**RAID**

Nadmiarowa macierz niezależnych dysków) - polega na współpracy dwóch lub więcej dysków twardych w taki sposób, aby zapewnić dodatkowe możliwości, nieosiągalne przy użyciu jednego dysku. RAID używa się w następujących celach:

* zwiększenie niezawodności (odporność na awarie),
* przyspieszenie transmisji danych,
* powiększenie przestrzeni dostępnej jako jedna całość.

**RAID 0**

Polega na połączeniu ze sobą dwóch lub więcej dysków fizycznych tak, aby były widziane jako jeden dysk logiczny. Powstała w ten sposób przestrzeń ma rozmiar taki jak N\*rozmiar **najmniejszego** z dysków. Dane są przeplecione pomiędzy dyskami.

Trzy dyski po 500 GB zostały połączone w RAID 0. Powstała przestrzeń ma rozmiar 1,5 TB. Szybkość zapisu lub odczytu jest prawie trzykrotnie większa niż na pojedynczym dysku. Oczywiście sumaryczna szybkość jest 3-krotnością szybkości najwolniejszego z dysków,

**RAID 1**

Polega na replikacji pracy dwóch lub więcej dysków fizycznych. Powstała przestrzeń ma rozmiar pojedynczego nośnika. RAID 1 jest zwany również mirroringiem.

Trzy dyski po 250GB zostały połączone w RAID 1. Powstała w ten sposób przestrzeń ma rozmiar 250 GB. Jeden lub dwa dyski w pewnym momencie ulegają uszkodzeniu. Cała macierz nadal działa.

**RAID 2**

Dane na dyskach są paskowane. Zapis następuje po 1 bicie na pasek. Potrzebujemy minimum 8 powierzchni do obsługi danych oraz dodatkowe dyski do przechowywania informacji generowanych za pomocą kodu Hamminga potrzebnych do korekcji błędów. Liczba dysków używanych do przechowywania tych informacji jest proporcjonalna do logarytmu liczby dysków, które są przez nie chronione. Połączone dyski zachowują się jak jeden duży dysk. Dostępna pojemność to suma pojemności dysków przechowujących dane.

**RAID 3**

Dane składowane są na N-1 dyskach. Ostatni dysk służy do przechowywania sum kontrolnych. Działa jak striping (RAID 0), ale w macierzy jest dodatkowy dysk, na którym zapisywane są kody parzystości obliczane przez specjalny procesor, przez co kontrolery potrzebne do przekierowania.

Pięć dysków po 250GB zostało połączonych w RAID 3. Powstała w ten sposób przestrzeń ma rozmiar 1TB (250 GB zarezerwowane na sumy kontrolne). Jeden dysk w pewnym momencie ulega uszkodzeniu. Cała macierz nadal działa. Po włożeniu nowego dysku na miejsce uszkodzonego jego zawartość odtwarza się.

**RAID 4**

RAID 4 jest bardzo zbliżony do RAID 3, z tą różnicą, że dane są dzielone na większe bloki (16, 32, 64 lub 128 kB). Takie pakiety zapisywane są na dyskach podobnie do rozwiązania RAID 0. Dla każdego rzędu zapisywanych danych blok parzystości zapisywany jest na dysku parzystości.

**RAID 5**

Poziom piąty pracuje bardzo podobnie do poziomu czwartego z tą różnicą, iż bity parzystości nie są zapisywane na specjalnie do tego przeznaczonym dysku, lecz są rozpraszane po całej strukturze macierzy. RAID 5 umożliwia odzyskanie danych w razie awarii jednego z dysków przy wykorzystaniu danych i kodów korekcyjnych zapisanych na pozostałych dyskach

Pięć dysków po 250GB zostaje połączonych w RAID 5. Powstała w ten sposób przestrzeń ma rozmiar 1 TB. Jeden dysk w pewnym momencie ulega uszkodzeniu. Cała macierz nadal działa. Po wymianie uszkodzonego dysku na nowy jego zawartość zostaje odtworzona.

**RAID 6**

Rozbudowana macierz typu 5 (często pojawia się zapis RAID 5+1). Zawiera dwie niezależne sumy kontrolne. Kosztowna w implementacji, ale dająca bardzo wysokie bezpieczeństwo.

