helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

Listy dyskusyjne. Zarządzanie



Autor: Alan Schwartz
Tłumaczenie: Sławomir Dzieniszewski
ISBN: 83-7197-962-2
Tytuł oryginału: Managing Mailing Lists
Format: B5, stron: 286

Chcesz stworzyć forum dyskusyjne dla odwiedzających Twoją stronę internetową? A może chcesz założyć listę dystrybucyjną, której subskrybenci będą powiadamiani o ważnych wydarzeniach związanych ze stroną? E-mail to jedna z najbardziej uniwersalnych usług internetowych, a listy dystrybucyjne stanowią znakomite narzędzie w procesie tworzenia społeczności sieciowych. Aby założyć listę potrzebujesz jedynie dostępu do Internetu, odpowiedniego programu do zarządzania nią i oczywiście nieco praktycznych informacji zawartych w tej książce. „Prowadzenie list dystrybucyjnych" to prawdziwa kopalnia przydatnej wiedzy zarówno dla prowadzących listy, jak i dla administratorów systemów.

Jeśli jesteś odpowiedzialny za stworzenie i prowadzenie listy dystrybucyjnej, w tej książce znajdziesz wszystkie potrzebne Ci informacje: od stworzenia wewnętrznego regulaminu listy, do rozwiązywania problemów z powracającymi, nieodebranymi wiadomościami. Nauczysz się tworzyć listy moderowane, nadzorować zapisywanie się nowych użytkowników oraz konfigurować listy dystrybucyjne tak, by ich użytkownicy mogli otrzymywać streszczenia prowadzonych tam dyskusji i korzystać z jej archiwów. Z książki „Prowadzenie list dystrybucyjnych" dowiesz się także, które aspekty obsługi listy leżą w Twojej gestii, a które należą do zadań administratora.

Dobór właściwego oprogramowania, sposób jego instalacji i konfiguracji, to przedstawione w książce tematy, które mogą zainteresować administratorów sieci. Znajdą oni w książce wiele cennych wskazówek ułatwiających współpracę z osobami odpowiedzialnymi za funkcjonowanie list.

Książka omawia cztery najpopularniejsze programy do obsługi list dystrybucyjnych: Majordomo, LISTSERV, ListProc i SmartList. Działają one pod kontrolą systemów opartych na Uniksie, LISTERV działa także na innych platformach, w tym Windows NT/2000/XP.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Podziękowania *7*

Przedmowa *9*

Rozdział 1. Wprowadzenie *15*

Podstawowe wiadomości o poczcie elektronicznej 15
Czym jest lista dystrybucyjna? 22
Korzystanie z programów zarządzających listami dystrybucyjnymi 25

Rozdział 2. Projektowanie listy dystrybucyjnej *29*

Dramatis Personae 29
Decyzje związane z tworzeniem listy dystrybucyjnej 30
Zapisywanie się na listę 33
Listy moderowane 38
Wyciągi 41
Archiwa list 48
Eksplodery, listy równorzędne i grupy dyskusyjne 49
Wielkie listy 51
Wybór programu zarządzającego listą 52

Rozdział 3. Zarządzanie listą za pomocą Listproc *57*

Tworzenie własnej listy 57
Pisanie wiadomości powitalnej 58
Testowanie listy 60
Administrowanie listą 60

Rozdział 4. Zarządzanie listą za pomocą Majordomo 65
Tworzenie własnej listy 65
Konfigurowanie listy 66
Pisanie wiadomości powitalnej 73
Testowanie listy 74
Administrowanie listą 75

Rozdział 5. Zarządzanie listą za pomocą SmartList 79
Tworzenie własnej listy 79
Pisanie wiadomości powitalnej 80
Testowanie listy 81
Administrowanie listą 82

Rozdział 6. Zarządzanie listą za pomocą LISTSERV Lite 85
Tworzenie własnej listy 85
Osobiste hasło i potwierdzanie poleceń 87
Konfigurowanie listy 87
Pisanie wiadomości powitalnej 93
Testowanie listy 95
Administrowanie listą 95

