



MINISTERSTWO EDUKACJI
i NAUKI



Ewa Sromala

**Administrowanie systemem operacyjnym Unix
312 [01].Z4.02**

Poradnik dla ucznia

Wydawca
Instytut Technologii Eksploatacji – Państwowy Instytut Badawczy
Radom 2005

Recenzenci:

mgr Bogdan Kostrzewa
mgr inż.. Krzysztof Pytel

Opracowanie redakcyjne:

mgr inż. Katarzyna Maćkowska

Konsultacja:

dr inż. Bożena Zając

Korekta:

mgr inż. Tomasz Sułkowski

Poradnik stanowi obudowę dydaktyczną programu jednostki modułowej 312[01].Z4.02 Programowanie w środowisku języka obiektowego zawartego w modułowym programie nauczania dla zawodu technik informatyk

Wydawca

Instytut Technologii Eksploatacji – Państwowy Instytut Badawczy, Radom 2005

SPIS TREŚCI

1. Wprowadzenie	4
2. Wymagania wstępne	5
3. Cele kształcenia	6
4. Materiał nauczania	7
4.1. Charakterystyka systemu Unix	7
4.1.1. Materiał nauczania	7
4.1.2. Pytania sprawdzające	8
4.1.3. Ćwiczenia	8
4.1.4. Sprawdzian postępów	9
4.2. Uruchamianie i zamykanie systemu	10
4.2.1. Materiał nauczania	10
4.2.2. Pytania sprawdzające	11
4.2.3. Ćwiczenia	11
4.2.4. Sprawdzian postępów	12
4.3. Praca wielodostępna i wielozadaniowa	13
4.3.1. Materiał nauczania	13
4.3.2. Pytania sprawdzające	14
4.3.3. Ćwiczenia	15
4.3.4. Sprawdzian postępów	16
4.4. System plików Unix – praca z plikami i katalogami	17
4.4.1. Materiał nauczania	17
4.4.2. Pytania sprawdzające	20
4.4.3. Ćwiczenia	20
4.4.4. Sprawdzian postępów	22
4.5. Programy użytkowe w środowisku Unix	23
4.5.1. Materiał nauczania	23
4.5.2. Pytania sprawdzające	25
4.5.3. Ćwiczenia	25
4.5.4. Sprawdzian postępów	26
4.6. Powłoka systemu i skrypty	27
4.6.1. Materiał nauczania	27
4.6.2. Pytania sprawdzające	29
4.6.3. Ćwiczenia	29
4.6.4. Sprawdzian postępów	30
4.7. Strumienie, potoki i filtry	31
4.7.1. Materiał nauczania	31
4.7.2. Pytania sprawdzające	32
4.7.3. Ćwiczenia	32
4.7.4. Sprawdzian postępów	33
4.8. Konta użytkowników i proces logowania	34
4.8.1. Materiał nauczania	34
4.8.2. Pytania sprawdzające	35
4.8.3. Ćwiczenia	35
4.8.4. Sprawdzian postępów	36

4.9. Zarządzanie procesami	37
4.9.1. Materiał nauczania	37
4.9.2. Pytania sprawdzające	37
4.9.3. Ćwiczenia	38
4.9.4. Sprawdzian postępów	38
4.10. Montowanie urządzeń i inne systemy plików	39
4.10.1. Materiał nauczania	39
4.10.2. Pytania sprawdzające	40
4.10.3. Ćwiczenia	40
4.10.4. Sprawdzian postępów	41
4.11. Bezpieczeństwo danych i kopia zapasowa	43
4.11.1. Materiał nauczania	43
4.11.2. Pytania sprawdzające	44
4.11.3. Ćwiczenia	45
4.11.4. Sprawdzian postępów	46
4.12. Sieć TCP/IP – konfiguracja i usługi	47
4.12.1. Materiał nauczania	47
4.12.2. Pytania sprawdzające	49
4.12.3. Ćwiczenia	49
4.12.4. Sprawdzian postępów	50
4.13. Powłoka graficzna systemu	51
4.13.1. Materiał nauczania	51
4.13.2. Pytania sprawdzające	51
4.13.3. Ćwiczenia	51
4.13.4. Sprawdzian postępów	53
4.14. Programy biurowe i użytkowe	54
4.14.1. Materiał nauczania	54
4.14.2. Pytania sprawdzające	54
4.14.3. Ćwiczenia	54
4.14.4. Sprawdzian postępów	56
4.15. Drukowanie	57
4.15.1. Materiał nauczania	57
4.15.2. Pytania sprawdzające	57
4.15.3. Ćwiczenia	58
4.15.4. Sprawdzian postępów	58
4.16. Instalacja systemu Unix	59
4.16.1. Materiał nauczania	59
4.16.2. Pytania sprawdzające	60
4.16.3. Ćwiczenia	60
4.16.4. Sprawdzian postępów	61
4.17. Uruchamianie programów napisanych dla systemu MS Windows	62
4.17.1. Materiał nauczania	62
4.17.2. Pytania sprawdzające	62
4.17.3. Ćwiczenia	62
4.17.4. Sprawdzian postępów	63
5. Sprawdzian osiągnięć	64
6. Literatura	68

1. WPROWADZENIE

Poradnik pomoże Ci w kształtowaniu umiejętności korzystania z systemu Unix i podobnych systemów. W poszczególnych rozdziałach omówione zostaną kolejne zagadnienia obsługi systemu, wykonywania w nim podstawowych czynności, naprawiania prostych i skomplikowanych awarii, oraz administrowania serwerów. Unix nie oznacza jednego, konkretnego systemu operacyjnego, lecz całą rodzinę systemów, które charakteryzują podobne cechy, takie jak stabilność, wielozadaniowość i wielodostępność. Różnią się natomiast takimi szczegółami jak zainstalowane oprogramowanie, czy składnia poleceń powłoki.

Dla uproszczenia, w całym poradniku stosowany jest system FreeBSD, jako przedstawiciel systemów z rodziny Unix. Dlaczego akurat ten system? Przemawia za tym kilka powodów. Po pierwsze jest on udostępniony zupełnie za darmo na licencji BSD, co oznacza że można go bez żadnych opłat wykorzystywać zarówno w szkole, w domu jak i później, w zastosowaniach komercyjnych w pracy. Po drugie, jest on najbardziej zbliżony do dużych, komercyjnych systemów Unix, które można będzie spotkać w późniejszej pracy zawodowej.

W poradniku znajdują się tylko najważniejsze informacje – definicje, terminy, schematy i podstawowe wiadomości. Są też zasygnalizowane zagadnienia, które trzeba uzupełnić z innych źródeł. Na szczęście w Internecie jak i w samym systemie istnieje bardzo dużo źródeł, z których takie poszerzone informacje można uzyskać. Przede wszystkim jest to elektroniczny podręcznik (manual) wbudowany w system. Wystarczy w konsoli wpisać `man nazwa_polecenia` aby uzyskać informacje o wybranym poleceniu. Np: `man man` pokaże instrukcję obsługi samego podręcznika, a `man date` - informacje dotyczące użycia polecenia do obsługi daty i czasu systemowego. Ponadto dużo informacji znajduje się na stronach Wolnej Encyklopedii Wikipedia (www.wikipedia.org.pl) oraz na licznych stronach o tematyce Unixowej, które można znaleźć za pomocą dobrej wyszukiwarki.

Ponadto ze wszystkimi pytaniami można zwrócić się do nauczyciela.

2. WYMAGANIA WSTĘPNE

- Przystępując do realizacji programu nauczania jednostki modułowej powinieneś umieć:
- sprawnie skorzystać z oprogramowania w zakresie zarządzania zasobami i przetwarzania tekstu,
 - skorzystać z różnych źródeł wiedzy w celu wyszukiwania informacji na zadany temat,
 - wyselekcjonować, uporządkować i przechować informacje za pomocą różnych narzędzi i na różnych nośnikach,
 - pracować w sieci komputerowej i skorzystać z podstawowych usług Internetowych, takich jak www i e-mail.

3. CELE KSZTAŁCENIA

W wyniku realizacji programu jednostki modułowej powinieneś umieć:

- wyjaśnić podstawowe cechy systemu Unix,
- wyjaśnić pojęcia wielodostępu i wielozadaniowości,
- zastosować programy do pracy z plikami i katalogami,
- swobodnie korzystać z różnych systemów plików,
- zastosować programy narzędziowe do wykonania prostych zadań takich jak edycja plików tekstowych,
- posłużyć się pocztą elektroniczną,
- zastosować polecenia powłok systemu takich jak csh i ksh,
- zastosować skrypty do zautomatyzowania czynności administracyjnych,
- uruchomić w systemie Unix programy napisane dla systemu DOS i Windows,
- skonfigurować i uruchomić system X Windows,
- użyć programów narzędziowych, graficznych i biurowych w środowisku X Windows,
- wykonać backup systemu,
- zainstalować system Unix na stacji roboczej i serwerze.

4. MATERIAŁ NAUCZANIA

4.1. Charakterystyka systemu Unix

4.1.1. Materiał nauczania

System Unix, został zapoczątkowany w latach 1960 i 1970 przez pracowników firm AT&T oraz Bell Labs, między innymi Kena Thompsona, Dennisa Ritchie i Douglasa McIlroya. Dzisiaj istnieje wiele jego odmian stworzonych przez AT&T oraz inne komercyjne firmy i organizacje non-profitowe.

Unix z założenia miał być systemem wszechstronnym, wielozadaniowym i wieloużytkownikowym. Oznacza to, że na jednym systemie może pracować na raz wiele osób, korzystając z wielu programów, co pozwala na wdrażanie go w rozmaite środowiska: stacje robocze, wielkie serwery obliczeniowe, systemy sterowania aparaturą przemysłową.

Istnieje wiele skojarzeń, które trafnie opisują ten system:

- większość informacji jest zapisywana w zwykłych plikach tekstowych (konfiguracja systemu, bazy danych programów etc),
- polecenia można wydawać z konsoli tekstowej,
- system plików ma budowę hierarchiczną,
- urządzenia i niektóre formy komunikacji międzyprogramowej utożsamiane są z plikami (monitor i drukarka mają swoje nazwy w drzewie plików i katalogów i można na nie przekierować wynik działania dowolnego programu),
- programistom kojarzy się głównie z językiem C.

Unix jest systemem sieciowym, co oznacza że są w niego wbudowane narzędzia pozwalające na korzystanie z sieci komputerowej i pełna obsługa protokołów TCP/IP, między innymi udostępnianie plików, drukarek i innych zasobów. Składa się zwykle z niewielkiego jądra (*kernel*) będącego trzonem systemu, odpowiedzialnego za uruchamianie procesów oraz zestawu aplikacji. Procesami mogą być aplikacje użytkownika lub usługi sieciowe.

Użytkownik komunikuje się z jądrem przez powłokę (shell). Jest to program, który przyjmuje polecenia od użytkownika i przekazuje je w odpowiedniej formie do jądra. Istnieje wiele powłok zarówno tekstowych jak i graficznych interfejsów użytkownika.

Przede wszystkim jednak Unix kojarzy się z niespotykaną stabilnością i skalowalnością. W rozwiązaniach, w których wymagana jest niezawodność i pewność działania stosuje się właśnie systemy z rodziny Unix, albo Unixo-podobne.

Nie każdy system, który odpowiada powyższej specyfikacji można nazwać Unixem. Istnieje zbiór standardów nazywany Single Unix Specification (SUS), nadzorowanych obecnie przez Austin Group (wcześniej IEEE i The Open Group) określający, które systemy można nazwać mianem „Unix“, a których nie można. Te drugie nazywamy Unixo-podobnymi (Unix-like systems). FreeBSD, który będzie wykorzystywany w tym poradniku jest właśnie takim systemem Unixo-podobnym. Powstał na uniwersytecie w Berkley, podobnie jak cała gałąź systemów BSD (OpenBSD, NetBSD etc). Nazwa BSD pochodzi od: Berkley System Distribution.

Do systemów Unix-like należy także Linux – zapoczątkowany przez Linusa Torvaldsa darmowy system zdobywający sobie coraz większą popularność wśród użytkowników domowych i instytucji dzięki licznym zaletom.

Ważną rolę odgrywa kwestia licencji i praw do użytkowania programów. Podczas gdy „duże i dojrzałe“ systemy Unix (np. Solaris) są najczęściej bardzo drogie i tylko największe

firmy mogą sobie na nie pozwolić, system Linux dostępny jest na licencji GPL (General Public Licence), która zezwala na jego nieodpłatnie użytkowanie z zachowaniem pewnych ograniczeń. Stanowi także, że twórcy udostępniają publicznie kod źródłowy swoich programów, dzięki czemu każdy może wносить do nich swoje poprawki, zmiany i pracować nad ich udoskonaleniem. Podobnie systemy *BSD dostępne są na licencji BSD, która również pozwala na użycie systemu i oprogramowania za darmo, zarówno do celów prywatnych jak i komercyjnych, a czasami nawet więcej.

Warto wiedzieć, że wszystkie systemy Unixowe, BSD i Linux są do siebie podobne. Oznacza to, że osoba, która potrafi korzystać, czy administrować FreeBSD, będzie w dużym stopniu znała wszystkie systemy Unixowe. Jeżeli zajdzie potrzeba administrowania serwerem opartym na Linux'ie albo Solaris'ie, trzeba będzie zapoznać się tylko ze szczegółami jego budowy takimi jak lokalizacja poszczególnych plików konfiguracyjnych, czy zmiany w składni użycia poleceń.

4.1.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie są podstawowe cechy systemu Unix?
2. Na czym polega rola jądra systemu?
3. Co decyduje o tym, który system można nazwać Unixem, a który nie?
4. Na jakiej licencji udostępniany jest system Linux, a na jakiej FreeBSD? Na czym polegają te licencje?

4.1.3. Ćwiczenia

Ćwiczenie 1

Wyszukaj w sieci Internet nazwy najpopularniejszych systemów Unix, *BSD oraz dystrybucji Linuxa.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wejść na stronę dobrej wyszukiwarki,
- 2) wpisać słowa kluczowe takie jak `bsd`, `linux`, `dystrybucja` i znaleźć odpowiednie strony `www`,
- 3) pamiętać o zasadach wyszukiwania i składni zapytań,
- 4) zapisać znalezione nazwy,
- 5) podzielić się rezultatami z innymi uczniami w grupie.

Wyposażenie stanowiska pracy:

- komputer z dostępem do Internetu.

Ćwiczenie 2

Znajdź w Internecie informacje na temat licencji GPL i BSD. Wypisz najistotniejsze różnice w tych dwóch licencjach.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś

- 1) przeszukując strony Wolnej Encyklopedii Wikipedii (pl.wikipedia.org) oraz polską stronę GNU (www.gnu.org.pl) znaleźć potrzebne informacje,
- 2) zapisać wyniki wyszukiwania,
- 3) podzielić się rezultatami z innymi uczniami w grupie.

Wyposażenie stanowiska pracy:

- komputer z dostępem do Internetu.

Ćwiczenie 3

Znajdź adresy 5 polskich witryn traktujących o systemach z rodziny Linux i BSD.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś

- 1) wejść na stronę dobrej wyszukiwarki,
- 2) wpisać słowa kluczowe,
- 3) znaleźć potrzebne informacje,
- 4) zapisać wyniki wyszukiwania,
- 5) podzielić się rezultatami z innymi uczniami w grupie.

Wyposażenie stanowiska pracy:

- komputer z dostępem do Internetu.

4.1.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) podać rodowód systemu Unix?	<input type="checkbox"/>	<input type="checkbox"/>
2) wymienić założenia systemu Unix?	<input type="checkbox"/>	<input type="checkbox"/>
3) wyjaśnić co to znaczy, że Unix jest systemem sieciowym?	<input type="checkbox"/>	<input type="checkbox"/>
4) rozwinąć i wyjaśnić akronim SUS?	<input type="checkbox"/>	<input type="checkbox"/>
5) umiejscowić LINUXa w rodzinie Unixowej?	<input type="checkbox"/>	<input type="checkbox"/>
6) opisać system BSD?	<input type="checkbox"/>	<input type="checkbox"/>

4.2. Uruchamianie i zamykanie systemu

4.2.1. Materiał nauczania

Istnieje wiele odmian Unixa i systemów Unixo-podobnych, takich jak BSD czy Linux (odmiany Linuxa nazywamy dystrybucjami). Niektóre są bardzo proste i mieszczą się w komputerze sterującym pracą lodówki czy w zegarku, inne są bardzo złożone i wykonują na raz wiele różnorodnych zadań. Mimo tych różnic, proces uruchamiania i zamykania systemu pozostaje podobny i składa się zwykle z tych samych elementów. W tym rozdziale zostanie przedstawiona procedura startu (bootup) i zamknięcia (shutdown) systemu Unix na przykładzie FreeBSD. Bardziej szczegółowe informacje są dostępne na stronie projektu FreeBSD (www.freebsd.org) oraz innych stronach poświęconych systemom Unix.

Uruchomienie systemu

Na początku BIOS (Podstawowy system wejścia-wyjścia) wyszukuje na dysku twardym specjalny blok rozruchowy MBR (Master Boot Record), który zawiera pierwsze instrukcje uruchomieniowe. Program zawarty w MBR wykonuje pierwsze testy i inicjuje ładowanie jądra systemu

Następuje załadowanie jądra (kernel) systemu. Jądro wykrywa sprzęt i podzespoły komputera, konfiguruje je, instaluje do nich sterowniki i ustawia najważniejsze parametry pracy systemu. Na końcu przekazuje kontrolę procesowi o nazwie „init“.