**RAID 0+1**

Macierz realizowana jako RAID 1, którego elementami są macierze RAID 0. Macierz taka posiada zarówno zalety macierzy RAID 0 - szybkość w operacjach zapisu i odczytu - jak i macierzy RAID 1 - zabezpieczenie danych w przypadku awarii pojedynczego dysku. Pojedyncza awaria dysku powoduje, że całość staje się w praktyce RAID 0. Potrzebne są minimum 4 dyski o tej samej pojemności.

**RAID 1+0**

Nazywana także RAID 10. Macierz realizowana jako RAID 0, którego elementami są macierze RAID 1. W porównaniu do swojego poprzednika (RAID 0+1) realizuje tę samą koncepcję połączenia zalet RAID 0 (szybkość) i RAID 1 (bezpieczeństwo) lecz w odmienny sposób. Tworzony jest duży stripe małych mirrorów, dzięki czemu podczas wymiany uszkodzonego dysku odbudowywany jest tylko fragment całej macierzy.

**Księgowanie**

Cechą odróżniającą system Ext3 od innych systemów plików z księgowaniem jest możliwość księgowania nie tylko meta-danych, ale również danych. Oznacza to, że np. o 12 ile reiserfs zachowa prawidłowe struktury systemowe, ale być może utraci dane z pliku, który był modyfikowany w trakcie awarii, o tyle Ext3 (w trybie journal) zarówno naprawi własne struktury, jak i odzyska nasz plik.

Mechanizm księgowania ma następujące tryby działania (ustalane przy montowaniu systemu plików):

**data=writeback** księgowanie tylko meta-danych, podobnie jak w pozostałych systemach plików; teoretyczne, powinno dawać najwyższą wydajność

**data=journal** pełne księgowanie (meta-dane i dane); zapisuje bloki dwa razy najpierw do journalu, potem na właściwą pozycję; daje pełne bezpieczeństwo, choć (teoretycznie) kosztem niskiej wydajności; duży obszar przeznaczony na journal znacząco zwiększa wydajność

**data=ordered** opcja dodana niedawno; zapisywane są tylko meta-dane, jednak, dzięki specjalnemu postępowaniu, zachowuje się podobnie do pełnego księgowania:

* łączy modyfikacje danych i odpowiadających im meta-danych w logiczne całości, zwane transakcjami
* zapisuje na dysk (na docelowe pozycje) dane
* dopiero po zapisaniu danych na dysk zapisuje meta-dane do journala

W ten sposób zachowana jest spójność całości przy mniejszej redukcji wydajności, niż w przypadku pełnego księgowania.

Księgowanie w Ext3 przebiega w następujący sposób:

* Modyfikowane bloki są umieszczane w buforach
* Bufory są łączone w jednostki logiczne zwane transakcjami ; transakcja działa atomowo - modyfikacje dotyczą wszystkich buforów w obrębie transakcji lub żadnego
* Transakcja przechodzi przez punkty kontrolne
* Zakończona(logicznie) transakcja może zostać zatwierdzona (commit) i zapisana do logu, oznacza to, że modyfikacje należy zapisać na dysku
* Zatwierdzona transakcja jest zapisywana (flush) na dysk; w razie awarii wszystkie dane można odzyskać, bo są już zapisane w logu
* Mechanizm księgowania odnotowuje, że transakcja została zakończona (fizycznie)

**Link sztywny**

jest to wskazanie na ten sam plik w innym miejscu niż nastąpiło pierwsze dowiązanie. Dla systemu plików nie ma więc różnicy skąd nastąpi interwencja na pliku bo i tak modyfikowany jest obiekt wskazywany przez i-węzeł. Ograniczeniem jest fakt iż dowiązanie twarde (sztywne) jest ograniczone do jednego systemu plików. Dopiero usunięcie ostatniego dowiązania twardego powoduje usunięcie pliku z dysku.

**Link symboliczny**

jest to wskazanie na dany plik, które może przekraczać systemy plików. Jednak usunięcie wskazywanego pliku powoduje, iż plik faktycznie zostanie on usunięty. Dowiązanie miękkie nie broni przed tym, można je porównać do skrótu z Windows'a.

- Dowiązania miękkie (symbolic link) **ln –s.** Tworzony jest nowy plik – wskazanie przez nazwę.