Rozdział 7. Zarządzanie listą za pomocą sendmail 99
Zestaw narzędzi wspomagających sendmail 99
Plik aliases 100
Listy dystrybucyjne 101
Serwer przesyłający pocztą pliki 105
Program obudowujący, ustawiający identyfikator 106

Rozdział 8. Rozwiązywanie problemów z listami 109
Błędy użytkowników 109
Odbijana poczta 109
Pętle pocztowe 111
Nadużywanie list dystrybucyjnych 113

Rozdział 9. Administrowanie serwerem Listproc 117
Ogólne zasady działania Listproc 117
Przygotowanie do instalacji Listproc 119
Kompilowanie i instalowanie serwera 121
Konfigurowanie serwera 131
Uruchamianie i zatrzymywanie serwera 137
Alias listproc 138
Dodawanie nowych list 138
Archiwa 144
Program ilp do interaktywnej komunikacji z serwerem 149
Codzienna eksploatacja serwera i rozwiązywanie problemów 151
Listy równorzędne i połączenia z grupami dyskusyjnymi 153
Własne modyfikacje serwera 155

Rozdział 10. Administrowanie serwerem Majordomo 161
Rzut oka na zasady działania serwera Majordomo 161
Przygotowanie do instalacji Majordomo 162
Kompilacja i instalacja 163
Konfigurowanie serwera 168
Dodawanie nowych list 169
Listy oferujące wyciągi 172
Archiwa 174
Codzienna eksploatacja serwera i rozwiązywanie problemów 177

Rozdział 11. Administrowanie serwerem SmartList 181
Zasada działania serwera 181
Przygotowanie do instalacji SmartList 181
Kompilacja i instalacja 182
Tworzenie nowych list 187
Listy moderowane 193
Listy oferujące wyciągi 194
Administrowanie listami 195
Archiwa 196
Listy równorzędne 199

Rozdział 12. Administrowanie serwerem LISTSERV Lite 201
Zasady działania serwera 201
Przygotowanie do instalacji LISTSERV Lite 203
Kompilowanie i instalowanie serwera 204
Konfigurowanie serwera 207
Uruchamianie i zatrzymywanie serwera 209
Dodawanie nowych list 210
Archiwa 214
Interfejs WWW 217
Usuwanie problemów i codzienne zarządzanie serwerem 219

Dodatek A Listprocu 221

Dodatek B Majordomo 241

Dodatek C SmartList 255

Dodatek D LISTSERV Lite 269

Skorowidz 277

10

Administrowanie serwerem Majordomo

Program zarządzania listami dystrybucyjnymi Majordomo, pierwotnie napisany przez Brenta Chapmana, a obecnie rozwijany przez Chana Wilsona, jest bardzo popularnym narzędziem zarządzania listami dystrybucyjnymi. Majordomo nie stara się robić wszystkiego, co tylko możliwe w zakresie zarządzania listami dystrybucyjnymi, lecz mimo to jest w stanie dostarczyć najbardziej nawet zaawansowanych funkcji serwera listy. Zapewnia opcjonalną możliwość zdalnego administrowania serwerem, a także zdalnego zarządzania listami przez właścicieli list, którzy sprawują prawie całkowitą kontrolę nad kwestiami związanymi z konfigurowaniem ich list — znacznie większą niż w przypadku serwerów Listproc czy SmartList.

W tym rozdziale opisano, jak instalować i konfigurować serwer Majordomo 1.94.4 i jak z niego korzystać; podczas pisania niniejszej książki była to najnowsza wersja dostępna na rynku. Jeśli korzystacie z jakiejś starszej wersji serwera Majordomo, warto ją unowocześnić, zaopatrując się w wersję 1.94.4. To istotne szczególnie w przypadku wersji starszych niż 1.92, mających poważne braki w systemie bezpieczeństwa, które mogą ułatwić włamanie do komputera serwera.