Proces „init“ na początku sprawdza, czy twarde dyski komputera (lub inne urządzenia pamięci masowej) są zdadne do użycia i gotowe, po czym przystępuje do konfiguracji pozostałych parametrów. Montuje przy tym wszystkie twarde dyski i napędy, ustawia karty sieciowe i uruchamia programy, które muszą być uruchomione przy starcie.

Końcową fazą procesu jest uruchomienie narzędzia, które będzie obsługiwało konsolę (ekran i klawiaturę) i uruchomi na nich w razie potrzeby powłokę (na FreeBSD tym narzędziem jest `getty`).

Podczas procesu uruchamiania systemu na ekranie pokazują się kolejne napisy informujące o aktualnie wykonywanej operacji i stanie systemu. Na końcu widać zgłoszenie się systemu i zachętę do zalogowania się.

Proces logowania polega na uwierzytelnieniu tożsamości użytkownika – poinformowaniu systemu, kto będzie na nim pracował. W tym celu należy podać nazwę użytkownika (login) i hasło (password). Na świeżo zainstalowanym systemie jest tylko jeden użytkownik – root. Jest to konto administratora systemu. Administrator logując się jako root i podając poprawne hasło zyskuje nieograniczone uprawnienia. Może wpływać na wszystkie aspekty działania systemu, usuwać, zmieniać i dodawać dane w dowolnych miejscach, nawet jeśli miałyby to zagrozić bezpieczeństwu i stabilności całego systemu. W życiu codziennym bardzo rzadko pracuje się korzystając z konta root, ponieważ nie są potrzebne aż takie uprawnienia. Jest to wręcz niewskazane, bo czasem przez nieuwagę można skasować istotne zbiory z dysku twardego!

Po zalogowaniu się system prezentuje znak zachęty powłoki systemu (shell) i czeka na instrukcje. Polecenia wpisuje się korzystając z klawiatury ze zwróceniem uwagi na wielkość liter (w systemie Unix Pies to nie to samo co PIES czy pies). Każdą komendę zatwierdza się przyciskiem Enter. Aby wylogować się, czyli zakończyć działanie wszystkich programów użytkownika, zamknąć powłokę i powrócić do pytania o login i hasło należy wpisać polecenie `logout`. Ważne jest, by za każdym razem przed odejściem od konsoli wylogować się albo zablokować dostęp. W przeciwnym wypadku ktoś może skorzystać z otwartego konta i podejrzeć dane, a nawet wykonać operacje, które normalnie nie byłyby wogóle dla niego dostępne.

Zamykanie systemu

Gdy komputer (lub serwer) nie jest już potrzebny, czasami zachodzi potrzeba jego wyłączenia. Czasami, ponieważ serwerów zazwyczaj nie wyłącza się bez ważnej potrzeby, a i wielodostępne komputery z systemem Unix, na których pracuje więcej niż jedna osoba pozostają często włączone przez 24h/dobę. Tym niemniej, system można wyłączyć (rozpocząć procedurę zamykania systemu) wykonując odpowiednie polecenie z konta użytkownika root. Tak więc należy najpierw zalogować się jako użytkownik root podając prawidłowe hasło, a następnie wpisać polecenie

```
# shutdown
```

Jest to skrypt inicjujący procedurę zamknięcia systemu. Dodaje się do niego dwa przełączniki informujące kolejno, że po wykonaniu procedury praca systemu ma zostać zatrzymana (-h lub wznowiona -r), oraz że procedura ma się rozpocząć natychmiast (now lub po określonym czasie). Istnieje wiele innych przełączników, o których można przeczytać w dokumentacji systemu (man shutdown).

Podczas procedury zamknięcia, wyłączane są wszystkie działające jeszcze programy, odmontowywane dyski twarde i wszystkie inne systemy plików oraz wylogowywani zostają wszyscy użytkownicy. Na końcu system wykonuje kilka czynności porządkowych (między innymi usuwanie niepotrzebnych plików tymczasowych). Wyłączenie komputera bez przestrzegania powyższej procedury może spowodować utratę lub uszkodzenie części danych.

4.2.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Z jakich etapów składa się kolejno procedura uruchamiania systemu?
2. Co oznacza skrót BIOS?
3. Jaki proces odpowiada za uruchomienie programów startujących razem z systemem?
4. Do czego służy konto root?
5. Co dzieje się podczas zamykania systemu?

4.2.3. Ćwiczenia

Ćwiczenie 1

Włącz system operacyjny, zaloguj się i wpisz w powłoce systemu polecenie `date`, aby sprawdzić bieżącą datę i czas. Następnie wyloguj się.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) włączyć komputer,
- 2) po pojawieniu się prośby o zalogowanie wpisać nazwę użytkownika i hasło podane przez nauczyciela,
- 3) po pojawieniu się znaku zachęty powłoki wpisać polecenie `date` i zatwierdzić klawiszem Enter,
- 4) na ekranie powinien się pojawić wynik tego polecenia – aktualna data i czas,
- 5) wpisać polecenie `logout` i zatwierdzić Enterem.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 2

Włącz system operacyjny, zaloguj się i wpisz w powłoce systemu polecenie `man man`, aby przeczytać instrukcje dotyczące dokumentacji elektronicznej.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś

- 1) włączyć komputer,
- 2) po pojawieniu się prośby o zalogowanie wpisać nazwę użytkownika i hasło podane przez nauczyciela,
- 3) po pojawieniu się znaku zachęty powłoki wpisać polecenie `man man` i zatwierdzić klawiszem Enter. Na ekranie powinien się pojawić wynik tego polecenia – przeglądarka pomocy systemowej,
- 4) poruszać się po tekście pomocy klawiszami strzałek w górę i w dół. Aby wyjść z programu - nacisnąć klawisz Q,
- 5) wpisać polecenie `logout` i zatwierdzić Enterem.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 3

Wyłącz system operacyjny.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś

- 1) zalogować się jako root wpisując nazwę użytkownika root i hasło podane przez nauczyciela (UWAGA: root ma inne hasło niż użytkownik z poprzednich ćwiczeń!),
- 2) sprawdzić w podręczniku (`man shutdown`), w jaki sposób należy poprawnie zamknąć system,
- 3) wpisać polecenie `shutdown` z atrybutami określającymi zatrzymanie systemu natychmiast (`now`) i potwierdzić Enterem, kiedy komputer napisze „System halted!” można go bezpiecznie wyłączyć.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

4.2.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) wymienić czynności wykonywane w trakcie uruchamiania systemu?	<input type="checkbox"/>	<input type="checkbox"/>
2) opisać, na czym polega logowanie?	<input type="checkbox"/>	<input type="checkbox"/>
3) opisać użytkownika root?	<input type="checkbox"/>	<input type="checkbox"/>
4) wyjaśnić pojęcie powłoki systemu?	<input type="checkbox"/>	<input type="checkbox"/>
5) poprawnie zamknąć system?	<input type="checkbox"/>	<input type="checkbox"/>

4.3. Praca wielodostępna i wielozadaniowa

4.3.1. Materiał nauczania

W rozdziale 4.1 można przeczytać, że każdy Unix jest systemem wielodostępnym i wielozadaniowym. Oto bliższe wyjaśnienie tych pojęć.

Wielodostęp

Termin ten określa sposób, w jaki kilka osób może pracować na jednym komputerze. Wyobraźmy sobie, że posiadamy bardzo silny komputer z bardzo szybkim procesorem i dużą ilością pamięci. Można na nim uruchomić każdy wymagający program komputerowy i zawsze pozostaną wolne zasoby – żaden program nie jest w stanie zagospodarować ich wszystkich. W normalnej sytuacji, gdy przy komputerze pracuje jedna osoba, te niewykorzystane zasoby marnują się – nikt z nich nie korzysta. Jeśli korzysta się z systemu, który zapewnia wielodostęp, inna osoba – użytkownik – może zalogować się na tym samym komputerze z innej lokalizacji (korzystając z monitora i klawiatury stojącej obok, albo z zupełnie innego miejsca na świecie) i wykonywać swoją pracę.

Dzięki wielodostępowi możliwe jest optymalne wykorzystanie wszystkich zasobów komputera i dzielenie ich między wszystkich użytkowników. Rozwiązuje też problem, który pojawia się, gdy dwie osoby chcą skorzystać z jednego komputera w tym samym czasie. Jedna z nich może użyć lokalnej konsoli (rozumianej jako monitor, klawiatura i ewentualnie mysz podłączone bezpośrednio do komputera), a druga konsoli zdalnej, czyli specjalnie przystosowanego do tego celu terminala (urządzenie które posiada monitor i klawiaturę, oraz niewielki układ pozwalający na połączenie się przez sieć komputerową ze zdalnym serwerem) lub innego komputera. Drugi przypadek nazywamy pracą zdalną, który w systemach Unix może być realizowany na wiele sposobów.

Implementacja wielodostępu w systemach Unix

Z technicznego punktu widzenia istnieje kilka sposobów podłączenia się do komputera. Są one od siebie niezależne i użytkownik może wybrać dowolny z nich, pod warunkiem, że administrator włączył odpowiednią usługę na komputerze.

Najprostszym sposobem jest dzielenie się lokalną konsolą. Praktycznie wszystkie Unixy mają tę funkcję wbudowaną w system od samego początku zaraz po instalacji. Aby druga osoba mogła skorzystać z konsoli lokalnej wystarczy użyć odpowiedniej kombinacji klawiszy – najczęściej ALT + Fx (lub CTRL + ALT + Fx), gdzie Fx to jeden z klawiszy funkcyjnych F2, F3, F4... Tak uruchamia się wirtualną konsolę, na której druga osoba może się zalogować i pracować w systemie. Aby wrócić do pierwszej konsoli należy nacisnąć odpowiednio ALT + F1 (lub CTRL + ALT + F1). Między konsolami można się dowolnie przełączać, a programy pozostawione na którejkolwiek z nich kontynuują swoje działanie nawet po przełączeniu się na inną.

Innym sposobem na równoczesną pracę na jednym komputerze jest praca zdalna za pomocą takich usług jak telnet lub jego szyfrowana wersja – SSH (secure shell). Można użyć monitora i klawiatury innego komputera aby przejąć kontrolę nad hostem, to znaczy komputerem podłączonym do Internetu lub innej sieci używającej protokołu TCP/IP, posiadającym swój adres i udostępniającym usługi innym komputerom w sieci. Warunkiem jest, by obydwa były podłączone do Internetu lub tej samej sieci LAN. Najprościej jest użyć wbudowanych w większość systemów klientów.

Żałómy, że użytkownik korzysta z komputera o nazwie „uczen”, a chce się połączyć z komputerem o nazwie „obcy”. Powinien w powłoce komputera bar wpisać polecenie

```
# telnet obcy
```

lub dla SSH odpowiednio

```
# ssh janek@obcy
```

„Obcy” to nazwa komputera, z którym chce się połączyć, a „janek” to login użytkownika, który się loguje (telnet pyta o login po połączeniu). W przypadku usługi SSH należy pamiętać o znaku at (@) pomiędzy nazwą użytkownika zdalnego, a nazwą hosta. Po wpisaniu hasła pokaże się powłoka hosta obcy i będzie można na nim pracować tak, jakby znajdował się na biurku obok.

Wielozadaniowość

Wielozadaniowość to cecha systemu operacyjnego, umożliwiająca równoczesne wykonywanie więcej niż jednego procesu. Oznacza to między innymi, że użytkownik na swoim komputerze może mieć uruchomionych kilka programów i pracować w nich jednocześnie. Przede wszystkim jednak, wielozadaniowość wykorzystywana jest przez system operacyjny do tego, by poza widocznymi dla użytkownika aplikacjami, uruchomić działające niewidocznie (w tle) usługi. Te usługi to programy, o których użytkownik zazwyczaj nie ma nawet pojęcia. Wykonują czynności takie jak synchronizowanie zegara w komputerze z dokładnymi serwerami czasu, porządkowanie twardego dysku, nasłuchiwanie czy do komputera nie jest podłączane nowe urządzenie, zarządzanie siecią etc. W ten sposób w tym samym czasie na komputerze uruchomionych jest od kilku do kilkudziesięciu procesów, z czego tylko kilka jest widocznych z konsoli.

W sytuacji gdy z jednego komputera korzysta więcej niż jeden użytkownik, każdy z nich ma uruchomioną określoną liczbę procesów. Działają one jednocześnie, nie przeszkadzając sobie nawzajem.

Należałoby się zastanowić jak to się dzieje, że programy na tym samym komputerze (używające tego samego procesora i pamięci) działają równoległe i jeden nie zakłóca działania drugiego. Zadanie to spoczywa zwykle na jądrze systemu. Tam specjalna część – planista – wykonuje algorytm szeregowania zadań do wykonania. Programy, które chcą wykonać operację, zgłaszają planiście zapotrzebowanie, a ten przydziela czas procesora w odpowiedniej kolejności wszystkim programom.

Więcej na temat wielozadaniowości i konkretnych algorytmów szeregowania zadań, można przeczytać w literaturze lub Internecie, szczególnie na stronach poświęconych dystrybucjom Linuxa i innym systemom Unix-like, w sekcji dotyczącej jądra systemu. Dostępne są tam dane na temat najnowszych stosowanych rozwiązań. Na stronie Wikipedii dostępna jest szczegółowa definicja i odnośniki do wyjaśnień terminów w niej użytych. Aby zobaczyć listę procesów aktualnie działających na komputerze, można użyć polecenia `ps ax` w powłoce systemu.

4.3.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Co oznacza pojęcie wielozadaniowości?
2. Co oznacza pojęcie wielodostępu?
3. Czym różni się telnet od SSH?
4. Jak kilku użytkowników może korzystać z jednego stanowiska pracy (jednej konsoli lokalnej)?
5. Jakie warunki muszą być spełnione, aby można się było zdalnie połączyć z jakimś hostem?

4.3.3. Ćwiczenia

Ćwiczenie 1

Zaloguj się na lokalnej konsoli jako użytkownik `uczen` i wykonaj polecenie `ls`, następnie zaloguj się na wirtualnej konsoli jako użytkownik `obcy` i wykonaj polecenie `ps`.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś

- 1) włączyć komputer,
- 2) po pojawieniu się prośby o zalogowanie wpisać nazwę użytkownika `uczeń` i hasło podane przez nauczyciela,
- 3) po pojawieniu się znaku zachęty powłoki wpisać polecenie `ls` i zatwierdzić klawiszem Enter. Na ekranie powinien się pojawić wynik tego polecenia – lista plików w katalogu domowym użytkownika `uczeń`,
- 4) nacisnąć kombinację klawiszy `ALT + CTRL + F2` i zalogować się jako `obcy` z hasłem podanym przez nauczyciela,
- 5) wpisać polecenie `ps` i zaakceptować je klawiszem `ENTER`. Na ekranie pojawi się wynik tego polecenia – lista procesów aktualnie działających na jego koncie (między innymi proces powłoki `tsh` i polecenia `ps`, które właśnie uruchomił),
- 6) wpisać polecenie `logout` i zatwierdzić `Enterem`,
- 7) wrócić na konsolę pierwszą i wylogować użytkownika `uczeń`.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i odpowiednimi kontami użytkowników (`uczeń` i `obcy`)

Ćwiczenie 2

Zaloguj się zdalnie na komputerze wskazanym przez nauczyciela jako użytkownik `uczeń`. Wykonaj na nim polecenie `ps`, aby zobaczyć listę procesów uruchomionych na tym komputerze.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zalogować się na komputerze przed którym siedzi (login i hasło wskaże nauczyciel),
- 2) w konsoli wpisać polecenie `ssh uczen@adres_komputera` wstawiając jako `adres_komputera`, wskazany przez nauczyciela adres komputera zdalnego. Po naciśnięciu Enter zostanie zapytany o hasło użytkownika `uczeń` (powinien wpisać je), a następnie obejrzeć powłokę komputera, do którego się właśnie połączył,
- 3) wykonać polecenie `ps`,
- 4) wylogować się z komputera zdalnego poleceniem `logout`,
- 5) wylogować się z komputera, przy którym pracuje.

Wyposażenie stanowiska pracy:

- dwa komputery z zainstalowanym systemem FreeBSD i odpowiednimi kontami użytkowników (`uczeń`), połączone w sieć.

4.3.4 Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) wyjaśnić, co to jest wielodostęp?	<input type="checkbox"/>	<input type="checkbox"/>
2) wymienić zalety wielodostępu?	<input type="checkbox"/>	<input type="checkbox"/>
3) opisać, jak można zrealizować wielodostęp?	<input type="checkbox"/>	<input type="checkbox"/>
4) wyjaśnić, co to jest wielozadaniowość?	<input type="checkbox"/>	<input type="checkbox"/>
5) wyjaśnić, co to jest proces?	<input type="checkbox"/>	<input type="checkbox"/>
6) połączyć się zdalnie ze wskazanym komputerem?	<input type="checkbox"/>	<input type="checkbox"/>

4.4. System plików Unix – praca z plikami i katalogami

4.4.1. Materiał nauczania

Na większości komputerów z zainstalowanym systemem Unix, system plików wygląda niemal identycznie. Jego podstawową cechą jest hierarchiczność. Oznacza to, że skonstruowany jest na bazie drzewa. Istnieje jeden korzeń (root) oznaczany jako pojedynczy ukośnik (/). Z korzenia (ang. root) wyodrębniają się katalogi, czyli gałęzie drzewa. W katalogach mogą znajdować się pliki (liście) i podkatalogi. Przedstawiony poniżej schemat przedstawia typową strukturę katalogów systemu Unix.