- Dowiązania sztywne (link) **ln.** Inkrementowany jest licznik wykorzystania pliku – wskazanie bezpośrednie

**Warunki brzegowe (jak duże pliki, jak duże partycje)**

W Ext2

Maksymalny rozmiar partycji to 4TB, a pojedynczego pliku 2GB.

Maksymalna długość nazwy pliku: 255 znaków.

**Szyfrowanie**

Loop-AES to moduł, implementujący szyfrowanie urządzeń blokowych, na przykład dysku twardego lub CD-ROM.  
Moduł szyfruje urządzenia blokowe podmontowane za pomocą *loop device*, będącego standardowym elementem kernela Linuxa. Do szyfrowania używany jest algorytm AES-128, AES-192 lub AES-256, dla którego kluczem jest skrót SHA obliczony na podstawie podanego przez użytkownika hasła (minimum 20 znaków).  
Szyfrowanie odbywa się w sposób przezroczysty dla systemu plików i jest praktycznie niezauważane w systemie. Najlepszą wydajność można osiągnąć konfigurując szyfrowane *loop device* na fizycznej partycji dysku, na którym następnie można założyć dowolny system plików.  
AES(ang. *Advanced Encryption Standard*) został zatwierdzony przez NIST (ang. *National Institute of Standards and Technology*) jako standard dla amerykańskiej administracji rządowej oraz sektora finansowego. Jego złamanie przy obecnie znanej technologii jest niemożliwe.

**Usługi SSH,**

**SSH (ang. secure shell)** czyli tłumacząc na polski "bezpieczna powłoka" jest standardem protokołów komunikacyjnych wykorzystywanych w sieciach komputerowych TCP/IP, w architekturze klient - serwer. W wąskim tego słowa znaczeniu SSH jest zdecydowanie lepszym następcą słynnego protokoły **telnet**. SSH podobnie jak telnet służą do łączenia się ze zdalnym komputerem. Jednakże SSH zapewnia szyfrowanie oraz umożliwia rozpoznawanie użytkownika na wiele różnych sposobów. W szerszym znaczeniu SSH jest wspólną nazwą dla całej rodziny protokołów. Obejmuje ona nie tylko podstawowe protokoły służące do zadań terminalowych, ale również do:

* przesyłania plików (SCP - Secure Copy Protocol, SFTP - Secure File Transfer Protocol)
* zdalnej kontroli zasobów komputera
* tunelowania i forwardowania

Wspólną cechą tych wszystkich protokołów jest identyczna z SSH technika szyfrowania połączenia, przesyłu danych oraz metoda rozpoznawania użytkownika.

**Na chwilę obecną SSH praktycznie wyparło inne "bezpieczne" protokoły jak Rlogin i RSH.**

Protokół składa się z 3 warstw:

* Transportowej
* Autentykacji (hasła, PKI, etc)
* Połączenia

**Warstwa transportowa**

Warstwa transportowa odpowiedzialna jest za:

* ustalenie szyfrowania,
* kompresji (opcjonalnie) oraz
* integralności danych.

Po przesłaniu za pomocą protokołu 1GB danych lub jeśli sesja trwa ponad godzinę następuje ponowne ustalenie wszystkich parametrów połączenia.

**Warstwa autentykacji**

Głównym zadaniem warstwy autentykacji jest wybór metody uwierzytelnienia użytkownika. Do wyboru jest kilka możliwości:

* **password** - najprostsze i najbardziej popularne uwierzytelnianie użytkownika za pomocą hasła.
* **publickey** - metoda uwierzytelniania za pomocą kluczy publicznych i prywatnych. Najczęściej wykorzystywane są klucze typu DSA lub RSA.   
  SSH wspiera również certyfikaty X.509.
* **GSSAPI** - metoda, w której wykorzystywane są mechanizmy typu Kerberos lub NTLM. Metody te są używane przez komercyjną wersję SSH.
* **keyboard-interactive** - jest to metoda uwierzytelniania, w której mieszczą się wszystkie inne metody autentykacji, które nie zostały tutaj wymienione. Może być to między innymi metoda uwierzytelniania za pomocą jednorazowego hasła pobieranego z tokena.