Rzut oka na zasady działania serwera Majordomo

Majordomo jest w zasadzie zbiorem skryptów Perla (i kilku programów napisanych w C), które uruchamiane są wtedy, gdy na adres serwera lub listy dystrybucyjnej nadejdzie jakaś poczta. Poczta wysłana na któryś z zarządzanych przez Majordomo adresów e-mail kierowana jest do programu *wrapper* — programu obudowującego, który zmienia identyfikatory użytkownika i grupy na odpowiednie identyfikatory użytkownika i grupy Majordomo, a następnie uruchamia skrypty Perla. Aby ten system bezpieczeństwa mógł działać, wszystkie pliki i katalogi Majordomo muszą być własnością specjalnej grupy Majordomo; zarówno użytkownik, jak i grupa muszą mieć prawa do zapisywania w nich informacji (tryb 664 lub 660 dla plików i tryb 775 lub 770 dla katalogów).

Majordomo zazwyczaj oczekuje, że użytkownicy będą się komunikować z serwerem listy za pośrednictwem adresu *majordomo*@`nazwa-hosta`. Dostarcza również skryptu odpowiadającego na pocztę nadsyłaną na adres `nazwa-listy`-*request*@`nazwa-hosta`. Skrypt ten informuje użytkownika, żeby wysyłał pocztę na adres *majordomo*. Można również polecić Majordomo, by wykonywał polecenia wysyłane na adres -*request*.

Poczta wysłana na adres listy jest przesyłana potokiem przez program *wrapper* do programu *resend*. Tenże *resend* formatuje wiadomość, która ma zostać rozesłana do subskrybentów lub przesłana do moderatora. Następnie wysyła ją na adres poczty wychodzącej z listy, który to adres jest aliasem typu `:include:` programu *sendmail*. Na ogół subskrybenci nie powinni być informowani o istnieniu tego aliasu pocztowego dla wychodzącej poczty (aliasu z członem *-outgoing*), aby zapobiec próbom obchodzenia przez nich programu *resend*.

Dzięki ukrywaniu również innych programów pod aliasem pocztowym listy Majordomo może także zapisywać wiadomości w archiwum lub tworzyć wyciągi.

Przygotowanie do instalacji Majordomo

Serwer Majordomo jest napisany w głównej mierze w Perlu, dlatego w systemie, w którym będzie działał, powinien być zainstalowany ten język. Majordomo wymaga Perla 4.036 lub Perla w wersji 5.002 lub nowszej, przy czym znacznie lepiej jest korzystać z Perla 5. Perl jest dostępny pod adresem *http://www.perl.com/CPAN/src/*.

Tworzenie użytkownika i grupy dla serwera Majordomo

Serwer Majordomo musi być zainstalowany na koncie specjalnego użytkownika. Konieczne jest utworzenie nowego użytkownika i grupy dla Majordomo oraz zainstalowanie plików w katalogu macierzystym tego użytkownika. Należy wybrać odpowiednią nazwę użytkownika. Najczęściej jest to *majordom* (pełna nazwa *majordomo* ma więcej niż osiem znaków i dlatego może sprawiać problemy niektórym systemom operacyjnym).

Konieczne jest również podjęcie decyzji, pod jakim identyfikatorem grupy będzie działać serwer Majordomo. Domyślnie to *daemon*, można jednak zmienić ten identyfikator na jakiś inny (na przykład specjalnie utworzony *majordom*).

W tym rozdziale przyjęto założenie, że został utworzony użytkownik *majordom* i grupa *majordom* oraz że katalogiem macierzystym użytkownika *majordom* jest */home/majordomo*. Dla katalogu tego powinien zostać ustawiony tryb 775.

Administrator serwera może zarządzać Majordomo za pomocą polecenia *su*, by zostać użytkownikiem *majordom*. Alternatywnie można dodać administratora serwera do grupy *majordom*. Większość procesu instalacji powinna być wykonywana przez użytkownika *majordom*.