- /
- /boot
- /bin
- /dev
- /etc
- /home
- /lib
- /usr
 - /usr/bin
 - /usr/lib
 - /usr/local
 - /usr/local/bin
 - /usr/local/etc
 - /usr/local/lib
 - /usr/local/share
 - /usr/local/sbin
 - /usr/share
 - /usr/sbin
 - /usr/src
- /mnt
- /root
- /sbin
- /tmp
- /var

Schemat 4.4.1: Drzewo katalogów Unix

- **/boot** – pliki używane przez system przy starcie.
- **/bin** – systemowe pliki wykonywalne, czyli programy, które można uruchomić z konsoli (np. ls, ps, cp itd).
- **/dev** – w Unixach system plików zapewnia między innymi dostęp do urządzeń. W tym katalogu znajdują się specjalne pliki reprezentujące wszystkie urządzenia w systemie.
- **/etc** – pliki konfiguracyjne. Wszystkie programy przechowują w tym miejscu swoje ustawienia.
- **/home** – katalogi domowe użytkowników. Każdy z użytkowników posiada tu swój katalog (o nazwie takiej samej jak jego login), w którym może przechowywać swoje pliki.
- **/lib** – najważniejsze biblioteki systemowe, między innymi moduły jądra, sterowniki urządzeń i inne.
- **/usr** – ten katalog jest drugą najważniejszą sekcją systemu plików. Zawiera dane tylko do odczytu, które mogą być udostępniane w całej sieci. Podkatalogi (bin, lib, sbin) mają takie samo przeznaczenie jak katalogi główne o tej samej nazwie.

- **/usr/share** – dodatkowe pliki programów (dane, rysunki, moduły).
- **/usr/local** – w tym miejscu instalowane są programy dodatkowe, które nie są częścią danej dystrybucji systemu (najczęściej zostały skompilowane ze źródeł przez administratora). Struktura podkatalogów ma te same funkcje co katalogi główne o tej samej nazwie.
- **/usr/src** – źródła systemu (w tym jądra) do samodzielnej kompilacji.
- **/mnt** – w tym katalogu tworzone są punkty montowania zewnętrznych źródeł danych.
- **/sbin** – pliki wykonywalne przeznaczone do użycia przez superużytkowników (administratora systemu i osoby przez niego upoważnione). Zwykli użytkownicy nie mają uprawnień do korzystania z nich.
- **/tmp** – pliki tymczasowe. Każdy użytkownik może tam zapisywać i kasować swoje pliki.
- **/var** – dane aplikacji, takie jak bazy danych, poczta e-mail, strony www.

Jak widać w odróżnieniu od systemów Windows, gdzie każda aplikacja ma swój własny katalog i w nim przechowuje wszystkie swoje pliki, w świecie Unixów wszystkie pliki są rozmieszczane po całym systemie plików, w zależności od ich charakteru. Na przykład biblioteki programu zainstalowanego przez administratora zostaną umieszczone w `/usr/lib`, pliki wykonywalne w `/usr/bin` a dane w `/var`. Taki podział bardzo upraszcza administrowanie systemem, w szczególności tworzenie kopii zapasowych danych (wiadomo, że są one w katalogu `/var`, który trzeba w całości archiwizować) i aktualizowanie programów.

W katalogu `/home` każdy z użytkowników ma swój katalog domowy. Może w nim dowolnie tworzyć, modyfikować oraz kasować pliki i katalogi. Do pozostałych katalogów systemu plików nie ma pełnego dostępu. Może na przykład uruchamiać programy z katalogu `/bin`, ale nie może w tym katalogu zapisywać żadnych informacji.

Oprócz katalogów z powyższego schematu, systemy Unix mogą mieć inne katalogi o przypisanej funkcji. Informacje o nich można uzyskać z dokumentacji systemu (polecenie `man` lub zasoby Internetu).

Nowością dla wcześniejszych użytkowników systemu Windows będzie to, że w systemie plików Unixa, pliki ukryte oznacza się stawiając kropkę przed ich nazwami. Zgodnie z tym plik „Pies” będzie plikiem widocznym, natomiast plik „.Pies” będzie plikiem ukrytym. Aby polecenie `ls` pokazało wszystkie pliki łącznie z ukrytymi, należy użyć przełącznika `-a` (`ls -a`).

Polecenia do obsługi plików

Poniższe polecenia, to dostarczane z systemem programy, które służą do wykonywania operacji na plikach i katalogach. Znajdują się w katalogach `/bin`, `/usr/bin`, `/sbin` lub `/usr/sbin`.

Nazwa polecenia	Przykład użycia	Opis
ls	ls	wyświetla listę plików w bieżącym katalogu (<code>list</code>)
cd	cd /usr/bin	zmienia bieżący katalog na podany (<code>change directory</code>)
	cd ..	wchodzi do katalogu nadrzędnego w stosunku do bieżącego
cp	cp plik1 plik2	kopiuje plik1 i zapisuje go pod nazwą plik2 (<code>copy</code>)
	cp -R /bin /tmp	kopiuje katalog /bin i rekursywnie całą jego zawartość do katalogu /tmp (przełącznika <code>-R</code> można użyć z poleceniami <code>cp</code> , <code>mv</code> i <code>rm</code>)

mv	mv plik1 plik2	przenosi plik1 do plik2 (zmiana nazwy) (move)
	mv /bin/ls /tmp/ls	przenosi plik ls z katalogu /bin do /tmp
rm	rm plik1	usuwa plik1 z dysku (remove)
ln	ln -s plik1 /tmp/plik	tworzy dowiązanie symboliczne (link) o nazwie plik (w katalogu /tmp) do pliku plik1 w bieżącym katalogu
mkdir	mkdir katalog1	tworzy podkatalog o nazwie katalog1 w bieżącym katalogu (make directory)
rmdir	rmdir /tmp/katalog1	usuwa „katalog1“ z katalogu /tmp (katalog1 musi być pusty) (remove directory)
touch	touch plik	tworzy pusty plik
chmod	chmod o+r moj_plik	zmienia prawa dostępu do pliku moj_plik (patrz rozdział o systemie zabezpieczeń) (change mode)
chown	chown uczen moj_plik	zmienia właściciela pliku moj_plik (change owner)
pwd	pwd	pokazuje, w którym katalogu znajduje się aktualnie użytkownik (print working directory)
locate	locate plik	odnajduje w systemie plików wskazany plik i podaje jego lokalizację (wymaga aktualnej bazy danych)
find	find -name „plan.rtf“	przeszukuje system plików w poszukiwaniu pliku o podanej nazwie.
cat	cat plik	wyświetla zawartość pliku

Każde z powyższych poleceń wymaga do działania argumentów (patrz przykłady użycia). Najczęściej są to nazwy plików i przełączniki (np. -X). Nazwy plików mogą być podane na kilka sposobów. Najprościej jest użyć samej nazwy (np. plik1). Wtedy system szuka określonego pliku w bieżącym katalogu i na nim wykonuje operację. Można także podać pełną (bezwzględną) ścieżkę do pliku. Przykładem może być /usr/bin/ps, która oznacza plik o nazwie *ps*, znajdujący się w podkatalogu bin katalogu usr. Ścieżka względna oznacza lokalizację danego pliku względem katalogu bieżącego. Na przykład, będąc w katalogu domowym użytkownika obcy (/home/obcy) można odwołać się do pliku znajdującego się w podkatalogu dokumenty tego katalogu używając ścieżki względnej dokumenty/plan_lekcji.txt. Ścieżka bezwzględnego dostępu do tego pliku wygląda następująco: /home/obcy/dokumenty/plan_lekcji.txt. W jednym poleceniu można użyć każdej i wszystkich opisanych metod. Na przykład, aby skopiować plik znajdujący się w podkatalogu dokumenty bieżącego katalogu do katalogu /tmp w systemie plików Unix, należy użyć polecenia:

```
# cp dokumenty/dane.dat /tmp/dane.dat
```

Można również napisać:

```
# cp /home/obcy/dokumenty/dane.dat /tmp/dane.dat
```

(przy założeniu że polecenie wykonywane jest z katalogu domowego użytkownika obcy). Warto zapamiętać, że ścieżki bezwzględne zawsze zaczynają się od znaku ukośnika (/) a względne nie.

W ścieżce względnej użycie symbolu dwóch kropek (..) oznacza katalog nadrzędny. Na przykład, będąc w katalogu /home/obcy można wydać polecenie:

```
# cd ..
```

co przeniesie użytkownika do katalogu /home. Polecenie:

```
# mv dokumenty/dane.dat ../../dane.dat
```

przeniesie plik dane.dat z podkatalogu dokumenty do katalogu / (dwa poziomy wyżej niż bieżący katalog /home/obcy).

Przykład:

```
# pwd
/home/obcy/dokumenty
# cd ..
# pwd
/home/obcy
# cd ../uczen
# pwd
/home/uczen
```

4.4.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jaką funkcję pełnią poszczególne katalogi w drzewie katalogów systemu plików Unix?
2. Co oznacza katalog domowy? Gdzie się on znajduje?
3. Jakie znasz polecenia do obsługi plików i katalogów w systemie Unix?
4. Co oznacza ścieżka bezwzględna i względna? Jak je rozpoznać?
5. Co określa symbol „..” (dwie kropki)?

4.4.3. Ćwiczenia

Ćwiczenie 1

Utwórz podkatalog o nazwie temp w katalogu domowym.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) sprawdzić, w jakim katalogu znajduje się,
- 2) sprawdzić jego zawartość,
- 3) wpisać polecenie tworzenia katalogu i uzupełnić nazwą podaną w Ćwiczeniu,
- 4) sprawdzić zawartość katalogu.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 2

Utwórz pusty plik modrzew a następnie zmień jego nazwę na brzoza. Skopiuj nowo powstały plik to katalogu tymczasowego /tmp i skasuj plik brzoza w swoim katalogu domowym

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) sprawdzić, w jakim katalogu znajduje się i/lub przejść do katalogu domowego,
- 2) utworzyć pusty plik o podanej nazwie,
- 3) sprawdzić zawartość katalogu,
- 4) zmienić nazwę pliku zgodnie ze specyfikacją,
- 5) skopiować plik do katalogu tymczasowego,
- 6) skasować plik w katalogu domowym,
- 7) sprawdzić zawartość katalogu domowego i katalogu tymczasowego.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 3

Stwórz następującą strukturę katalogów w swoim katalogu domowym:

dom

- | - gabinet
- | - sypialnia
- | - kuchnia
 - | | - lodowka
 - | | - toster
 - | | - kuchenka
- | - pokoj_dzienny

Utwórz pliki o nazwach stosownych do nazwy katalogów w katalogu domowym. Przekopiuj je do odpowiednich katalogów korzystając z różnych lokalizacji.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przejść, jeśli trzeba, do katalogu domowego,
- 2) sprawdzić, jakie nowe katalogi powinny się w nim znaleźć,
- 3) utworzyć je kolejno,
- 4) przejść do katalogu „kuchnia”,
- 5) utworzyć pozostałe katalogi,
- 6) sprawdzić zawartość katalogu domowego,
- 7) sprawdzić zawartość katalogu „kuchnia”,
- 8) utworzyć w katalogu domowym pliki, o których mowa w ćwiczeniu (przynajmniej jeden do każdego katalogu),
- 9) kontynuować pracę w katalogu domowym,
- 10) przekopiować odpowiednie pliki do katalogów gabinet i sypialnia,
- 11) przejść do katalogu kuchnia,
- 12) przekopiować odpowiednie pliki z katalogu domowego do podkatalogów,

- 13) przejść do katalogu pokój_dzienny,
- 14) przekopiować odpowiednie pliki z katalogu domowego,
- 15) wrócić do katalogu domowego,
- 16) sprawdzić rezultat.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

4.4.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) opisać strukturę plików?	<input type="checkbox"/>	<input type="checkbox"/>
2) podać przeznaczenie katalogu home?	<input type="checkbox"/>	<input type="checkbox"/>
3) ukryć nazwę pliku?	<input type="checkbox"/>	<input type="checkbox"/>
4) utworzyć podaną strukturę katalogu?	<input type="checkbox"/>	<input type="checkbox"/>
5) skopiować plik?	<input type="checkbox"/>	<input type="checkbox"/>
6) zmienić katalog?	<input type="checkbox"/>	<input type="checkbox"/>
7) sprawdzić nazwę bieżącego katalogu?	<input type="checkbox"/>	<input type="checkbox"/>

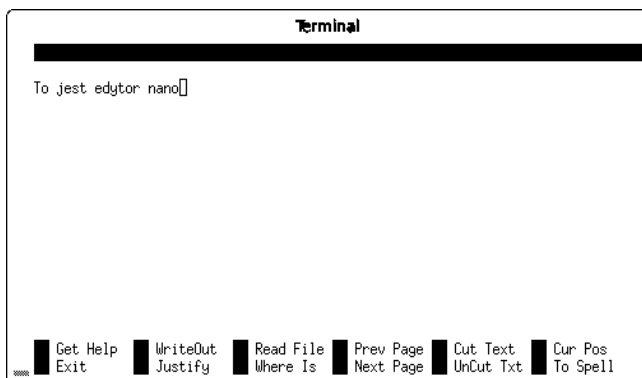
4.5. Programy użytkowe w środowisku Unix

4.5.1. Materiał nauczania

W niniejszym rozdziale przedstawionych zostanie pokrótce kilka przydatnych programów dostępnych z powłoki systemu Unix. Należy zaznaczyć, że nie wszystkie będą dostępne na każdym komputerze z zainstalowanym Unixem. Niektóre mogły zostać wyłączone przez administratora lub nie zostały w ogóle zainstalowane. Instalowanie brakujących programów zostanie opisane w następnych rozdziałach.

nano

Nano jest prostym edytorem tekstu. Pozwala na edycję i tworzenie dowolnych plików tekstowych i zapisywanie ich na twardym dysku. Polecenia w programie wydaje się za pomocą kombinacji klawiszy, których lista pokazana jest zawsze w dolnej części ekranu. Warto wiedzieć, że w notacji komputerowej znak: ^ oznacza klawisz CTRL, więc polecenie zapisania pliku i wyjścia: ^X będzie oznaczało kombinację klawiszy CTRL+X. Aby za pomocą nano otworzyć plik, należy z powłoki systemu wydać polecenie `nano nazwa_pliku` lub `nano /pełna/sciezka/do/pliku.txt`.



mail

Jest to bardzo prosty program pocztowy. Wywołany bez żadnego parametru pozwala na przeczytanie poczty e-mail danego użytkownika zgromadzonej na lokalnym serwerze pocztowym. Wywołany z adresem e-mail jako argumentem, służy do wysłania wiadomości e-mail przez lokalny serwer pocztowy.

mutt

Nieco bardziej rozbudowany program pocztowy. Służy do odbierania i wysyłania poczty z konsoli tekstowej.

wget

Służy do ściągania plików z Internetu. Mimo prostoty jest bardzo dobrym menadżerem pobierania. Obsługuje wznawianie transmisji, protokoły HTTP i FTP oraz oferuje wiele innych ciekawych funkcji. Aby go użyć, należy w konsoli wpisać `wget lokalizacja`, gdzie lokalizacja to pełna ścieżka do pliku w Internecie, włącznie z oznaczeniem protokołu na początku (np. `http://www.wp.pl/index.php`)

tar

Program archiwizujący. Pozwala na archiwizowanie i pakowanie plików w archiwa „tar” (tape archive format) (nie skompresowane), „.tgz” lub „.tar.gz” (skompresowane). Przykłady użycia:


```
# tar -czf archiwum.tgz /home/uczen
```

tworzy archiwum o nazwie „archiwum.tgz” i pakuje do niego katalog /home/uczen.

```
# tar -zxf archiwum.tgz
```

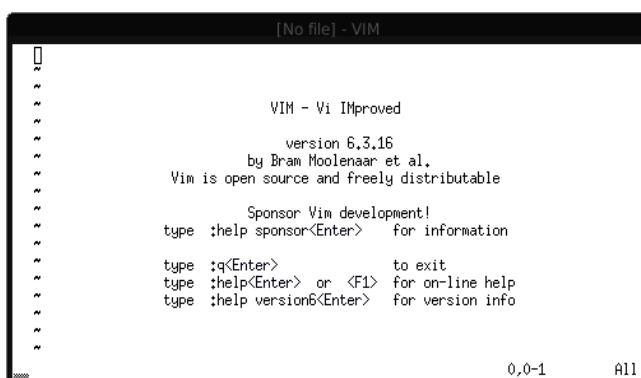
rozpakowuje zawartość pliku „archiwum.tgz” do bieżącego katalogu.

gzip i gunzip

Programy służą do kompresowania (gzip) (GNU **zip**) i dekompresowania (gunzip) plików. Jako argument należy podać nazwę pliku, który ma być skompresowany lub zdekompresowany.

vim

vim (**vi improved**) to bardziej ambitny niż nano edytor tekstów. Jest ulepszoną wersją znanego i popularnego edytora vi (**visual in ex**). Praca w nim odbywa się w trzech trybach: wpisywania, nadpisywania i komend. Tryb wpisywania uaktywnia się naciskając na klawiaturze przycisk I. Wtedy vim działa jak inne edytory tekstu i pozwala wprowadzać znaki. w trybie nadpisywania (Shift+R), Vim nadpisuje znaki nad aktualnie znajdującymi się pod kursorem (analogicznie jak inne edytory po przełączeniu w tryb nadpisywania – Replace. W trybie komend (znak dwukropka na klawiaturze) edytorowi można wydawać polecenia. Polecenie help powoduje wyświetlenie pomocy. Polecenie: w nazwa_pliku spowoduje zapisanie pliku pod nazwą nazwa_pliku, a polecenie: q spowoduje wyjście z programu. Z każdego trybu wychodzi się naciskając przycisk ESC. Nie wchodząc do żadnego z powyższych trybów można wykonywać niektóre czynności edycyjne, naciskając przyciski z literami: **x** – kasuje jeden znak, **dd** – kasuje całą linię, **o** – przechodzi do następnej linii i włącza tryb wpisywania, **p** – wstawią linię usuniętą wcześniej przez dd.



emacs

Kolejny edytor tekstu, ceniony przez programistów (editor **macros**).

whereis

Polecenie whereis podaje pełną, bezwzględną ścieżkę do programu podanego jako parametr. Np. polecenie whereis vim na większości komputerów zwróci na ekranie: /usr/bin/vim. Oznacza to, że program vim znajduje się w katalogu /usr/bin.

lynx

Mała, ale funkcjonalna przeglądarka WWW, pracująca w trybie tekstowym.