**typy autoryzacji**

Głównym zadaniem warstwy autentykacji jest wybór metody uwierzytelnienia użytkownika. Do wyboru jest kilka możliwości:

* **password** - najprostsze i najbardziej popularne uwierzytelnianie użytkownika za pomocą hasła.
* **publickey** - metoda uwierzytelniania za pomocą kluczy publicznych i prywatnych. Najczęściej wykorzystywane są klucze typu DSA lub RSA.   
  SSH wspiera również certyfikaty X.509.
* **GSSAPI** - metoda, w której wykorzystywane są mechanizmy typu Kerberos lub NTLM. Metody te są używane przez komercyjną wersję SSH.
* **keyboard-interactive** - jest to metoda uwierzytelniania, w której mieszczą się wszystkie inne metody autentykacji, które nie zostały tutaj wymienione. Może być to między innymi metoda uwierzytelniania za pomocą jednorazowego hasła pobieranego z tokena.

**SSL- jaki klucz**

SSL jest protokołem otwartym, nie należącym do żadnej instytucji. SSL można zalecić jako standardowe zabezpieczenie stosowane na stronach WWW i w serwerach internetowych. Firma Netscape współpracuje z organizacją W3C przy rozwijaniu i standaryzowaniu niezawodnych mechanizmów i protokołów dla Internetu. Protokół SSL umożliwia użytkownikom i serwerom szyfrowanie i ochronę danych przesyłanych w sieci WWW i w Internecie. Serwer i przeglądarka, które przesyłają między sobą informacje, muszą współpracować z protokołem SSL. Dzięki protokołowi SSL użytkownicy i serwery mogą chronić dane przekazywane w Internecie za pomocą trzech usług: \*potwierdzanie tożsamości serwerów za pomocą certyfikatów cyfrowych (utrudnia to podszycie się pod autoryzowanych użytkowników);\* ochrona przesyłanych danych za pomocą szyfrowania (uniemożliwia przechwytywanie informacji);\* zapewnienie integralności danych podczas ich przesyłania (utrudnia wandalizm).

Na stosie protokołów TCP/IP warstwa SSL znajduje się między warstwą aplikacji, zawierającą protokoły HTTP, SMTP, Telnet, FTP, Gopher i NNTP, a warstwą transportową (obejmującą moduł TCP — Transport Control Protocol) i warstwą sieciową, która zawiera moduł IP (Internet Protocol). Dzięki umieszczeniu warstwy SSL pod warstwą aplikacji, a nad warstwami protokołu TCP/IP, może ona skorzystać z istniejących internetowych standardów komunikacyjnych nie stając się równocześnie protokołem aplikacji

Dane przesyłane w Internecie między przeglądarką i serwerem współpracującymi z protokołem SSL są zaszyfrowane. Gwarantuje to, że wysłane informacje dotrą do adresata nie zmodyfikowane i nie odczytane przez nikogo po drodze.

**Szyfrowanie z kluczem publicznym** jest najpewniejszą dostępną metodą szyfrowania danych. Podczas inicjowania transmisji (rozpoczynania przesyłania danych) protokół SSL chroni klucz sesji za pomocą tej metody.

Protokół SSL stosuje technologię potwierdzania tożsamości serwerów i szyfrowania danych opracowaną przez firmę RSA Data Security, Inc., która jest właścicielem algorytmu RSA. Protokół SSL szyfruje przesyłane komunikaty, tak aby zapewnić im wysoki poziom bezpieczeństwa podczas ich przesyłania między serwerem i przeglądarką. Ponadto stosowanie innego klucza sesji dla każdej transmisji uniemożliwia hakerowi użycie danych zdobytych podczas jednej transmisji do ataku na kolejną.

Zestaw protokołów TCP/IP obsługuje dwa rodzaje usług (połączeń): usługa **z potwierdzaniem pośrednim** wykonuje różnorodne operacje (na przykład weryfikowanie pakietów) na każdym etapie przesyłania danych. Natomiast usługa **z potwierdzaniem przez adresata** ignoruje pośrednie etapy (i usługi) między nadawcą i odbiorcą. Protokół SSL zapewnia wysoki poziom bezpieczeństwa korzystając z połączeń z potwierdzaniem przez adresata

**Tunelowanie**

zestawienie połączenia między dwoma odległymi hostami tak, by stworzyć wrażenie, że są połączone bezpośrednio.

inaczej przekierowywanie portów polega na przesyłaniu niezabezpieczonych pakietów protokołów TCP (POP3, SMTP czy HTTP) przez bezpieczny protokół SSH.

