-n - wyłączenie weryfikacji sformatowanej dyskietki

passwd [-l][-u][-d][-S] nazwa_użytkownika

Opis : zmiana hasła użytkownika

Opcje :

-l - zablokowanie konta

-u - ponowne udostępnienie konta

-S - wyświetlenie informacji o stanie konta

su nazwa_użytkownika

Opis : chwilowa zmiana aktywnego użytkownika

insmod [-fp] [-o] plik_modułu

Opis : instalowanie modułu jądra

Opcje :

-f - usiłuje załadować moduł nawet jeśli wersja uruchomionego jądra i jądra dla którego moduł został skompilowany nie zgadza się

-p - sprawdza czy moduł może być załadowany bez kłopotów

-o nazwa_modułu - podanie dokładnej nazwy modułu

rmmod [-a] moduł

Opis : usuwanie modułu

Opcje :

-a - usuń wszystkie nieużywane moduły

lsmod

Opis : listowanie aktywnych modułów

depmod moduł

Opis : sprawdzanie zależności między modułami

modprobe moduł

Opis : instalowanie modułu z potrzebnymi opcjami i modułami

modinfo [-adnp] plik_modułu

Opis : informacje o modułach

Opcje :

- a - wypisuje autora modułu
- d - wypisuje opis modułu
- n - wypisuje nazwę pliku modułu
- p - wypisuje parametry wspierane przez moduł

top [-p]

Opis : listuje procesy o największym obciążeniu procesora i wyświetla ich statystyki w czasie

rzeczywistym

Opcje :

- p - monitoruje jedynie proces o danym id procesu

pidof nazwa_programu

Opis : wyświetlenie numeru procesu wybranego programu

ps [-auxr][ttx]

Opis : listuje aktywne procesy

Opcje :

- a - pokaż procesy innych użytkowników
- u - format użytkownika - podaje nazwę użytkownika i czas startu
- x - pokaż procesy bez kontrolującego terminala
- r - pokaż tylko pracujące procesy
- ttx - pokaż tylko procesy z kontrolującym tty XX

kill [-s] numer_procesu

Opis : zakańcza działanie procesu

Opcje :

- s - rodzaj wysłanego sygnału

jobs

Opis : listuje aktywne programy

fg numer_zadania

Opis : ustawienie procesu na front

bg numer_zadania

Opis : przeniesienie procesu w tło

ifconfig <if> <IP> <nm> <bc>

Opis : konfigurowanie interfejsu sieciowego

Opcje :

if - rodzaj interfejsu (np.: eth0)

IP - adres IP interfejsu

nm - maska sieci

bc - adres rozgłoszeniowy

ifconfig interfejs up

Opis : uruchamianie interfejsu sieciowego

ifconfig interfejs down

Opis : zatrzymywanie interfejsu sieciowego

route add|del [-net|-host] gw cel netmask dev

Opis : kontrola trasowania

Opcje :

add - dodanie drogi do tablicy trasowania

del - usunięcie drogi z tablicy trasowania

-net - oznacza, że celem jest sieć

-host - oznacza, że celem jest stacja

gw brama - adres bramy sieciowej

cel - docelowa sieć lub stacja

netmask netmaska - maska sieci

dev interfejs - wymusza związanie trasy z podanym interfejsem

netstat [-riM]

Opis : wyświetla połączenia sieciowe i statystyki połączeń

Opcje :

-r -wyświetl tablice routingu

-i - wyświetla tablice wszystkich działających interfejsów sieciowych

-M - wyświetl listę wszystkich maskaradowanych sesji

grep [-civ] wzorzec [plik]

Opis : wyszukiwanie ciągów znaków

Opcje :

-c - podaje liczbę wierszy w których znaleziono pasujący wzorzec

-i - pomija różnicę w wielkości liter

-v - wyświetla wiersze, w których nie ma podanego wzorca

plik - nazwa pliku, w którym następuje przeszukanie

find

Opis : szukanie plików

whereis [-bms] nazwa pliku

Opis : lokalizuje źródła, binaria i dokumentację man

Opcje :

-b - szukaj tylko binariów

-m - szukaj tylko sekcji dokumentacji man

-s - szukaj tylko źródeł

man polecenie

Opis : manual systemowy

info

Opis : podręcznik GNU

apropos klucz

Opis : przeszukuje bazy z dokumentacją pod kątem słów kluczowych

date [-ds]

Opis : podaje lub ustawia datę systemową i czas

Opcje :

-d DATA - podaje datę i czas, określone podanym wyrażeniem

-s DATA - ustawia podany czas i datę

logout , exit

Opis : zakończenie sesji

clear

Opis : czyszczenie ekranu terminala

shutdown [-akh] czas ostrzeżenie

Opis : restart lub wyłączenie systemu

Opcje :

-a - użyj **/etc/shutdown.allow** w celu zabronienia wykonywania tego polecenia

-k - wyślij komunikat o zamknięciu ale nie wykonuj prawdziwego zamknięcia systemu

-h - zatrzymaj po wyłączeniu systemu

czas - czas po którym system zostanie zamknięty.