4.5.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. W jaki sposób otworzysz plik do edycji za pomocą programu nano?
2. Jakich programów użyjesz do sprawdzenia poczty e-mail z konsoli tekstowej?
3. Jakie programy służą do obsługi skompresowanych plików?
4. Jakie tryby obsługuje edytor vim? Jak się między nimi przełączać?
5. W jakich sytuacjach skorzystasz z polecenia whereis?

4.5.3. Ćwiczenia

Ćwiczenie 1

Przedstaw główne cechy edytora emacs. Udostępnij utworzony plik.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wejść na stronę Wikipedii,
- 2) zapoznać się z informacjami na temat edytora emacs,
- 3) uruchomić edytor emacs i zapisać jego cechy korzystając z wyszukanych informacji,
- 4) zapisać i udostępnić plik innym użytkownikom.

Wyposażenie stanowiska pracy:

- komputer z edytorem tekstu i dostępem do sieci Internet.

Ćwiczenie 2

Wyślij wiadomość e-mail na adres podany przez nauczyciela, korzystając z dowolnego programu pocztowego dostępnego z konsoli tekstowej.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) otworzyć konsolę i wpisać polecenie uruchamiające wybrany program pocztowy,
- 2) na innej konsoli wyświetlić informację podręcznikową na temat tego programu,
- 3) zwrócić szczególną uwagę na sposób poinformowania programu o zakończeniu wpisywania treści wiadomości,
- 4) wysłać wiadomość i poprosić o sprawdzenie doręczenia.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD, serwerem pocztowym i programem pocztowym.

Ćwiczenie 3

Utwórz w edytorze vim plik tekstowy o nazwie „wizytowka.txt” w swoim katalogu domowym. Umieść w nim swoje imię, nazwisko, klasę i nazwę szkoły.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) w konsoli wpisać polecenie vim aby uruchomić edytor,
- 2) przejść do trybu wpisywania naciskając klawisz I,
- 3) wpisać wszystkie wymagane dane,
- 4) na zakończenie nacisnąć ESC, a następnie zapisać plik wpisując z klawiatury odpowiednie polecenie,
- 5) sprawdzić, czy plik został zapisany,
- 6) wyjść z programu.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i programem vim.

4.5.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) wymienić kilka edytorów tekstu?	<input type="checkbox"/>	<input type="checkbox"/>
2) wymienić programy pocztowe?	<input type="checkbox"/>	<input type="checkbox"/>
3) wymienić narzędzie pobierania plików?	<input type="checkbox"/>	<input type="checkbox"/>
4) zarchiwizować pliki?	<input type="checkbox"/>	<input type="checkbox"/>
5) rozpakować archiwum?	<input type="checkbox"/>	<input type="checkbox"/>
6) znaleźć ścieżkę dostępu do programu o podanej nazwie?	<input type="checkbox"/>	<input type="checkbox"/>
7) obejrzeć stronę www w trybie tekstowym?	<input type="checkbox"/>	<input type="checkbox"/>

4.6. Powłoka systemu i skrypty

4.6.1. Materiał nauczania

Powłoka (shell), to program tłumaczący polecenia wydawane przez użytkownika na konsoli, na rozkazy zrozumiałe dla jądra systemu. Z powłoki można uruchamiać programy wpisując ich nazwę (lub pełną ścieżkę) oraz wykonywać operacje na plikach. Istnieją programy wspomagające pracę z plikami, takie jak Midnight Commander (polecenie mc), ale nie zawsze są one zainstalowane na komputerze. Warto więc znać podstawowe polecenia powłoki.

Istnieje wiele powłok. Różnią się listą dostępnych poleceń, składnią niektórych z nich i innymi szczegółami. Do najpopularniejszych należą Korn Shell (ksh), Berkley C Shell (csh) i Bourne-Again Shell (bash, sh). Powłokę można wybrać wykonując polecenie `chsh` z konsoli. System zapyta o pełną ścieżkę do powłoki (przyda się `whereis`) lub w edytorze `vi` (podobny do `vim`) otworzy plik z danymi konta użytkownika, wśród których znajduje się wybrana powłoka (Shell:)

Aby zautomatyzować wykonywane w powłoce czynności, można pisać skrypty. Są to specjalnie przygotowane pliki tekstowe, zawierające kolejne polecenia, które powłoka ma wykonać. Mogą to być nazwy programów lub wewnętrzne polecenia danej powłoki. Możliwe jest też wstawianie komentarzy (linie zaczynające się od znaku #), które nie będą interpretowane przez powłokę, ale ułatwią zrozumienie działania skryptu. Każdy skrypt musi zaczynać się od deklaracji, czyli specjalnej linii składającej się ze znaków „#!“, oraz ścieżki do powłoki, która ma wykonywać dany skrypt.

Przykład skryptu dla powłoki sh podany jest poniżej:

```
#!/bin/sh
echo "Witaj swiecie"
sleep 2
echo "To jest koniec skryptu"
```

Skrypt ten wypisze na ekranie frazę „Witaj swiecie“ (polecenie powłoki: `echo`), odczeka 2 sekundy bezczynnie (polecenie: `sleep`) a na koniec wypisze „To jest koniec skryptu“. Pierwsza linia pełni rolę deklaracji. Zakłada ona, że plik powłoki `sh` znajduje się w katalogu `/bin`. Jeśli skrypt nie będzie działał, należy poleceniem `whereis sh` sprawdzić lokalizację tej powłoki i poprawić deklarację. Polecenia `echo` i `sleep` to polecenia powłoki, których można użyć zarówno w skryptach jak i bezpośrednio z konsoli.

Skrypt należy napisać za pomocą dowolnego edytora tekstów (np. `vim` lub `nano`) i zapisać w pliku. Ostatnim etapem jest nadanie plikowi uprawnień do wykonywania (aby system wiedział, że jest to plik wykonywalny, możliwy do uruchomienia). Można użyć polecenia:

```
# chmod u+x nazwa_skryptu
```

Uruchomienie skryptu odbywa się poprzez wpisanie z konsoli jego pełnej ścieżki, lub jeśli znajduje się on w bieżącym katalogu, poprzez podanie jego nazwy poprzedzonej sygnaturą bieżącego katalogu (`./`), np:

```
# ./nazwa_skryptu
```

lub

```
# /home/uczen/nazwa_skryptu
```

W skryptach powłoki często używa się funkcji warunkowych. Powodują one wykonanie jakiejś instrukcji pod warunkiem zajścia jakiegoś zdarzenia lub uzyskania wartości „prawda” po obliczeniu wartości wyrażenia logicznego. W powłoce sh ma ona składnię:

if [warunek]; **then** instrukcja; **fi**

Przy budowaniu wyrażeń logicznych (warunków) korzysta się z operatorów testujących.

Tabela 1. Operatory testujące.

operator	znaczenie		prawda, gdy:	podobne operatory
	english	polski		
-n	non zero	nie zero	argument ma nie-zerową długość	
-z	zero	zero	argument ma zerową długość	
-d	directory	katalog	istnieje katalog o nazwie podanej jako argument	
-f	file	plik	istnieje plik o nazwie podanej jako argument	
-eq	equal	równy	argumenty są typu całkowitego i są równe	-neq (nie są równe)
=			argumenty typu string są równe	!= (nie są równe)
-lt	later	mniejszy	pierwszy argument typu całkowitego jest mniejszy niż drugi	-gt (większy) -ge (większy lub równy) -le (mniejszy lub równy)

Poniższy skrypt sprawdza, czy istnieje plik o nazwie „plik1” i wyświetla odpowiedni komunikat:

```
#!/bin/sh
if [ -f "plik1" ]; then
    echo "Plik istnieje"
else
    echo "Pliku nie ma"
fi
```

Inny przykładowy skrypt wypisze na ekranie aktualną datę i godzinę (program date), następnie uruchomi program pocztowy (mail) w celu odczytania poczty użytkownika, a po jego zakończeniu wyczyści ekran (polecenie clear).

```
#!/bin/sh
# linie zaczynające się od znaku komentarza
# nie są brane przez skrypt pod uwagę

# polecenie date podaje aktualny czas (możemy podać
# pełną ścieżkę do programu date)
/bin/date
# uruchamiamy program pocztowy (możemy również podać
# samą nazwę programu)
mail
# na koniec wyczyścimy ekran
clear
```

Ważnym z punktu widzenia skryptów elementem każdej powłoki jest dostęp do zmiennych środowiskowych. Niektóre z nich przechowują informacje wykorzystywane przez powłokę, na przykład zmienna `PATH` w większości powłok zawiera informacje o adresach (ścieżka), gdzie znajdują się pliki wykonywalne. Dlatego, aby uruchomić program nie trzeba za każdym razem wpisywać jego pełnej ścieżki. O zmiennych, ich przeznaczeniu i metodach dostępu do nich warto przeczytać z dokumentacji konkretnych implementacji powłok i z obszerniejszych publikacji na temat systemu Unix.

4.6.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie znasz powłoki systemu Unix?
2. Do czego służą skrypty powłoki i jak się ich używa?
3. Jak zastosować w skrypcie powłoki `sh` instrukcję warunkową?

4.6.3. Ćwiczenia

Ćwiczenie 1

Na swoim koncie zmień powłokę na `tcsh`.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) sprawdzić, w jakim katalogu znajduje się `tcsh`. w powłoce wpisać `chsh` i w odpowiednie miejsce formularza wpisać pełną ścieżkę do pliku `tcsh`,
- 2) zatwierdzić zmiany,
- 3) powtórzyć polecenie `chsh` i sprawdzić przypisaną nazwę powłoki,
- 4) można przywrócić pierwotne ustawienia.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 2

Napisz skrypt, który wyświetli aktualną datę, a następnie zawartość pliku `/etc/motd`, wreszcie, po upływie 5 sekund, wyczyści ekran.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) uruchomić edytor tekstu,
- 2) wpisać linię identyfikującą powłokę,
- 3) dopisać komentarz (również do wszystkich pozostałych poleceń),
- 4) w następnej linii wpisać polecenie wyświetlające datę systemową,
- 5) w kolejnej linii wpisać polecenie pokazujące na ekranie zawartość pliku, o którym mowa w Ćwiczeniu,
- 6) dopisać polecenie wstrzymujące pracę na 5 sekund. W ostatniej linii pliku wpisać polecenie czyszczące ekran,
- 7) zapisać plik,
- 8) nadać uprawnienia umożliwiające wykonywanie pliku,
- 9) sprawdzić działanie.

- Wyposażenie stanowiska pracy:
- komputer z zainstalowanym systemem FreeBSD.

4.6.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) wymienić nazwy powłok?	<input type="checkbox"/>	<input type="checkbox"/>
2) napisać prosty skrypt?	<input type="checkbox"/>	<input type="checkbox"/>
3) zdefiniować uprawnienia do pliku?	<input type="checkbox"/>	<input type="checkbox"/>
4) uruchomić skrypt?	<input type="checkbox"/>	<input type="checkbox"/>
5) komentować polecenia w skrypcie?	<input type="checkbox"/>	<input type="checkbox"/>

4.7. Strumienie, potoki i filtry

4.7.1. Materiał nauczania

W Unixie z każdym procesem związane są przynajmniej trzy tzw. strumienie:

- **stdin** – standardowy strumień wejściowy związany z klawiaturą. Z niego pobierane są znaki wpisywane przez użytkownika (standard input). Posiada oznaczenie 0.
- **stdout** – standardowy strumień wyjściowy związany z ekranem. Ten strumień reprezentuje wszystkie dane wyświetlane przez program (standard output). Oznaczany jest numerem 1.
- **stderr** – standardowy strumień błędów, który zwykle również kojarzony jest z ekranem, chociaż jest oddzielnym strumieniem, oznaczanym numerem 2. Programy kierują na niego wszystkie komunikaty o błędzie (standard error).

Dzięki zastosowaniu strumieni, programy są bardziej elastyczne. Nie są na stałe związane z konkretnym urządzeniem wejścia i wyjścia (klawiatura i ekran). Użytkownik może decydować zarówno skąd pobierane są dane, jak i gdzie skieruje wyniki swojej pracy.

Aby przekierować strumień danych wyjściowych polecenia `ls` do pliku, należy użyć operatora „>” i podać nazwę pliku wyjściowego. Na przykład polecenie:

```
# ls /etc
```

wyświetli zawartość katalogu `/etc` na ekranie (konkretniej, na standardowym urządzeniu wyjścia `stdout`, które domyślnie związane jest z ekranem).

Natomiast polecenie:

```
# ls /etc > /tmp/lista.txt
```

spowoduje zapisanie listy plików z katalogu `/etc` do pliku `/tmp/lista.txt`.

Należy zwrócić uwagę, że w razie wystąpienia błędów (np. katalog `/etc` nie istnieje), komunikaty o nich nie zostaną zapisane do pliku `/tmp/lista.txt`, a wyświetlone na ekranie. Spowodowane jest to tym, że przekierowanie do pliku dotyczy tylko strumienia `stdout`, a błędy wysyłane są nadal do strumienia `stderr`. Strumień `stderr` przekierowuje się analogicznie, wykorzystując operator „2>”. Aby przekierować obydwa strumienie do tego samego pliku należy użyć operatora „&>”:

```
# ls /etc > /tmp/lista.txt 2> /tmp/bledy.txt
```

```
# ls /etc &> /tmp/lista.txt
```

Przekierowania powyższe powodowały utratę danych w plikach, na które wskazują strumienie. Używając podobnych operatorów „>>”, „2>>” i „&>>” można dopisać odpowiednie strumienie do istniejących plików.

Czasami przydatne jest przekierowanie strumienia wejściowego (`stdin`). Wówczas wykonywany proces zamiast pobierać znaki z klawiatury, będzie je pobierał z pliku. Takie przekierowanie realizuje się za pomocą operatora „<”.

Załóżmy, że plik `/tmp/wejscie` wygląda następująco:

```
echo Start!  
ls /  
echo Koniec!
```

Wówczas polecenie:

```
# bash < /tmp/wejscie
```

uruchomi powłokę `bash`, która zamiast przyjmować polecenia od użytkownika z klawiatury, wczyta je z pliku.

Czasami przydaje się przekierować strumień wyjściowy jednego programu do strumienia wejściowego innego programu. Służy do tego operator „|” (pionowa kreska). Można w ten sposób wykorzystać takie programy (filtry) jak `head` (pokazuje kilka pierwszych linii strumienia wejściowego), `tail` (pokazuje kilka ostatnich linii strumienia wejściowego), `grep` (pokazuje tylko te linie strumienia wejściowego, które zawierają słowo podane jako parametr polecenia `grep`), czy `sort` (sortuje linie strumienia wejściowego). Na przykład:

```
# ls /etc | sort
```

pokaże nazwy plików w katalogu `/etc` posortowane w porządku alfabetycznym, natomiast

```
# ls /etc | sort | head
```

pokaże tylko kilka pierwszych linii z listy plików katalogu `/etc`, posortowanej w kolejności alfabetycznej.

Można przefiltrować wyjście polecenia `ls /` tak, aby pokazało ono tylko nazwy plików i katalogów zawierających w sobie frazę `bin` (w głównym katalogu systemu plików są dwa takie katalogi: `/bin` oraz `/sbin`) za pomocą filtra `grep`, poleceniem:

```
# ls / | grep bin
```

Istnieją też programy nazywane pagerami. Należą do nich między innymi `more` i `less`. Ich zadaniem jest umożliwienie użytkownikowi przejrzania długiego strumienia wejściowego, nie mieszczącego się na ekranie, za pomocą strzałek góra/dół na klawiaturze. Przykładem zastosowania tego polecenia niech będzie listing katalogu `/dev`, który zawiera pliki powiązane ze wszystkimi urządzeniami dostępnymi w systemie i jest zawsze bardzo długi.

```
# ls /dev | less
```

Istnieje możliwość rozgałęzienia potoku. Realizowane jest za pomocą polecenia `tee`, które można umieścić w dowolnym miejscu potoku.

4.7.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Co to jest strumień, a co to jest potok?
2. Jakich operatorów używa się do przekierowywania strumieni?
3. Jakie filtry wykorzystuje się przy przetwarzaniu strumieni wyjściowych w potokach?