Istnieją dwa rodzaje przekierowania portów: lokalne (wychodzące) oraz zdalne (przychodzące).

**Quota- ograniczenie miejsca**

**Soft Limit** - określa maksymalną objętość dysku, jaką użytkownik ma przyznaną. Ale jeśli połączymy to z grace period, to soft limit określa nam granicę, po przekroczeniu której użytkownik zostaje ostrzeżony.  
**Hard Limit** - działa to jedynie przy ustawieniu grace period. Określa to absolutny limit na dysku, którego dany użytkownik nie może przekroczyć.  
**Grace Period** - jest to okres czasu (łaski), po jakim zacznie działać opcja soft link. Wartości tej opcji podaje się poleceniem edquota -t. Jednostkami czasu mogą być: sekundy, minuty, godziny, dni, tygodnie lub miesiące.

**Quota FTP**

Działający na zasadzie klient-serwer protokół umożliwiający przesyłanie plików przez Internet. FTP jest najpopularniejszą metodą kopiowania plików z odległej maszyny do lokalnej lub odwrotnie.

* FTP jest protokołem 8-bitowym, dlatego nie wymaga specjalnego kodowania danych na postać 7-bitową, tak jak ma to miejsce w przypadku poczty elektronicznej (patrz standardy MIME, base64, quoted-printable, uuencode).
* Do komunikacji wykorzystywane są dwa połączenia TCP. Jedno z nich jest połączeniem kontrolnym za pomocą którego przesyłane są np. polecenia do serwera, drugie natomiast służy do transmisji danych m.in. plików. FTP działa w dwóch trybach: aktywnym i pasywnym, w zależności od tego, w jakim jest trybie, używa innych portów do komunikacji.
* Jeżeli FTP pracuje w trybie aktywnym, korzysta z portów: 21 dla poleceń (połączenie to jest zestawiane przez klienta) oraz 20 do przesyłu danych. Połączenie nawiązywane jest wówczas przez serwer. Jeżeli FTP pracuje w trybie pasywnym wykorzystuje port 21 do poleceń i port o numerze > 1024 do transmisji danych, gdzie obydwa połączenia zestawiane są przez klienta. W sieciach ukrytych za firewallem komunikacja z aktywnymi serwerami FTP jest możliwa, tylko pod warunkiem, jeżeli odpowiednie porty na firewallu (routerze) są zwolnione. Możliwe jest zainstalowanie wielu serwerów FTP na jednym i tym samym routerze. Warunkiem jest rozdzielenie portów przez router dla każdego serwera.

**Przebieg sesji FTP**

● Sesja FTP składa się z trzech części. Pierwsza z nich to zalogowanie się na odległy komputer, kiedy to trzeba podać nazwę użytkownika i hasło. Na ich podstawie serwer FTP identyfikuje użytkownika i przydziela mu na czas sesji określone prawa dostępu do swoich zasobów. Posiadanie konta i związanych z nim odpowiednich uprawnień na komputerze docelowym jest warunkiem koniecznym do rozpoczęcia transmisji jakichkolwiek danych. Konto takie jest przydzielane przez administratora systemu.

● Druga część sesji to już "właściwa" praca, podczas której wykonuje się operacje na odległych plikach i katalogach - kopiowanie, przenoszenie, zmiana nazwy, itp. Przed rozpoczęciem transmisji konieczne jest podanie identyfikatora użytkownika oraz hasła zabezpieczających odległy komputer przed niepowołanym dostępem. Na serwerach FTP z dostępem publicznym - tzw. anonimowym FTP - identyfikatorem użytkownika jest zazwyczaj anonymous, a hasłem własny adres poczty elektronicznej.

● Etap ostatni to zamknięcie sesji poleceniem close. Niektóre serwery wysyłają na zakończenie krótką statystykę, w której zawarte są m.in. informacje o czasie trwania sesji oraz ilości przesłanych danych.

**Adres:** Pełny adres internetowy (URL) wygląda nieco inaczej niż w przypadku serwerów WWW, bowiem zamiast prefiksu http:// jest używany ftp://. Pozostając przy poprzednim przykładzie, URL serwera FTP firmy Microsoft będzie miał postać: ftp://ftp.microsoft.com.