Forma czasu : HH(godzina):MM(minuty) lub +m gdzie m jest liczbą minut do odczekania

halt

Opis : zamknięcie systemu

reboot

Opis : przeładowanie systemu

Dodatek B: Spis poleceń informacyjnych systemu Linux.

Polecenia stosowane w Linuksie i przedstawiające informacje o systemie.

<code>cal</code>	wyświetla kalendarz na aktualny miesiąc
<code>cat /etc/issue</code>	wyświetla informacje, jakiej dystrybucji Linuksa używasz
<code>cat /proc/cpuinfo</code>	wyświetli informacje o procesorze (procesorach).
<code>cat /proc/filesystems</code>	pokazuje używane rodzaje systemów plików
<code>cat /proc/interrupts</code>	pokazuje informacje o używanych przerwaniach, i nazwy urządzeń im przypisanych.
<code>cat /proc/pci</code>	lista informacji o urządzeniach podłączonych do magistrali PCI
<code>cat /proc/version</code>	informacje o wersji Linuksa
<code>chage -l nazwa_uzytkow</code>	pokazuje informacje o ważności hasła
<code>date</code>	wyświetla bieżącą datę i czas systemowy oraz pozwala na ich zmianę.
<code>df -h</code>	informacje się o wykorzystaniu dysków twardych
<code>du / -bh more</code>	pokazuje ile miejsca na dysku zajmują poszczególne katalogi.
<code>echo \$PATH</code>	wyświetla zawartość zmiennej środowiskowej "PATH".
<code>free</code>	informacja o wykorzystaniu pamięci.
<code>finger nazwa_uzytkownika</code>	dane systemowe użytkownika.
<code>hostname</code>	wyświetla nazwę hosta.
<code>id nazwa_uzytkownika</code>	wyświetla identyfikator grupy (gid) i użytkownika (uid) oraz grupy danego użytkownika
<code>last</code>	ujrzysz listę wszystkich użytkowników ostatnio logujących się do twojego systemu.
<code>lastb</code>	informacje o ostatnich nieudanych próbach logowań do systemu.
<code>less /var/log/dmesg</code>	pokazuje komunikaty jądra zrzuczone po starcie systemu.
<code>lsmod</code>	(list modules) pokazuje załadowane moduły jądra,
<code>printenv</code>	pokazuje aktualnie zdefiniowane zmienne środowiskowe.
<code>ps</code>	(proces status) wyświetla listę aktualnie działających procesów będących własnością użytkownika. Polecenie ps axu more pokaże wszystkie działające procesy w systemie razem z nazwą ich właścicieli.
<code>pstree</code>	wyświetla listę aktualnie działających procesów w postaci drzewa.

<code>pwd</code>	wyświetla nazwę aktualnego katalogu roboczego.
<code>rwho -a</code>	pokazuje użytkowników zalogowanych na innych komputerach w sieci. Aby komenda działała musi być włączona ta usługa (rwho).
<code>set more</code>	pokazuje ustawienia środowiska użytkownika.
<code>top</code>	wyświetla listę procesów w kolejności według użycia procesora i zajętości pamięci, najbardziej wymagające jako pierwsze.
<code>uname -a</code>	wyświetla wszystkie dane o systemie i jądrze Linuksa.
<code>uptime</code>	wyświetla czas od ostatniego restartu, oraz tzw. średnią ładowania (ang. load average). Jest to tempo ładowania procesów do pamięci. Polecenie pokazuje średnią z ostatniej minuty, oraz pięciu i piętnastu minut.
<code>w</code>	informuje kto jest zalogowany i ile czasu w systemie, co robi, ile zasobów procesora używa.
<code>who</code>	pokazuje użytkowników zalogowanych do systemu, oraz czas ich rejestracji i typ terminala.
<code>whoami</code>	wyświetli Twoją nazwę użytkownika.

Niektóre z tych poleceń służą nie tylko do celów informacyjnych, lecz z odpowiednią opcją umożliwiają inne działanie. Aby dowiedzieć się czegoś więcej najlepiej zajrzeć do elektronicznych podręczników (man).