4.7.3. Ćwiczenia

Ćwiczenie 1

Zapisz listę uruchomionych procesów w pliku `/tmp/procesy.lst`

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wpisać polecenie, które wyświetli listę procesów,
- 2) zapamiętać wygląd ekranu,
- 3) ponowić polecenie (strzałka w górę) przekierowując rezultat do wskazanego pliku,
- 4) obejrzeć zawartość pliku,
- 5) porównać z zapamiętanym obrazem.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 2

Zapisz posortowaną alfabetycznie listę plików z katalogu /etc, które zawierają w nazwie frazę „.conf“, do pliku /tmp/konfiguracja.lst.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wyświetlić na ekranie listę plików znajdujących się w podanym katalogu,
- 2) zapamiętać kilka o nazwach zaczynających się na początkowe litery alfabetu,
- 3) powtórzyć polecenie dodając polecenie sortowania,
- 4) powtórzyć polecenie uzupełniając o filtr wybierający podaną frazę,
- 5) zapamiętać fragment wyświetlanej informacji,
- 6) powtórzyć polecenie przekierowując rezultat do pliku o podanej nazwie i lokalizacji,
- 7) obejrzeć zawartość pliku,
- 8) porównać z zapamiętanym obrazem.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 3

Do pliku /tmp/konfiguracja.lst z poprzedniego ćwiczenia dopisz linię zawierającą frazę „Koniec pliku“

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) obejrzeć ostatnie linie pliku utworzonego w poprzednim ćwiczeniu,
- 2) zapamiętać,
- 3) napisać na konsoli polecenie wyświetlenia podanego tekstu i uzupełnić operatorem dołączania do pliku,
- 4) obejrzeć ponownie końcowe linie pliku, o którym mowa w ćwiczeniu,
- 5) zlokalizować przekierowaną frazę.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD..

4.7.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) wymienić strumienie?	<input type="checkbox"/>	<input type="checkbox"/>
2) przekierować rezultat działania polecenia do pliku?	<input type="checkbox"/>	<input type="checkbox"/>
3) wczytać polecenia powłoki z pliku?	<input type="checkbox"/>	<input type="checkbox"/>
4) zastosować filtr sortujący dane?	<input type="checkbox"/>	<input type="checkbox"/>
5) obejrzeć nagłówki strumienia wejściowego?	<input type="checkbox"/>	<input type="checkbox"/>
6) wyszukać frazę w strumieniu wejściowym?	<input type="checkbox"/>	<input type="checkbox"/>
7) stronicować strumień wejściowy?	<input type="checkbox"/>	<input type="checkbox"/>

4.8. Konta użytkowników i proces logowania

4.8.1. Materiał nauczania

Każdy użytkownik korzystający z systemu musi mieć swoje konto. Systemy z rodziny BSD przechowują informacje o kontach w pliku `/etc/master.passwd`, natomiast systemy Linux w plikach `/etc/passwd` i `/etc/shadow` (zaszyfrowane hasła). Pliki te są tak skonstruowane, że jednemu użytkownikowi odpowiada jedna linijka. Pola danych oddzielone są znakiem dwukropka. Jako pierwsze pole każdej linijki występuje zawsze nazwa użytkownika (czyli login) składająca się ze znaków alfanumerycznych – liter i cyfr – i ewentualnego podkreślnika, myślnika lub kropki. Dostępny zestaw znaków jest zależny od systemu. W kolejnych polach znajdują się: UID, czyli numeryczny identyfikator użytkownika, GID, czyli numer grupy, do której użytkownik ten należy, jego katalog domowy oraz domyślna powłoka. Hasła w postaci zaszyfrowanej przechowywane są jako drugie pole pliku `/etc/master.passwd` w systemach BSD (w tym FreeBSD), lub w osobnym pliku `/etc/shadow`, np. na Linuxach. Istnieje również metoda przechowywania informacji o kontach w specjalnej bazie danych dostępnej dla wszystkich komputerów w sieci NIS. Można o nich przeczytać w poważnych pozycjach książkowych poświęconych administrowaniu Unixami w środowisku sieciowym.

Administrator przypisuje użytkowników do grup, które są określone w pliku `/etc/group`. Każdy użytkownik musi mieć zdefiniowaną swoją grupę główną (grupę logowania) i opcjonalnie dodatkowe grupy, do których należy.

Do tworzenia kont użytkowników w systemie FreeBSD służy interaktywny skrypt `adduser`, lub skrypt `pw` obsługiwany z linii poleceń. Szczegółowe informacje na temat tych dwóch skryptów dostępne są w pomocy systemowej `man`.

Aby zmienić własne hasło należy użyć polecenia `passwd`. System zapyta o stare hasło w celu potwierdzenia tożsamości, dwukrotnie o nowe w celu weryfikacji błędów literowych i po sprawdzeniu czy nowe hasło spełnia wymagania bezpieczeństwa, zmieni odpowiedni wpis. Administrator może użyć polecenia `passwd` w celu zmiany haseł innych użytkowników, dodając jako parametr nazwę tego użytkownika. Nie pojawi się pytanie o stare hasło użytkownika. Do usuwania kont użytkowników używa się polecenia `rmuser` lub `pw`.

Aby tworzyć lub usuwać konta, trzeba być oczywiście zalogowanym jako administrator (`root`).

Procedura logowania

Przy próbie zalogowania się system sprawdza, czy podane hasło zgadza się z zapisanym w odpowiednich plikach. Jeśli tak, to uruchomiona zostanie powłoka użytkownika, a jej standardowe wejście i wyjście zostanie przypisywane do konsoli użytkownika. Po uruchomieniu powłoki uruchamiane zostaną też polecenia zawarte w skryptach startowych. Są to w zależności od powłoki pliki `.login`, `.profile`, lub `.cshrc`.

Aby spowodować, że pewien program uruchomi się zaraz po zalogowaniu się użytkownika, należy do któregoś z jego plików startowych dodać linię z nazwą programu, np:

```
echo "fortune" >> /home/uczen/.cshrc
```

spowoduje pokazanie śmiesznej sentencji z programu `fortune` za każdym razem, kiedy użytkownik `uczen` zaloguje się.

Użytkownik może nie tylko pracować na kilku konsolach, ale też zalogować się korzystając z różnych kont (na przykład o różnych uprawnieniach). Pracując w takim systemie warto mieć narzędzie pozwalające ustalić tożsamość. Przydatne będą dwa proste polecenia: `whoami` i `who`. Pierwsze z nich wyświetla nazwę użytkownika, a drugie listę zalogowanych użytkowników z informacjami o używanych konsolach i czasie logowania.

4.8.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie polecenia służą do zarządzania kontami użytkowników w systemie FreeBSD?
2. W jakich plikach przechowywane są informacje o kontach?
3. Jakie czynności wykonywane są podczas logowania użytkownika? W jaki sposób wpływają na zachowanie systemu?

4.8.3. Ćwiczenia

Ćwiczenie 1

Utwórz konto użytkownika janosik za pomocą skryptu adduser. Zbadaj możliwości kreowania konta.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wprowadzić proponowane polecenie dodawania użytkownika,
- 2) wypełnić według poniższego wzoru formularz rejestracyjny polecenia (może wpisać inne dane),

```
# adduser
Username: janosik <ENTER>
Full name: Jan Osik <ENTER>
Uid (Leave empty for default): <ENTER>
Login group [janosik]: users <ENTER>
Login group is users. Invite janosik into other groups? []:<ENTER>
Login class [default]: <ENTER>
Shell (sh csh tcsh rssh nologin) [sh]: tcsh <ENTER>
Home directory [/home/janosik]: <ENTER>
Use password-based authentication? [yes]: <ENTER>
Use an empty password? (yes/no) [no]: <ENTER>
Use a random password? (yes/no) [no]: <ENTER>
Enter password: sekret <ENTER>
Enter password again: sekret <ENTER>
Lock out the account after creation? [no]: <ENTER>
Username   : janosik
Password   : *****
Full Name  : Jan Nosik
Uid        : 1027
Class      :
Groups     : users
Home       : /home/janosik
Shell      : /bin/tcsh
Locked     : no
OK? (yes/no): yes <ENTER>
adduser: INFO: Successfully added (janosik) to the user database.
Add another user? (yes/no): no <ENTER>
Goodbye!
```

- 3) zanotować na kartce najistotniejsze dane,
- 4) zalogować się jako użytkownik janosik na innej konsoli,
- 5) wrócić do głównej konsoli.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 2

Zmień hasło użytkownika janosik i zaprojektuj dla niego komunikat powitalny.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wprowadzić polecenie umożliwiające zmianę hasła użytkownika janosik,
- 2) dopisać na końcu hasła cyfrę 2,
- 3) sprawdzić rezultat,
- 4) przekierować do jednego ze skryptów startowych frazę powitalną,
- 5) sprawdzić wyświetlanie komunikatu powitalnego.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

4.8.4. Sprawdzian postępów

Czy potrafisz:

	Tak	Nie
1) przedstawić sposób przechowywania kont użytkowników?	<input type="checkbox"/>	<input type="checkbox"/>
2) zmienić swoje hasło?	<input type="checkbox"/>	<input type="checkbox"/>
3) utworzyć konto użytkownika?	<input type="checkbox"/>	<input type="checkbox"/>
4) uruchomić aplikację przy logowaniu użytkownika?	<input type="checkbox"/>	<input type="checkbox"/>

4.9. Zarządzanie procesami

4.9.1. Materiał nauczania

Czasami zachodzi potrzeba wykonania na raz kilku czynności, na przykład odebrania poczty e-mail i napisania notatki w edytorze tekstu. W takich sytuacjach wygodne jest uruchomienie na raz kilku programów i praca z nimi. Ograniczeniem jest fakt, że tylko jeden program może pracować na pierwszym planie, to znaczy widać jego wyjście na ekranie i można wydawać mu rozkazy z klawiatury. Pozostałe programy pracują w tle, ale w każdej chwili można się do nich przełączyć.

Najlepiej rozpatrzyć to na przykładzie. Załóżmy, że uruchomiony jest edytor tekstu Vim. Można w nim pisać dowolny tekst. Aby przenieść Vim w tło, należy nacisnąć kombinację klawiszy CTRL+Z, a następnie wpisać polecenie `bg` z parametrem `1`, co przypisze programowi identyfikator o wartości `1`. Powłoka zgłosi w konsoli stan procesu (Stopped – Zatrzymany). Dzieje się tak, ponieważ edytor tekstu wymaga, aby jego stdin i stdout były podłączone do konsoli, a gdy nie jest to możliwe (program działa w tle), powłoka zatrzyma jego działanie. Listę zadań w tle można obejrzeć za pomocą polecenia `jobs`. Powinien pojawić się proces właśnie przeniesiony w tło z przypisanym identyfikatorem `[1]`. Można teraz uruchomić nowe zadanie lub przywołać proces z tła na pierwszy plan za pomocą polecenia `fg` z parametrem odpowiadającym procesowi, w tym przypadku: `1`.

Zadania wyświetlone za pomocą `jobs` można wyłączyć poleceniem `kill %x`, gdzie `x` oznacza numer zadania. Przy okazji warto wspomnieć o kombinacji CTRL+C, która permanentnie zabija proces działający na pierwszym planie.

Screen – menadżer okien

Aby móc pracować z dwoma programami równocześnie, warto zapoznać się z programem `screen`. `Screen` jest pełnoekranowym menadżerem okien. Umożliwia otwarcie kilku wirtualnych terminali na jednej konsoli a następnie przełączanie się między nimi kombinacją klawiszy. Po uruchomieniu, `screen` otwiera pierwsze okno (właściwie okno z numerem 0) i uruchamia w nim domyślną konsolę. Aby otworzyć kolejne okno należy skorzystać z kombinacji klawiszy CTRL+A i nacisnąć C. Właściwie wszystkie polecenia programu `screen` składają się z kombinacji CTRL+A, a następnie jednego lub dwóch przycisków z klawiatury.

Aby zobaczyć listę otwartych okien należy nacisnąć CTRL+A, a potem W. Lista ukaże się w lewym dolnym rogu ekranu, a przy aktualnie otwartym oknie widoczna będzie gwiazdka. Przełączanie między oknami odbywa się poprzez CTRL+A i numer okna. Można też przełączać się kombinacjami CTRL+A i N, oraz CTRL+A i P dla okna następnego i poprzedniego. W każdym oknie można uruchomić inny program, który będzie działał nawet po przełączeniu się do innego okna. Każdą powłokę (i okno jej towarzyszące) wyłącza się poleceniem `exit`. Po zamknięciu wszystkich okien `screen` zgłosi zakończenie swojej pracy i wyłączy się.

4.9.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Co oznacza, że proces działa w tle?
2. Jakie polecenia służą do przełączania się między procesami?
3. Jak przełączać się między procesami za pomocą programu `screen`?

4.9.3. Ćwiczenia

Ćwiczenie 1

Uruchom jednocześnie edytor tekstu nano i edytor vim. Wyłącz nano, a następnie poleceniem kill zabij proces edytora vim.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) uruchomić program nano,
- 2) nacisnąć CTRL+Z i zapisać jako pierwszy proces w tle,
- 3) sprawdzić procesy w tle,
- 4) uruchomić edytor vim i przenieść go w tło jako drugi proces,
- 5) sprawdzić procesy,
- 6) przywołać nano na pierwszy plan,
- 7) wyłączyć w normalny sposób,
- 8) sprawdzić procesy,
- 9) zakończyć działanie edytora vim poleceniem `kill`,
- 10) sprawdzić procesy.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i programami vim i nano.

Ćwiczenie 2

Za pomocą programu screen uruchom na raz trzy programy nano. W każdym z nich otwórz inny plik konfiguracyjny z katalogu /etc.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) uruchomić screen,
- 2) w pierwszym oknie sprawdzić listę plików w katalogu /etc,
- 3) uruchomić edytor nano i otworzyć jeden z plików z tego katalogu,
- 4) włączyć następne okno screen'a,
- 5) uruchomić nano z kolejnym plikiem,
- 6) otworzyć trzecie okno, a w nim nano z trzecim plikiem,
- 7) przeglądać zawartość okien podając odpowiedni numer,
- 8) przeglądać okna sekwencyjnie najpierw wybierając „następne”, a później „poprzednie”,
- 9) zamknąć okna.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i programami screen oraz nano.

4.9.4. Sprawdzian postępów

Czy potrafisz:

- | | Tak | Nie |
|---|--------------------------|--------------------------|
| 1) uruchomić zadanie w tle? | <input type="checkbox"/> | <input type="checkbox"/> |
| 2) sprawdzić, jakie programy pracują w tle? | <input type="checkbox"/> | <input type="checkbox"/> |
| 3) skorzystać z menadżera screen? | <input type="checkbox"/> | <input type="checkbox"/> |
| 4) zakończyć wybrany program poleceniem kill? | <input type="checkbox"/> | <input type="checkbox"/> |

4.10. Montowanie urządzeń i inne systemy plików

4.10.1. Materiał nauczania

Na dużych stacjach roboczych i serwerach sieciowych bardzo często umieszcza się każdy z katalogów głównych systemu plików (/usr, /etc, /home itd) na innej partycji dysku. Zapobiega to utracie danych na wypadek awarii i podnosi wydajność systemu. Poszczególne partycje montuje się w kolejnych katalogach. Pojęcie to oznacza, że jeżeli pewna partycja zamontowana jest w katalogu /usr, to wszystkie pliki i podkatalogi w tym katalogu znajdują się na tej partycji.

Podobnie montuje się zewnętrzne systemy plików, na przykład dyskietki, płyty CD i DVD, napędy taśmowe, zewnętrzne dyski twarde itd. Do ich montowania stworzono specjalny katalog /mnt. W nim administratorzy tworzą podkatalogi takie jak /mnt/dyskietka, /mnt/cdrom i w nich montują poszczególne urządzenia. Informacje o zamontowanych systemach plików uzyskuje się uruchamiając polecenie mount bez parametrów. Warto też zajrzeć do pliku /etc/fstab. Tam zapisane są informacje o systemach plików montowanych bądź zdefiniowanych przy starcie systemu.

Dobrym źródłem informacji jest też dokumentacja systemu, dostępna na przykład na stronie www.freebsd.org. Na temat napędów można tam znaleźć informacje przedstawione w zarysie w poniższej tabeli:

Rodzaj napędu	Nazwa urządzenia
IDE	ad
IDE CDROM	acd
SCSI i USB Mass storage device	da
SCSI CDROM	cd
Napędy dysków elastycznych	fd

Montowanie systemu plików

```
# mount -t typ urządzenie punkt_montowania
```

Poszczególne parametry mają następujące znaczenie:

- **typ** - typ systemu plików. Standardowo dyski FreeBSD używają systemu plików ufs, dyski systemu Windows: msdos, a dyski CD-ROM: cd9660.
- **urządzenie** - ścieżka do odpowiedniego urządzenia w katalogu /dev. Standardowo stacja dyskietek to /dev/fd0, CD-ROM to /dev/acd0 a kolejne dyski twarde to /dev/adXs1Y, gdzie X to kolejny numer dysku twardego a Y to litera przypisana do konkretnej partycji.
- **punkt_montowania** - katalog, w którym ma być zamontowany system plików, np. /mnt/dyskietka.

Odmontowywanie

Przed odłączeniem zewnętrznego systemu plików trzeba go odmontować. Zapobiegnie to utracie danych spowodowanej niedokończonymi operacjami zapisu.

```
# umount urządzenie
```

lub

```
# umount punkt_montowania
```


Formatowanie

Dyskietkę można sformatować poleceniem `fdformat /dev/fd0`. Następnie należy utworzyć nowy system plików:

```
# newfs urządzenie
```

Dyski, partycje i plastry

Dyski IDE w systemie FreeBSD reprezentowane są przez urządzenia blokowe w katalogu `/dev` o nazwach `ad0`, `ad1`, `ad2` itd. Zwykle na dysku FreeBSD tworzy się jedną partycję, ponieważ będzie potem ona podzielona na plastry (partycje BSD) programem `disklabel`. Plastry oznaczane są jako `ad0s1a`, `ad0s1b`, `ad0s1d` itd. Zmienia się tylko ostatnia litera, z wyłączeniem „c”, ponieważ „c” reprezentuje cały twardy dysk. Tak więc aby sformatować drugą partycję (b) na trzecim dysku IDE (`ad2`) należy wydać polecenie `newfs /dev/ad2s1b`. Partycje i plastry można tworzyć za pomocą odpowiednich narzędzi w programie `sysinstall`, albo za pomocą narzędzi dostępnych z linii komend (`fdisk`, `bsdlabel` lub `disklabel`). Szczegółowe informacje dostępne są we FreeBSD Handbook (www.freebsd.org) - podręczniku FreeBSD.