**Zarządzanie procesami**

Skrypty startowe. Do podstawowych zadań wykonywanych przez skrypty startowe należą:

* + - Ustawianie nazwy komputera
    - Ustawianie strefy czasowej
    - Kontrola dysków programem **fsck** (tylko w trybie automatycznym)
    - Montowanie dysków systemowych
    - Usuwanie starych plików z katalogu **/tmp**
    - Konfiguracja interfejsów sieciowych
    - Uruchomienie demonów i usług sieciowych

Pliki konfiguracyjne są zwykle miniskryptami powłoki **sh**, które skrypty startowe włączają, aby zdefiniować wartości pewnych zmiennych interpretera poleceń.

Tradycyjny proces **init** definiuje 7 poziomów, z których każdy reprezentuje pewien szczególny zakres usług, które powinien wykonywać system:

* + - Poziom 0 – jest poziomem w którym system jest wyłączony
    - Poziom 1 lub S – reprezentuje tryb jednodostępny
    - Poziomy 2..5 – poziomy wielodostępne
    - Poziom 6 – poziom ponownego uruchamiania (restartu)

Poziom 1 umożliwia dostęp z prawami administratora bez potrzeby logowania Poziom S przed przejściem do trybu jednodostępnego prosi o autoryzację

Proces **init** interpretuje skrypt konfiguracyjny **/etc/inittab** i na podstawie niego przechodzi od poziomu 0 do poziomu docelowego zdefiniowanego w tym skrypcie uruchamiając polecenia które mają zadziałać (lub działać dalej). Katalogi **/etc/rc0.d, …, /etc/rc6.d** zawierają linki do skryptów, które mają zostać wykonane na odpowiednim poziomie przez proces **init**

**Ponowne uruchamianie i zamykanie systemu**

* + 1. Użycie polecenia **shutdown**
    2. Użycie polecenia  **halt** lub **reeboot**
    3. Użycie polecenia **telinit** do zmiany poziomu pracy ( telinit 0)
    4. Użycie polecenia **poweroff** nakazującego wyłączenie zasilania systemu

Użycie polecenia **shutdown** to najbezpieczniejsza, najdelikatniejsza i najbardziej dokładna metoda przechodzenia do zatrzymania lub wznowienia systemu, czy też przejścia do trybu jednodostępnego.

**shutdown –h 09:30 „zamykam system”**

Z **procesami** związane są nie 2 lecz 4 numery:

* Rzeczywisty UID i GID
* Efektywny UID i GID
* Numery rzeczywiste są wykorzystywane do rozliczeń.
* Numery efektywne są wykorzystywane do ustalania praw dostępu.
* Rzeczywiste i efektywne numery są zazwyczaj takie same.

**Sygnały** to żądania przerwań na poziomie procesu. Zdefiniowano około 30 rodzajów sygnałów.

Przykładowe sygnały:

1 HUP – rozłączenie

2 INT – przerwanie

3 QUIT – wyjście

9 KILL – zabicie

**SMTP**

**Potrzebne programy**

MUA (Mail User Agent),

MTA (Mail Transfer Agent) oraz

MDA (Mail Delivery Agent) – te trzy rodzaje oprogramowania są niezbędne do poprawnego działania poczty elektronicznej, ponieważ właśnie w podanej wyżej kolejności wiadomość email wędruje pomiędzy tymi trzema agentami.

Pierwszy z nich to nic innego jak klient poczty elektronicznej, najbardziej znany dla przeciętnego użytkownika. Jego funkcje to przede wszystkim tworzenie, wysyłanie oraz czytanie wiadomości. Pozostałe czynności wykonywane często przez programy MUA to np. pobieranie wiadomości, tworzenie i zarządzanie skrzynkami pocztowymi umieszczonymi na dysku twardym, przekazywanie gotowych do wysłania e-maili do odpowiedniego programu MTA, sortowanie i filtrowanie poczty. Najpopularniejszym chyba programem tego typu w środowisku windowsowym jest Outlook Express.

Gdy już nastąpi utworzenie wiadomości, gotowej do przesłania, trafia ona do kolejnego ogniwa w łańcuchu czyli programu MTA, który z reguły znajduje się na serwerze, udostępniającym skrzynki internetowe. Zadaniem tego programu jest przesłanie, za pomocą protokołu SMTP , wiadomości do kolejnych serwerów. Ostatnim etapem jest dotarcie emaila do serwera odbiorcy. Najczęściej spotykanym tego typu programem jest UNIX-owy serwer Sendmail i Postfix.

Ostatni etap to dostarczenie wiadomości do odpowiedniej skrzynki odbiorczej adresata, czyli konkretnie mówiąc jest to nic innego jak miejsce na dysku twardym, skąd , za pomocą programu MUA można daną wiadomość odczytać czy też pobrać. Za dostarczanie listów odpowiedzialny jest program z grupy MDA, którego przykładem jest UNIX-owy program Procmail, a podobne zadanie może również pełnić sendmail. Ważną cechą jest jednak, że do odbierania, wysyłania poczty wystarczą same programy MUA oraz MTA.

Na początku istnienia poczty charakterystyczne było, że oprogramowanie MUA znajdowało się na komputerach, które były też maszynami odbierającymi wiadomości. Ponieważ z biegiem czasu sytuacja zmieniła się na rzecz komputerów „domowych”, co wymusiło powstanie protokołu dostępu do poczty znajdujących się na odległych komputerów. W ten właśnie sposób powstały protokoły POP (Post Office Protocol) i IMAP (Internet Message Access Protocol)

**POSTFIX**

**Poczta- kolejki,**

Menedżer kolejek w Postfiksie wykonuje większość pracy z przetwarzaniem wiadomości email. Ostatecznym zadaniem komponentów Postfiksa przyjmujących pocztę jest przekazanie wiadomości do menedżera kolejek. Służy do tego demon cleanup, który powiadamia menedżera kolejek o umieszczeniu nowej wiadomości w wejściowej kolejce wiadomości. Po otrzymaniu nowej wiadomości menedżer kolejek używa trivial-rewrite do ustalenia informacji o trasie: metody transportowej, która ma zostać użyta, następnego hosta, gdzie ma zostać doręczona, oraz adresu odbiorcy.

Menedżer kolejek utrzymuje cztery różne kolejki: wejściowa (incoming), aktywna (active), wiadomości odroczonych (deferred) i uszkodzonych (corrupt). Po początkowym procesie oczyszczania kolejka wejściowa jest pierwszym przystankiem dla nowych wiadomości. Jeśli zasoby systemu są dostępne, menedżer przenosi ja do kolejki aktywnej i wywołuje jeden z agentów doręczających. Wiadomości, których nie można doręczyć, są przenoszone do kolejki wiadomości odroczonych.

**Poczta- doręczanie,**

Postfix do ustalania, dla których adresów docelowych może przyjmować wiadomości i w jaki sposób ma sie odbyć doręczenie, wykorzystuje koncepcje klas adresów. Główne klasy adresów to local (lokalny), virtual alias (alias wirtualny), virtual mailbox (wirtualna skrzynka pocztowa) i relay (przekazanie). Pod adresy docelowe niezaliczające sie do żadnej z tych klas wiadomości są doręczane poprzez siec za pomocą klienta SMTP (zakładając, że wiadomość została odebrana przez autoryzowanego klienta). W zależności od klasy adresu menedżer kolejek wywołuje odpowiedniego agenta doręczającego do obsłużenia wiadomości.

**Aliasy**

Jest to poprostu dodatkowa nazwa dla twojej skrzynki pocztowej. Czyli mówiąc krócej, możesz posiadać 2 lub więcej różne adresy e-mail, a mimo to, pocztę odbierzesz z jednego konta. Weźmy taki przykład: Ostatnio coraz bardziej popularne stały się adresy typu: `tomek.stala@domena.pl`, więc chcemy aby poczta wysłana właśnie na taki adres trafiała na konto `tomek` istniejące juz na naszym serwerze. Plik odpowiedzialny za listę aliasów jest zdefiniowany w zmiennych `alias\_maps` i `alias\_database`. Powracając do naszego przykładu, w pliku o nazwie `aliases` (którego nazwę i ścieżkę zdefiniowaliśmy w opisanych zmiennych w pliku `main.cf`), dodajemy nowy alias. Wpisów dokonujemy w następujący sposób - (najpierw wpisujemy nazwę aliasa, następnie piszemy dwukropek, oddzielamy spacją lub tabulatorem i wpisujemy nazwę użytkownika, którego konto znajduje się już w naszym systemie)