4.10.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Wyjaśnij pojęcie: punkt montowania.
2. Wyjaśnij pojęcie: urządzenie blokowe.
3. Wyjaśnij pojęcie: partycja.
4. Wyjaśnij pojęcie: plaster.
5. Jakie znasz rodzaje systemów plików?
6. Jak oznaczane są poszczególne urządzenia blokowe (dyski, partycje, plastry) w systemie FreeBSD?

4.10.3. Ćwiczenia

Ćwiczenie 1

Odczytaj zawartość dyskietki.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zalogować się jako root,
- 2) utworzyć w katalogu `/mnt` folder, w którym zamontuje urządzenie (może też skorzystać ze znajdującego się w `/mnt` katalogu `floppy`),
- 3) umieścić dyskietkę w napędzie,
- 4) wpisać polecenie montowania urządzeń pamiętając, że należy wybrać odpowiedni typ systemu plików, urządzenie i podać adres bezwzględny katalogu montowania.
- 5) wylistować zawartość dyskietki poleceniem poznany w rozdziale 4.4.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym system operacyjnym i dyskietka z dowolnymi plikami.

Ćwiczenie 2

Odczytaj zawartość płyty CD-ROM.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zalogować się jako root,
- 2) wpisać polecenie montowania pamiętając o parametrach (cd9660, cd0a),
- 3) wpisać polecenie listowania zawartości płyty,
- 4) zanotować nazwy kilku obiektów z płyty,
- 5) odmontować napęd,
- 6) spróbować wylistować zawartość,
- 7) porównać wypisane nazwy z zawartością podaną przez nauczyciela.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym system operacyjnym i płyta CD-ROM w standardzie ISO-9660/Joliet.

Ćwiczenie 3

Odczytaj zawartość napędu PenDrive przyłączonego do portu usb.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zalogować się jako root,
- 2) wpisać polecenie montowania pamiętając, że system widzi PenDrive jako urządzenie da0s1,
- 3) spróbować zamontować PenDrive w innym katalogu,
- 4) podzielić się z grupą uwagami na temat przebiegu próby,
- 5) odmontować napęd,
- 6) odpowiedzieć na pytanie co stanie się, gdy jako typ urządzenia przy kolejnym montowaniu wybierze usb (z kolejnym numerem)?

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym system operacyjnym i PenDrive z dowolną zawartością.

Ćwiczenie 4

Utwórz partycję na dysku wskazanym przez nauczyciela, sformatuj ją i zamontuj w katalogu /mnt/magazyn.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zalogować się jako root,
- 2) uruchomić program `sysinstall`,
- 3) użyć kolejno narzędzi `fdisk` i `Disk Label` aby utworzyć partycję,
- 4) utworzyć na niej system plików poleceniem `newfs`, a następnie zamontować ją poleceniem `mount`,
- 5) sprawdzić widoczność partycji.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym system operacyjnym i czystym dyskiem twardym – nie podzielonym na partycje.

4.10.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) uzyskać informacje o zamontowanych urządzeniach?	<input type="checkbox"/>	<input type="checkbox"/>
2) zamontować napęd?	<input type="checkbox"/>	<input type="checkbox"/>
3) odłączyć napęd?	<input type="checkbox"/>	<input type="checkbox"/>
4) sformatować partycję?	<input type="checkbox"/>	<input type="checkbox"/>

4.11. Bezpieczeństwo danych i kopia zapasowa

4.11.1. Materiał nauczania

Zabezpieczanie systemu plików

Systemy Unix zabezpieczają przed niepowołanym dostępem, autoryzując każdą próbę dostępu do konsoli nazwą użytkownika i hasłem. Powoduje to, że komputer od samego początku wie, jaka osoba z niego korzysta i może jej przypisać szereg ograniczeń i uprawnień. System plików nie jest tu wyjątkiem. Każdemu plikowi i katalogowi przypisane są następujące parametry:

- **właściciel** (owner) – identyfikator użytkownika, który stworzył dany plik,
- **grupa** (group) – grupa, do której należy dany plik (domyślnie grupa logowania właściciela),
- **zestaw uprawnień** (permissions) – czyli lista uprawnień, jakie posiada do danego pliku jego właściciel, grupa i inni użytkownicy.

Właściciela danego pliku można ustawić za pomocą polecenia `chown` (change owner), podając jako pierwszy parametr nowego właściciela, a jako drugi - nazwę pliku/katalogu, którego zmiana się tyczy. Podobnie, przynależność danego pliku do grupy zmienia się poleceniem `chgrp` (change group), ale jako pierwszy parametr podaje się nazwę nowej grupy. Należy pamiętać, że takich zmian może dokonać jedynie root.

Każdemu plikowi jego właściciel przypisuje zestaw uprawnień. Składa się on z trzech części:

- uprawnienia właściciela (u - user),
- uprawnienia grupy (g - group),
- uprawnienia pozostałych (o - others).

Każda z części może zawierać uprawnienia do odczytu (r -read), zapisu (w -write), wykonywania pliku (x - execute) oraz brak uprawnień (minus). W przypadku katalogów, uprawnienie do wykonywania oznacza możliwość wejścia do danego katalogu.

Pełny zestaw uprawnień zapisuje się jako dziewięć znaków (trzy grupy po trzy znaki), z czego pierwsza grupa dotyczy właściciela, druga grupy, a trzecia pozostałych użytkowników:

```
uuugggooo  
rwxrwxrwx
```

Na przykład zestaw `rwx--r-x` oznacza, że właściciel (pierwsze trzy znaki) ma prawo odczytu, zapisu i wykonywania tego pliku, członkowie grupy przypisanej do tego pliku (środkowe trzy znaki) nie posiadają żadnych uprawnień, a pozostali (ostatnie trzy znaki) mogą ten plik odczytywać (r) i wykonywać (x), nie mogą natomiast do niego zapisywać.

Załóżmy, że istnieje plik `ksiegi.dat`, którego właścicielem jest `jnowak`, grupa przypisana do tego pliku to `ksiegowosc` a zestaw uprawnień to `rw-r-----`. Będzie to oznaczało, że użytkownik `jnowak` może odczytywać i zapisywać plik `ksiegi.dat`. Członkowie grupy `ksiegowosc` mogą tylko odczytywać ten plik, a pozostali nie mają do niego żadnych uprawnień.

Aby zmienić zestaw uprawnień do pliku używa się polecenia `chmod` (change mode). Jako pierwszy parametr podaje się operację, jaka ma być wykonana, a jako drugi - nazwę pliku. Aby właścicielowi pliku (u) dodać prawo do wykonywania go (x) należy wydać polecenie:

```
# chmod u+x nazwapliku
```

Podobnie, aby grupie odebrać wszystkie prawa do pliku, należy użyć polecenia:

```
# chmod g-rwx nazwapliku
```

Można też określić konkretne prawa dla właściciela (u) i pozostałych (o) za pomocą znaku „=”:

```
# chmod u=rx,o=r nazwapliku
```

Uprawnienia można również podawać w sposób numeryczny. Przyjmuje się, że prawo do odczytu ma wartość 4, prawo do zapisu: 2, a do uruchamiania: 1. Wpisując polecenie podaje się trzycyfrową liczbę, której pierwsza cyfra to suma wszystkich uprawnień dla właściciela, druga liczba to suma uprawnień dla grupy, a trzecia - dla pozostałych. Na przykład, uprawnienia do pliku `plac.e.rtf` zapisane w postaci numerycznej jako 604 oznaczają prawo do odczytu i zapisu ($4 + 2 = 6$) dla właściciela, i prawo do odczytu (4) dla pozostałych.

Uprawnienia plików w katalogu sprawdza się poleceniem `ls -l`. Wyświetli ono listing wszystkich plików w szczegółowym formacie, z uwzględnieniem uprawnień do pliku.

Archiwizacja

Archiwizacja danych to jedno z najpoważniejszych zagadnień w pracy administratora serwerów. W przypadku awarii systemu, musi on być w stanie odtworzyć wszystkie dokumenty użytkowników, ich pocztę e-mail, strony WWW serwera Internetowego, a czasem i złożone bazy danych. Warto sporządzić sobie plan archiwizacji, który będzie zawierał listę plików/katalogów podlegających archiwizowaniu, interwał wykonywania kopii zapasowych, metodę archiwizacji i plan przechowywania nośników z kopiami zapasowymi. Nośniki te powinny być chronione przed zniszczeniem, ogniem, wilgocią i kradzieżą, a już na pewno nie powinny znajdować się w bezpośrednim sąsiedztwie serwera (jaki pożytek z kopii zapasowej, która ulegnie zalaniu razem z serwerem).

Bardzo dobrym narzędziem do wykonywania kopii zapasowych jest program `tar`. Jego dokumentacja dostępna jest w Internecie i publikacjach dotyczących administrowania serwerami. W przypadku dużej ilości informacji warto rozważyć zakup streamera, czyli napędu przechowującego ogromne ilości danych na stosunkowo tanich taśmach magnetycznych.

Przy planowaniu archiwizacji dobrze jest uwzględnić poniższe lokalizacje:

- /home – katalogi domowe użytkowników,
- /var/mail lub /var/spool/mail – poczta e-mail,
- /var/www – strony WWW serwera Internetowego,
- /etc – pliki konfiguracyjne,
- /var/mysql lub /var/db/mysql – bazy danych mySQL,
- /var/named – dane serwera DNS.

4.11.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie polecenia związane są z systemem zabezpieczeń Unix?
2. Jakie parametry określające bezpieczeństwo przypisane są każdemu plikowi?
3. Na czym polega numeryczna notacja uprawnień do plików i katalogów?
4. Jak tworzy się plan archiwizacji? Na co trzeba zwrócić uwagę?
5. Kto może zmieniać uprawnienia do plików i katalogów?
6. Jakie lokalizacje należy wziąć pod uwagę planując archiwizację danych?

4.11.3. Ćwiczenia

Ćwiczenie 1

Utwórz puste pliki i nadaj im odpowiednie uprawnienia: lodz - odczyt dla właściciela i grupy, warszawa - odczyt i zapis dla właściciela, zapis dla pozostałych, krakow - odczyt dla grupy, zapis dla właściciela, uruchamianie i odczyt dla pozostałych, trojmiasto - wszystkie uprawnienia dla wszystkich.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) utworzyć wszystkie (puste) pliki wymienione w ćwiczeniu,
- 2) sprawdzić, czy zostały utworzone, najlepiej jednocześnie przeglądając nadane domyślne uprawnienia (długi listing),
- 3) przypisać każdemu plikowi uprawnienia zgodnie ze specyfikacją,
- 4) sprawdzić rezultat.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 2

Nadaj plikowi trojmiasto z Ćwiczenia 1 uprawnienia `rxwxrw-r--` za pomocą reprezentacji numerycznych.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) obliczyć numeryczne wartości uprawnień dla każdej z grup,
- 2) nadać uprawnienia za pomocą odpowiedniego polecenia,
- 3) sprawdzić rezultat.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i programem tar.

Ćwiczenie 3

Za pomocą polecenia tar stwórz archiwum (w /tmp) swojego katalogu domowego.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) użyć polecenia tar z parametrami `-czf` do stworzenia archiwum (patrz poprzednie rozdziały),
- 2) pamiętać aby wcześniej wejść do katalogu /tmp, lub podać pełną ścieżkę do tworzonego archiwum,
- 3) utworzyć katalog tymczasowy i rozpakować w nim archiwum. Sprawdzisz w ten sposób poprawność wykonania ćwiczenia.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i programem tar.

4.11.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) zmienić właściciela pliku?	<input type="checkbox"/>	<input type="checkbox"/>
2) nadać wybrane prawo korzystania z pliku?	<input type="checkbox"/>	<input type="checkbox"/>
3) usunąć określone prawo korzystania z pliku?	<input type="checkbox"/>	<input type="checkbox"/>
4) nadać kilka praw korzystania z pliku?	<input type="checkbox"/>	<input type="checkbox"/>
5) zarchiwizować wybrany katalog?	<input type="checkbox"/>	<input type="checkbox"/>

4.12. Sieć TCP/IP – konfiguracja i usługi

4.12.1. Materiał nauczania

Protokół TCP/IP

Protokół TCP/IP jest podstawowym protokołem komunikacyjnym używanym w obecnych sieciach komputerowych. Również Internet pracuje w oparciu o ten protokół. Określa on, że każdy komputer w sieci ma swój adres (adres IP) składający się z czterech oddzielonych od siebie kropkami liczb (0-255), z czego 0.0.0.0 i 255.255.255.255 są adresami specjalnymi i nie mogą być używane do nazywania komputerów. Takich adresów specjalnych jest jeszcze kilka, na przykład 127.x.x.x oznacza komputer lokalny. Duże sieci komputerowe są podzielone na mniejsze podsieci, z których każda ma swój adres i maskę podsieci określającą, która część adresu IP określa podsieć, a która adres komputera. Dla zapisu dziesiętnego można informacje zebrać w tabeli:

Tabela 2. Definicje i rozmiary klas sieci.

Klasa	x1.x2.x3.x4	x1	x2.x3.x4	ilość sieci
A	sieć.host.host.host	od 1 do 126	0..255	126
B	sieć.sieć.host.host	od 128 do 191	0..255	64□256
C	sieć.sieć.sieć.host	od 192 do 223	0..255	32□256□256

Ponadto istnieje model OSI opracowany przez organizację ISO, który całą komunikację sieciową między komputerami dzieli na 7 warstw i opisuje, jakie zadania na każdej z nich spoczywają.

Tabela 3. Warstwy modelu OSI.

7	Aplikacji.	Warstwa prezentuje usługi, które są realizowane przez aplikacje (przesyłanie plików, poczta elektroniczna itp.)
6	Prezentacji.	Warstwa odpowiada za format wymiany danych pomiędzy komputerami (kodowanie i dekodowanie danych do postaci wspólnej i zrozumiałej przez niższe warstwy)
5	Sesji.	Warstwa pozwala aplikacjom z różnych komputerów realizować sesję (nawiązywać, wykorzystywać i kończyć połączenie).
4	Transportowa.	Warstwa odpowiada za poprawne przesyłanie danych (dzielenie i łączenie pakietów, potwierdzenie odbioru, ewentualnie żądanie powtórzenia transmisji).
3	Sieciowa.	Warstwa decyduje o przesyłaniu informacji określoną drogą (routing) z pominięciem poprawności.
2	Łączy danych.	Warstwa nadzoruje pakiety logiczne (ramki) danych stosownie do typu sieci (ramki zawierają fizyczne adresy nadawcy i odbiorcy).
1	Fizyczna.	Warstwę stanowi układ elektroniczny przesyłania danych przez fizyczny nośnik (kabel sieciowy, fala elektromagnetyczna).

W oparciu o ten model tworzone są protokoły komunikacyjne i analizowane jest działanie sieci komputerowych.

Konfigurowanie i kontrolowanie sieci w FreeBSD

Każde urządzenie wykorzystywane przez komputer do komunikacji sieciowej jest reprezentowane w systemie przez moduł jądra udostępniający interfejs sieciowy. Polecenie `ifconfig` pozwala konfigurować te interfejsy.

Dostępne interfejsy sieciowe (aktywne i nieaktywne) można sprawdzić za pomocą polecenia:

```
# ifconfig -a
```

Konfigurowanie interfejsu umożliwia dodanie dodatkowych parametrów:

```
# ifconfig nazwa_int adres_ip netmask maska_podsieci up
```

Bardzo przydatnym poleceniem jest `netstat`. W zależności od podanych parametrów realizuje kilka funkcji (np. `man netstat`). Umożliwia na przykład w określonych odstępach czasu podawanie informacji dotyczących ruchu pakietów lub wyświetlanie statystyki wybranego protokołu. Tablicę routingu można zobaczyć wydając polecenie:

```
# netstat -r
```

Istnieją programy narzędziowe, które wyświetlają i umożliwiają zmianę tablicy routingu pakietów w sieci. Na przykład dodanie bramy domyślnej umożliwia polecenie:

```
# route add default gw adres_bramy
```

Ważnym narzędziem diagnostycznym sieci jest polecenie `ping`. Przeprowadza ono test sprawdzający, czy określony host i połączenie z nim działa prawidłowo. Umożliwia też sprawdzenie, jaka część przesyłanych pakietów jest gubiona oraz jaki jest czas ich przesłania. Niestety, można wyłączyć odpowiadanie na zapytanie `ping`. Polecenie może wyglądać następująco:

```
ping -c 5 host
```

Istnieje możliwość sprawdzenia, jaką drogą podąża pakiet do określonego hosta. Program `traceroute` wysyła Datagramowy Protokół Użytkownika (UDP) z niskim czasem życia (TTL) (kolejno od 1), aż osiągnie on zadanego hosta. Jednocześnie rejestruje adresy routerów, na których pakiet wygasa. Dzięki temu możliwe jest ustalenie ścieżki, jaką pakiet podąża do danego hosta:

```
traceroute host
```

Pliki konfiguracyjne sieci znajdują się w następujących lokalizacjach:

- `/etc/hostname`
- `/etc/rc.conf`
- `/etc/resolv.conf`
- `/etc/hosts`

Usługi sieciowe

Systemy Unixowe bardzo często są używane jako serwery sieciowe. Dlatego istnieje dla nich cała gama oprogramowania sieciowego. Są to przede wszystkim demony, czyli programy lub procesy działające niezauważalnie w tle i wykonujące czynności usługowe na rzecz systemu i klientów. Nasłuchują one na poszczególnych portach (TCP/IP) na połączenia

od innych komputerów, a po nawiązaniu komunikacji serwują im treści lub dane zgodne ze swoim przeznaczeniem. Najbardziej znane programy to:

- Apache – serwer WWW dostępny razem z systemem,
- ProFTPD – serwer FTP,
- BIND – serwer DNS,
- Sendmail, Procmail, Qmail – serwer poczty.

Skrypty startowe powyższych demonów znajdują się w `/etc/rc.d` oraz `/usr/local/etc/rc.d`, a pliki konfiguracyjne w `/etc` i `/usr/local/etc`.

Do uruchamiania mniejszych, rzadziej używanych usług służy demon `inetd`. Jego plik konfiguracyjny znajduje się w `/etc/inetd.conf`. Po każdorazowej modyfikacji tego pliku trzeba powiadomić o tym fakcie program, wysyłając sygnał HUP poleceniem:

```
# killall -HUP inetd
```

Czasami na serwerze trzeba filtrować pakiety danych wchodzące i wychodzące w celu zwiększenia bezpieczeństwa, za pomocą tak zwanego FireWall'a. Służą do tego pakiety takie jak Packet Filter (PF), `ipfw`, `ipchains` i inne.

4.12.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie narzędzia służą do konfigurowania sieci w systemie FreeBSD?
2. Jakie narzędzia służą do kontrolowania sieci?
3. Jakie są warstwy modelu OSI? Jakie aspekty komunikacji opisuje każda z nich?
4. Co to jest demon?

4.12.3. Ćwiczenia

Ćwiczenie 1

Ustaw adres IP swojego komputera na `10.127.0.x`, gdzie `x` to Twój numer w dzienniku. Maskę podsieci `255.255.255.0`, bramę domyślną `10.127.0.254`.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zalogować się jako `root`,
- 2) sprawdzić, który interfejs jest kartą sieciową,
- 3) zmienić ustawienie IP komputera za pomocą polecenia `ifconfig`,
- 4) dodać bramę domyślną za pomocą polecenia `route add`,
- 5) użyć narzędzia `ping` do sprawdzenia połączenia,
- 6) zrestartować komputer aby przywrócić oryginalne ustawienia sieciowe.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 2

Sprawdź opóźnienie w komunikacji między komputerem z którego korzystasz, a pozostałymi komputerami w pracowni. Sprawdź też, jakie są opóźnienia do hostów internetowych: `www.wp.pl` i `www.yahoo.com`.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) sprawdzić i zapisać adresy trzech wybranych komputerów w pracowni,
- 2) użyć narzędzia ping wpisując kolejno zapisane adresy,
- 3) odczytać opóźnienie,
- 4) porównać z odczytami innych uczniów w grupie,
- 5) użyć ponownie narzędzia ping wpisując adresy internetowe,
- 6) porównać rezultaty z opóźnieniami w sieci lokalnej.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

Ćwiczenie 3

Prześledź trasę, jaką wędrują pakiety od komputera z którego korzystasz, do każdej z lokalizacji sprawdzanej w Ćwiczeniu 2.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) użyć narzędzia `traceroute`, aby prześledzić trasę pakietów dla każdego z adresów,
- 2) powinieneś zauważyć, że trasa w sieci lokalnej jest znacznie krótsza niż trasa do komputerów w sieci ogólnosiwiatowej

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD.

4.12.4. Sprawdzian postępów

Czy potrafisz:

- 1) odczytać tablicę routingu?
- 2) sprawdzić komunikację i opóźnienie sieci?
- 3) prześledzić trasę pakietów do wybranego hosta?

Tak **Nie**

- | | |
|--------------------------|--------------------------|
| <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/> |

4.13. Powłoka graficzna systemu

4.13.1. Materiał nauczania

Wiele osób utożsamia systemy Unixowe z linią poleceń i konsolą tekstową, trudnymi poleceniami i złożonymi instrukcjami. Oczywiście istnieje też graficzny interfejs użytkownika - X Windows, pracujący praktycznie na każdej odmianie Unixa.

Najtrudniejsze zadanie jak zwykle spoczywa na administratorze, który musi skonfigurować Iksy do współpracy ze sprzętem i programami. Na FreeBSD większość potrzebnych plików znajduje się w katalogu /usr/X11R6. Podkatalog bin zawiera zestaw poleceń dla użytkowników, a sbin – dla administratora. W etc zdefiniowana jest konfiguracja.

Aby utworzyć działającą konfigurację serwera X, trzeba jako root wydać polecenie `xorgcfg -textmode`. Ten program spróbuje wykryć istniejący sprzęt graficzny i zapyta o pozostałe szczegóły niezbędnej konfiguracji, po czym zapisze plik konfiguracyjny w odpowiednim miejscu (/etc/X11/xorg.conf lub /etc/X11/XF86Config w starszej wersji). Serwer X uruchamia się poleceniem `startx`.

Aplikacje dla X Windows działają w architekturze klient-serwer. Oznacza to, że serwer X (główna część systemu) może działać na serwerze, a użytkownicy mogą się do niego podłączać ze stacji roboczych.

Dla serwera X dostępne są tak zwane menadżery okien i programy do zarządzania pulpitem. Najpopularniejsze z nich to KDE i Gnome, choć jest ich znacznie więcej. Ułatwiają one i uprzyjemniają pracę ze środowiskiem graficznym dając dostęp do menadżera zadań, ładnego interfejsu użytkownika i zestawu gotowych do użycia aplikacji. Warto wybrać odpowiednią opcję i zainstalować KDE lub Gnome podczas instalacji systemu.

4.13.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Dlaczego instaluje się systemy graficzne?
2. Wymień menadżery pulpitu dla X Windows?
3. Jakie programy umożliwiają skonfigurowanie serwera X?

4.13.3. Ćwiczenia

Ćwiczenie 1

Skonfiguruj serwer X za pomocą jednego z narzędzi konfiguracyjnych.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zalogować się jako root,
- 2) użyć narzędzia `xorgcfg` lub podobnego do utworzenia pliku konfiguracyjnego,
- 3) odpowiadać na pytania kreatora, jeśli nie ma pewności co do konfiguracji sprzętowej komputera, może użyć programu `dmesg` (kreatora może w każdej chwili zamknąć poleceniem CTRL+C),
- 4) po zakończeniu i zapisaniu pliku konfiguracyjnego sprawdzić jego zawartość za pomocą dowolnego edytora tekstu i spróbować uruchomić serwer X za pomocą polecenia `startx`.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i Xorg

Ćwiczenie 2

Uruchom system X Windows z domyślnym menadżerem pulpitu. Uruchom kilka aplikacji.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) uruchomić tryb graficzny,
- 2) odszukać menu,
- 3) uruchomić kilka aplikacji,
- 4) zanotować nazwy najciekawszych aplikacji i ich krótki opis,
- 5) podzielić się rezultatami z innymi uczniami w grupie.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD, Xorg i KDE.

Ćwiczenie 3

Korzystając z narzędzi powłoki graficznej obejrzyj zawartość różnych nośników danych.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) odszukać menadżera plików,
- 2) sprawdzić, jakie urządzenia zostały zamontowane przez system,
- 3) obejrzeć zawartość nośników,
- 4) dla każdego z nośników odczytać uprawnienia do korzystania z zawartych na nim katalogów i plików,
- 5) uporządkować obiekty nazwami, a następnie według innego kryterium,
- 6) odczytać datę ostatniej modyfikacji,
- 7) odmontować i zamontować każde z urządzeń w nowym katalogu,
- 8) sprawdzić możliwość przeglądania zawartości.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD, Xorg i KDE, dyskietka, CD/DVD, pamięć flash z danymi.

Ćwiczenie 4

Korzystając z narzędzi środowiska graficznego przekopiuj dane pomiędzy różnymi nośnikami.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) utworzyć nowy katalog w katalogu domowym oraz na wybranym nośniku zapisywalnym,
- 2) przekopiować wskazane pliki z nośnika zewnętrznego do nowego katalogu w katalogu domowym,
- 3) utworzyć krótki plik tekstowy w katalogu domowym i przekopiować go na nośnik zewnętrzny,
- 4) umożliwić edytowanie pliku tekstowego przez wszystkich użytkowników.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD, Xorg i KDE, dyskietka, CD/DVD, pamięć flash z danymi.

Ćwiczenie 5

Korzystając z narzędzi środowiska graficznego dokonaj modyfikacji ustawień zgodnie z własnymi preferencjami.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) odszukać narzędzie konfiguracji,
- 2) zmienić parametry reakcji myszy,
- 3) zmienić tapetę pulpitu,
- 4) zmienić położenie paska zadań,
- 5) zmienić czcionkę, za pomocą której wyświetlane są informacje sytemu,
- 6) sprawdzić wskazania zegara systemowego.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD, Xorg i KDE.

4.13.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) utworzyć działającą konfigurację serwera X?	<input type="checkbox"/>	<input type="checkbox"/>
2) uruchomić serwer X?	<input type="checkbox"/>	<input type="checkbox"/>
3) przystosować środowisko graficzne do własnych preferencji?	<input type="checkbox"/>	<input type="checkbox"/>
4) montować urządzenia za pomocą narzędzi środowiska graficznego?	<input type="checkbox"/>	<input type="checkbox"/>
5) ustawić uprawnienia do korzystania z katalogów i plików?	<input type="checkbox"/>	<input type="checkbox"/>

4.14. Programy biurowe i użytkowe

4.14.1. Materiał nauczania

Darmowe systemy Unix-like coraz częściej są wykorzystywane w instytucjach państwowych, szkołach i firmach. Dzieje się tak dlatego, że dostępna jest dla nich bogata gama bezpłatnych programów użytkowych, biurowych, graficznych i finansowych, również w języku polskim. Ich lista jest bardzo długa. Poniżej przedstawiono reprezentantów kilku kategorii.

- **Scribus** - dobry program z gatunku DTP. Pozwala na projektowanie plakatów, ogłoszeń itp. Można wykorzystać przy tworzeniu gazetki szkolnej albo redagowaniu ogłoszenia o sprzedaży podręczników.
- **KWord** – edytor tekstu z darmowego pakietu KOffice. Posiada wiele zaawansowanych funkcji i możliwość importu/exportu z formatu Microsoft Word.
- **KSpread** – arkusz kalkulacyjny z darmowego pakietu KOffice z funkcją importu/exportu do formatu Microsoft Excel.
- **Kontakt** – program do zarządzania informacjami osobistymi (PIM). Integruje się z książką adresową, kalendarzem i klientem poczty.
- **KPresenter** – narzędzie do tworzenia prezentacji multimedialnych (pakiet Koffice). Potrafi pracować z prezentacjami programu Microsoft PowerPoint.
- **OpenOffice.org** – darmowy pakiet biurowy posiadający edytor tekstu, arkusz kalkulacyjny, program do tworzenia prezentacji multimedialnych, narzędzie do tworzenia baz danych i aplikacje graficzne. Jest zgodny z Microsoft Office. Cechuje go wysoka stabilność i przejrzystość. Prawdziwy komfort pracy.
- **Gimp** – znakomity i bardzo rozbudowany edytor grafiki operujący warstwami o możliwościach podobnych do Adobe Photoshop. W Internecie dostępnych jest bardzo wiele filtrów (do wykorzystania przez grafików i fotografów) pozwalających wykonać każdą operację na dowolnym pliku graficznym.
- **Konqueror** – przeglądarka internetowa obsługująca większość współczesnych technologii.
- **Firefox** – znakomita, elastyczna przeglądarka internetowa firmy Mozilla.
- **FireBird** – jeden z najlepszych programów pocztowych.
- **Evolution** – program typu PIM ze zintegrowanym klientem poczty e-mail.

4.14.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Wymień jeden z edytorów tekstu WYSIWYG dla platformy Unix.
2. Wymień jeden z edytorów grafiki dla platformy Unix
3. Wymień jeden z programów biurowych dla platformy Unix.
4. Wymień jeden z programów pocztowych dla platformy Unix.
5. Jakie arkusze kalkulacyjne można stosować na platformie Unix?

4.14.3. Ćwiczenia

Najlepszym ćwiczeniem służącym zapoznaniu się z nowymi aplikacjami jest ich regularne używanie. Spróbuj uruchomić każdą z wymienionych aplikacji, która jest dostępna na Twoim komputerze. Kiedy już to zrobisz, spróbuj wykonać poniższe ćwiczenia.

Ćwiczenie 1

Za pomocą jednego z programów DTP lub edytora tekstów, zaprojektuj ulotkę reklamującą zajęcia koła komputerowego w Twojej szkole.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zaprojektować ulotkę,
- 2) zgromadzić lub wykonać elementy graficzne za pomocą dostępnego w komputerze oprogramowania,
- 3) wybrać program DTP,
- 4) wpisać tekst i dodać elementy graficzne zgodnie z projektem,
- 5) zapisać rezultat,
- 6) zaprezentować rezultat pozostałym uczniom w grupie.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD, dostęp do Internetu.

Ćwiczenie 2

Za pomocą programu graficznego dostępnego na Twoim komputerze (np. Gimp) zaprojektuj plakat reklamujący platformę Unix w zastosowaniach szkolnych.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zaprojektować plakat,
- 2) zgromadzić elementy graficzne, których nie będzie w stanie wykonać samodzielnie,
- 3) umieścić elementy na plakacie zgodnie z projektem korzystając z warstw,
- 4) zapisać rezultat,
- 5) zaprezentować rezultat pozostałym uczniom w grupie.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD, dostęp do Internetu.

Ćwiczenie 3

Wstaw do dokumentu tekstowego wykres sporządzony za pomocą arkusza kalkulacyjnego. Opisz krótko wykres.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wybrać temat, który zostanie przedstawiony za pomocą wykresu (wyniki nauczania, wyniki zawodów sportowych, bezrobocie w regionie itp),
- 2) uruchomić arkusz kalkulacyjny i wprowadzić dane,
- 3) wykonać wykres,
- 4) nadać wykresowi atrakcyjną i czytelną postać,
- 5) uruchomić edytor tekstu,
- 6) wpisać tytuł dokumentu,
- 7) wkleić wykres z arkusza,
- 8) opisać wykres,
- 9) sformatować tekst,
- 10) zapisać pracę,
- 11) zaprezentować pracę pozostałym uczniom w grupie,
- 12) uczeń może skorzystać z pakietu KOffice lub OpenOffice, to drugie rozwiązanie jest bardziej wskazane ze względu na dostępność pakietu na platformie Unix, MacIntosh i Windows.

- Wyposażenie stanowiska pracy:
- komputer z zainstalowanym systemem FreeBSD i pakietem biurowym, dostęp do Internetu.

Ćwiczenie 4

Znajdź w sieci Internet nazwy innych programów dla systemu Unix-like z kategorii: edycja grafiki, obsługa multimediiów, programowanie.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) uruchomić wyszukiwarkę sieciową,
- 2) wpisać nazwę kategorii,
- 3) wypisać nazwy znalezionych programów,
- 4) zanotować krótką informację o każdym,
- 5) podzielić się wynikami z innymi uczniami w grupie.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i dostępem do sieci.

Ćwiczenie 5

Znajdź w sieci Internet obrazy przedstawiające najciekawsze miejsca, jakie odwiedziłeś i zbuduj korzystając z nich krótka prezentację multimedialną.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) uruchomić wyszukiwarkę sieciową,
- 2) wpisać frazę, która umożliwi wyszukanie grafik,
- 3) zapisać znalezione obrazy w folderze domowym,
- 4) wybrać i uruchomić program pozwalający przeglądać obrazy i wybrać kilka znalezionych obrazów,
- 5) przyciąć obrazy do jednakowego rozmiaru (na przykład 400x300) i zapisać w podkatalogu,
- 6) utworzyć program umożliwiający budowanie prezentacji multimedialnych i utworzyć kilka slajdów wstawiając wybrane obrazy,
- 7) pokazać prezentację innym uczniom w grupie.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i dostępem do sieci.

4.14.4. Sprawdzian postępów

Czy potrafisz:

	Tak	Nie
1) napisać cv?	<input type="checkbox"/>	<input type="checkbox"/>
2) utworzyć wykres przedstawiający Twoje oceny za ostatni semestr?	<input type="checkbox"/>	<input type="checkbox"/>
3) przyciąć dowolny obrazek pobrany z sieci Internet?	<input type="checkbox"/>	<input type="checkbox"/>
4) wysłać list z załącznikiem do znajomych?	<input type="checkbox"/>	<input type="checkbox"/>
5) obliczyć $\sqrt[3]{10}$?	<input type="checkbox"/>	<input type="checkbox"/>
6) zaprojektować ulotkę o szkole?	<input type="checkbox"/>	<input type="checkbox"/>
7) utworzyć prezentację multimedialną o swoich zainteresowaniach?	<input type="checkbox"/>	<input type="checkbox"/>
8) uruchomić program edukacyjny?	<input type="checkbox"/>	<input type="checkbox"/>

4.15. Drukowanie

4.15.1. Materiał nauczania

Systemy Unix korzystają z kilku menadżerów wydruku. Najbardziej popularne to LPR i CUPS (Common Unix Printing System). Ten drugi jest bardziej uniwersalny i nowoczesny. Dzięki CUPS komputer pracuje jako serwer wydruku. Może przyjmować zadania do drukowania od komputerów klientów, przetwarzać je (kolejkowanie, ustalanie formatu) i wysyłać do drukarki.

Konfiguracja drukarki

Drukarki konfiguruje się za pomocą przeglądarki internetowej. Należy wpisać w niej adres `http://localhost:631/admin`, aby połączyć się z usługą administracyjną CUPS, a następnie:

- wybrać: Printers -> Add printer,
- wpisać nazwę drukarki pod jaką ma być widoczna w systemie (np. hp656c) oraz opcjonalnie jej opis (Description)
- wybrać urządzenie, do którego podpięta jest drukarka (dla HP powinno być "AppSocket/JetDirect" a dla innych "USB Printer # 1")
- wybrać URI (Uniform Resource Identifier) dla drukarki (najlepiej "lpd://hostname/queue"; trzeba wpisać ręcznie),
- wybrać rodzaj sterownika dla drukarki (jeśli jest kilka). Drukarki HP powinny mieć "HP" a reszta RAW albo GENERIC.
- wybrać sterownik drukarki (jeśli jest kilka).

Drukarka powinna być zainstalowana. Aby ją przetestować należy wrócić do strony głównej panelu CUPS i w górnym menu wybrać Printers, a następnie przy dodanej właśnie drukarce - „Print test page“ (drukuj stronę testową).

W KDE można skonfigurować i przetestować drukarkę w bardziej przyjazny dla użytkownika sposób, używając panelu drukowania w KDE Control Center (Panel sterowania KDE).

Drukowanie

Drukowanie w aplikacjach pod X Windows odbywa się za pomocą poleceń menu. W linii komend do wydrukowania pliku można użyć polecenia `lpr`. Na przykład, aby wydrukować zawartość pliku `/etc/fstab` na drukarce o nazwie „hp“ należy wpisać:

```
lpr -P hp /etc/fstab
```

4.15.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie systemy wydruku są dostępne w systemach Unixowych?
2. Opisz proces instalowania drukarki w systemie CUPS.
3. Co należałoby zmienić w powyższym opisie, gdyby drukarka podłączona była portem równoległym (LPT), lub byłaby udostępniona przez komputer w sieci?

4.15.3. Ćwiczenia

Ćwiczenie 1

Zainstaluj drukarkę sieciową udostępnioną w pracowni komputerowej na swoim komputerze.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś

- 1) uruchomić przeglądarkę WWW i wpisać adres do panelu administracyjnego CUPS,
- 2) dodać drukarkę,
- 3) prawidłowe ustawienia poda nauczyciel,
- 4) sprawdzić drukowanie.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i CUPS. Sieć komputerowa i drukarka udostępniona w tej sieci.

Ćwiczenie 2

Wydrukuj plik utworzony w aplikacji biurowej (poprzedni rozdział) na zainstalowanej w Ćwiczeniu 1 drukarce sieciowej.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) otworzyć odpowiedni dokument i z menu Plik (File) wybrać polecenie Drukuj (Print).

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD, Xorg i CUPS. Sieć komputerowa i drukarka udostępniona w tej sieci.

4.15.4. Sprawdzian postępów

Czy potrafisz:

- | | Tak | Nie |
|----------------------------|--------------------------|--------------------------|
| 1) skonfigurować drukarkę? | <input type="checkbox"/> | <input type="checkbox"/> |
| 2) wydrukować dokument? | <input type="checkbox"/> | <input type="checkbox"/> |

4.16. Instalacja systemu Unix

4.16.1. Materiał nauczania

W zależności od używanego systemu, proces instalowania wygląda różnie. Czasami instaluje się cały system z płyty kompaktowej, czasami używa się dyskietki startowej, a resztę instalator pobiera z Internetu. Czasami cały proces wykonuje się przez sieć. Aby prześledzić proces instalacji krok po kroku, najlepiej skorzystać z dokumentacji dostarczonej z systemem operacyjnym lub z Internetu.

Etapy instalowania

- **Przydzielenie miejsca na dysku** – ten etap polega na podzieleniu dysku na partycje, utworzeniu systemów plików i ustaleniu punktów montowania. Instalator poprosi też o utworzenie partycji wymiany i przygotowuje ją odpowiednio. W przypadku serwerów i ważnych komputerów dobrze jest podzielić dysk w taki sposób, żeby przynajmniej katalogi /home, /boot, /usr, /var i /tmp znajdowały się na osobnych partycjach.
- **Wybranie składników instalacji** – tutaj wybiera się składniki systemu, które zostaną zainstalowane. Należy dobrze przemyśleć wybór. Pierwsza zasada bezpieczeństwa zaleca, żeby wyłączać wszystko co jest niepotrzebne. Tutaj też trzeba zdecydować, czy potrzebny jest dostęp do systemu X Windows i menadżerów pulpitu takich jak KDE lub Gnome.
- **Wybranie medium instalacji** – trzeba wskazać instalatorowi, gdzie znajdują się pakiety do zainstalowania. Zwykle do wybiera się między lokalnymi dyskami, napędem CD/DVD i Internetem (http lub ftp).
- **Kopiowanie plików** – ten etap odbywa się automatycznie. Operator jest jedynie proszony o wkładanie kolejnych płyt (jeśli wybrano instalację z CD).
- **Ustawianie systemu** – w trakcie tego etapu instalator będzie zadawał najwięcej pytań. Pozwoli skonfigurować kartę sieciową, połączenie z Internetem, najważniejszy sprzęt, wybrać język i ustawić hasło roota.
- **Uruchomienie systemu** – zawsze po zakończeniu instalacji konieczne jest ponowne uruchomienie komputera. Spowoduje to uruchomienie właściwego jądra systemu z dysku i załadowanie nowo zainstalowanego systemu.

Dodawanie i usuwanie składników

Prawie każdy system Unix wyposażony jest w mechanizm pakietów, który pozwala na łatwe instalowanie i usuwanie składników systemu. FreeBSD także ma taki system. Są to programy `pkg_add`, `pkg_delete` i `pkg_info`, które służą do dodawania, usuwania i pobierania informacji o zainstalowanych pakietach. Pakiety mają rozszerzenie `.tgz` i są specjalnie spreparowanymi archiwami `tar` skompresowanymi algorytmem `gzip`. Można również posłużyć się programem `sysinstall` aby po instalacji zmienić wybrane komponenty i ustawienia systemu (np. domyślny menadżer pulpitu w X Windows).

System portów

Systemy BSD dysponują specjalnym narzędziem do instalacji oprogramowania – portami. Są one o tyle lepsze od pakietów poznanych wcześniej, że porty kompilowane są ze źródeł na komputerze na którym są instalowane. Daje to większą wydajność i dopasowanie konkretnego oprogramowania do potrzeb i warunków instalacji.

Drzewo portów znajduje się w katalogu `/usr/ports`. Znajdują się tam katalogi z tematycznie poszeregowanymi portami. Typowa procedura instalacji portu polega na:

1. Wejściu do katalogu z portem,
Wykonaniu polecenia `make install` jako `root`,
Jeśli instalacja powiedzie się, wykonaniu polecenia `make clean`, aby wyczyścić katalog ze zbędnych plików.

Instalacja ze źródeł

Częstą praktyką u doświadczonych administratorów jest kompilowanie i instalowanie oprogramowania z kodu źródłowego. Procedura jest tylko trochę bardziej skomplikowana niż w przypadku portów. Należy:

1. Ściągnąć archiwum z kodem źródłowym programu i rozpakować je.
2. Wejść do katalogu źródeł (zwykle nazywa się podobnie jak archiwum).
3. Wykonać polecenie: `./configure` (czasami z dodatkowymi parametrami).
4. Wykonać polecenie `make install clean`.

4.16.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie są etapy instalowania systemu Unix? Na czym one polegają?
2. Czym należy się kierować wybierając pakiety do instalacji?
3. Jakie istnieją metody instalowania dodatkowego oprogramowania?
4. Jakie są zalety i wady każdej z metod opisanej powyżej?

4.16.3. Ćwiczenia

Ćwiczenie 1

Zainstaluj system FreeBSD na swoim komputerze.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) postępować dokładnie według instrukcji nauczyciela, nieprawidłowa instalacja systemu może spowodować utratę danych na wszystkich dyskach komputera.

Wyposażenie stanowiska pracy:

- komputer z czystym dyskiem twardym, płyty instalacyjne systemu FreeBSD.

Ćwiczenie 2

Za pomocą systemu portów zainstaluj program `links` (kategoria `www`)

Sposób wykonania ćwiczenia Aby wykonać ćwiczenie powinieneś

- 1) przejść do katalogu zawierającego program `links` (`/usr/ports/www/links`),
- 2) uruchomić proces instalacyjny korzystając z narzędzia `make`,
- 3) zakończyć instalację (`make clean`).

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i drzewem portów w katalogu `/usr/ports`.

4.16.4. Sprawdzian postępów

Czy potrafisz:

- 1) zainstalować system FreeBSD?
- 2) wyświetlić informacje o zainstalowanych pakietach?
- 3) dodać składnik do systemu?

Tak **Nie**

<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

4.17. Uruchamianie programów napisanych dla systemu MS Windows

4.17.1. Materiał nauczania

Czasami, używając systemu Unix zachodzi potrzeba uruchomienia konkretnej aplikacji znanej z systemu Microsoft Windows. Istnieje program o nazwie Wine, który jest implementacją Windowsowego API, działającą w X Windows. Oznacza to, że dzięki niemu można pod FreeBSD, Linuxem, Solarisem czy innym Unixem uruchomić prawie każdy program napisany dla Windows. Wine jest do pobrania ze strony www.winehq.com. Jest też w pakietach instalacyjnych FreeBSD, oraz w drzewie portów. Instaluje się go tak samo jak każdą inną aplikację. Wine wymaga dużo konfigurowania ze strony użytkownika, ale gdy już uda się stworzyć działający plik konfiguracyjny, będzie działało naprawdę dobrze. Aby uruchomić jakiś program w Wine, należy skorzystać z polecenia X Windows (terminal):

```
wine nazwa_programu_windowskowego.exe
```

Na stronie producenta jest też dostępna baza danych aplikacji działających w Wine.

Windows i Unix w jednej sieci

Unix wyśmienie radzi sobie w sieci zdominowanej przez komputery z Windowsem. Potrafi obsłużyć protokół SMB, znany szerzej jako udostępnianie plików i drukarek Windows, dzięki pakietowi Samba. Instalacja i konfiguracja jest bardzo łatwa dzięki panelowi administracyjnemu Swat, dostępnemu z przeglądarki WWW. Jako ciekawostkę można podać, że Samba znakomicie spisuje się jako kontroler domeny Windows i często zastępuje przeznaczony do tego Windows Server.

4.17.2. Pytania sprawdzające

Odpowiadając na pytania sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie aplikacje można uruchomić za pomocą wine, a z jakimi są problemy?
2. Do czego służy pakiet Samba? Jaką rolę odgrywa w nim Swat?
3. W jaki sposób skonfigurować udostępnianie katalogu w Sambie?

4.17.3. Ćwiczenia

Ćwiczenie 1

Uruchom program Notatnik za pomocą Wine

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) skopiować plik notepad.exe do swojego katalogu domowego,
- 2) uruchomić terminal, wejść do swojego katalogu domowego i uruchomić program wine podając jako argument pełną nazwę pliku notatnika,
- 3) sprawdzić działanie.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD i Wine. Dostęp do pliku notepad.exe z systemu MS Windows.

Ćwiczenie 2

Udostępnić katalog /tmp dla klientów używających systemu MS Windows.

Aby wykonać ćwiczenie powinieneś:

- 1) otworzyć przeglądarkę i wpisać adres panelu administracyjnego Swat (<http://localhost:901>),
- 2) przejść do zakładki „Share settings“ i dodać udostępniony katalog,
- 3) jako ścieżkę (path) wpisać /tmp, a jako nazwę udziału (share name): „public“ i kliknąć na „Create share“,
- 4) jeśli to możliwe, sprawdzić udostępnienie.

Wyposażenie stanowiska pracy:

- komputer z zainstalowanym systemem FreeBSD, samba i swat, sieć komputerowa.

4.17.4. Sprawdzian postępów

Czy potrafisz:

Tak **Nie**

- | | | |
|--|--------------------------|--------------------------|
| 1) uruchomić wybrany program systemu MS Windows? | <input type="checkbox"/> | <input type="checkbox"/> |
| 2) udostępnić katalog systemu FreeBSD użytkownikom MS Windows? | <input type="checkbox"/> | <input type="checkbox"/> |

5. SPRAWDZIAN OSIĄGNIĘĆ

INSTRUKCJA DLA UCZNIĄ

1. Przeczytaj uważnie instrukcję.
2. Podpisz imieniem i nazwiskiem kartę odpowiedzi.
3. Zapoznaj się z zestawem pytań testowych.
4. Test zawiera 15 pytań. Do każdego pytania dołączone są cztery możliwości odpowiedzi. Tylko jedna odpowiedź jest prawidłowa.
5. Udzielaj odpowiedzi wyłącznie na załączonej karcie odpowiedzi, stawiając w odpowiedniej rubryce znak ×. W przypadku pomyłki należy błędną odpowiedź zaznaczyć kółkiem, a następnie ponownie zakreślić odpowiedź prawidłową.
6. Pracuj samodzielnie, bo tylko wtedy będziesz miał satysfakcję z wykonanego zadania..
7. Jeżeli udzielenie odpowiedzi będzie Ci sprawiało trudność, wtedy najlepiej odłóż jego rozwiązanie na później. Wróć do niego, gdy zostanie Ci wolny czas.
8. Na rozwiązanie testu masz 40 minut.

ZESTAW PYTAŃ TESTOWYCH

1. Który zbiór cech systemu Unix charakteryzuje go najlepiej?
 - a) wielozadaniowy, wieloprocesorowy
 - b) wieloprocesorowy, wieloprocesowy
 - c) wielozadaniowy, wielodostępny, sieciowy
 - d) wieloprocesorowy, wielodostępny, sieciowy
2. Z czego składa się system Unix?
 - a) jądra systemu
 - b) aplikacji systemowych
 - c) powłok systemowych
 - d) wszystkie odpowiedzi są prawidłowe
3. Którego polecenia użyjesz do zamontowania dyskietki?
 - a) `newfs /dev/fd0 /mnt`
 - b) `newfs /mnt/dyskietka`
 - c) `mount -t cd9660 /dev/fd0 /mnt/dyskietka`
 - d) `mount -t msdos /dev/fd0 /mnt/dyskietka`
4. Z ilu warstw składa się model OSI?
 - a) 5
 - b) 3
 - c) 7
 - d) 10
5. Pierwsza linijka skryptu może wyglądać w następujący sposób:
 - a) `#!/bin/sh`
 - b) `#/bin/sh`
 - c) `echo „Początek skryptu“`
 - d) `bash >> /dev/null`

6. Pakiet Samba służy do:
 - a) edycji tekstu
 - b) komunikacji sieciowej
 - c) komponowania muzyki
 - d) żadne z powyższych

7. Skrypt powłoki może zawierać:
 - a) komentarze
 - b) nazwy programów
 - c) instrukcje warunkowe
 - d) wszystkie powyższe

8. Drukowanie w systemie Unix zapewnia
 - a) system X
 - b) system KDE
 - c) system CUPS
 - d) powłoka

9. Do bezpośredniej zmiany uprawnień do pliku służy polecenie
 - a) chmod
 - b) chown
 - c) chsh
 - d) chrx

10. Plików, które można uruchomić z konsoli, należy szukać w katalogu
 - a) /bin
 - b) /etc
 - c) /lib
 - d) /mnt

11. Wskaż przeglądarkę WWW
 - a) lynx
 - b) mutt
 - c) tar
 - d) wget

12. Polecenie: `ls -F 2> blad 1>&2`
skieruje listing
 - a) na ekran
 - b) do pliku: blad
 - c) na ekran i do pliku: blad
 - d) nigdzie

13. Wybierz polecenie, które utworzy strukturę katalogów /test1/test2/test3
- a) nie można jednym poleceniem
 - b) `mkdir /test1/test2/test3`
 - c) `mkdir -R /test1/test2/test3`
 - d) `mkdir -p /test1/test2/test3`
14. Potrzebujesz obejrzeć drugą, trzecią i czwartą linię pliku. Wpiszesz polecenie
- a) `cat plik | head -4 | tail -2`
 - b) `cat plik | head -4 | tail -3`
 - c) `cat plik | head -2 | tail -4`
 - d) `cat plik | tail -2 | head -4`
15. Zapisz jednym poleceniem: w pliku informację o zawartości katalogu bieżącego, a na ekranie informację o ilości plików w katalogu bieżącym
- a) `ls -l | plik.txt | wc -l`
 - b) `ls -l > plik.txt | wc -l`
 - c) `ls -l |tee plik.txt | wc -l`
 - d) `ls -l | wc -l | tee plik.txt`

KARTA ODPOWIEDZI

Imię i nazwisko.....

Administrowanie systemem operacyjnym Unix

Zakreśl poprawną odpowiedź.

Nr zadania	Odpowiedź				Punkty
1	a	b	c	d	
2	a	b	c	d	
3	a	b	c	d	
4	a	b	c	d	
5	a	b	c	d	
6	a	b	c	d	
7	a	b	c	d	
8	a	b	c	d	
9	a	b	c	d	
10	a	b	c	d	
11	a	b	c	d	
12	a	b	c	d	
13	a	b	c	d	
14	a	b	c	d	
15	a	b	c	d	
Razem:					

6. LITERATURA

1. Carling M., Dennis J., Degeler S., Linux Administracja, Robomatic 2000
2. Frisch A., Unix – Administracja systemu, O'Reilly & Associates, Inc. 1997
3. Hunt C., Serwery sieciowe Linuksa, Mikom, 2000
4. Lucas M., FreeBSD. Podstawy administracji systemem, HELION 2004