Zdobywanie Majordomo

Kod źródłowy serwera Majordomo jest dostępny z anonimowego konta FTP pod adresem *ftp.greatcircle.com* w katalogu */majordomo*. Należy ściągnąć plik *majordomo-1.94.4.tar.gz* do katalogu */home/majordomo* i tam go rozpakować. Wspomniany plik *tar* utworzy katalog o nazwie *majordomo-1.94.4*, do którego po rozpakowaniu pliku należy się przenieść, korzystając z polecenia *cd*:

```
% gzcat majordomo-1.94.4.tar.gz | tar xvf -      W systemach odmiany System V należy użyć tar xovf
% cd majordomo-1.94.4
```

Kompilacja i instalacja

Następnie można przystąpić do kompilowania programu *wrapper* serwera Majordomo i instalowania całego systemu serwera.

Budowanie programu wrapper

Majordomo wykorzystuje program *wrapper*, ustawiający odpowiedni identyfikator użytkownika, by upewnić się, że tworzące serwer skrypty, gdy są przyzywane za pośrednictwem aliasów pliku *aliases* programu *sendmail*, będą uruchamiane przez zaufanego użytkownika z grupy serwera Majordomo. Aby zbudować program *wrapper*, należy najpierw dokonać odpowiedniej edycji pliku *Makefile*, a następnie uruchomić polecenie *make wrapper*.

Oto niektóre z ważnych makr pliku w *Makefile*:

`PERL`

Pełna ścieżka do katalogu Perla.

`W_HOME`

Katalog, w którym są przechowywane programy Majordomo. Program *wrapper* uruchamia tylko programy znajdujące się w tym katalogu. Domyślnie jest to katalog macierzysty użytkownika *majordom*. Ponieważ Majordomo generuje dość dużo różnych plików i katalogów, stwierdziłem, że o wiele bardziej eleganckim rozwiązaniem jest utworzenie podkatalogu w katalogu użytkownika i przypisanie tegoż podkatalogu makru `W_HOME`. W tym rozdziale przyjęto założenie, że utworzono katalog */home/majordomo/runtime*, który został przypisany makru `W_HOME`.

`W_USER` i `W_GROUP`

`W_USER` i `W_GROUP` są identyfikatorami użytkownika i grupy dla serwera Majordomo. Definiując je w tych makrach, należy używać numerów identyfikatorów użytkownika i grupy, a nie ich nazw.

`FILE_MODE`, `EXEC_MODE` i `HOME_MODE`

Makra te ustawiają uprawnienia, odpowiednio: dla zainstalowanych plików serwera, programów wykonywalnych i katalogu `W_HOME`. Domyślne uprawnienia dają prawa do odczytu każdemu. Jeśli te ustawienia są zbyt otwarte i mogą wiązać się ze znacznym ryzykiem dla lokalnego systemu, można spróbować przypisać `FILE_MODE` wartość 640, a `EXEC_MODE` — wartość 750.

`WRAPPER_*` i `POSIX`

Trzy makra `WRAPPER_` i makro `POSIX` dbają o to, by Majordomo prawidłowo obsługiwał uprawnienia. Instrukcje, jakiego zestawu tych makr użyć, można znaleźć w pliku *Makefile*.

`MAIL_GID`

Niektóre agenty MTA, w tym wcześniejsze wersje programu *sendmail*, nie zawsze pozwalały programom pocztowym na ustawianie nagłówka From protokołu SMTP. Majordomo stara się tak zmienić nagłówek From protokołu SMTP, aby wiadomość wyglądała na wysłaną z adresu *-request* zamiast przez użytkownika *majordom*. Jeśli w naszym systemie nie będzie można tego zrobić, konieczne okaże się przypisanie makru `MAIL_GID` identyfikatora grupy lokalnego agenta MTA.

`W_PATH`, `W_SHELL` i `W_MAJORDOMO_CF`

Makra te są przypisywane odpowiednim zmiennymi środowiskowym, gdy program *wrapper* uruchamia skrypt Perla. Wartości przypisane im domyślnie będą w większości przypadków najlepsze.

Po zmodyfikowaniu pliku *Makefile* należy użyć polecenia *make wrapper*, by zbudować program *wrapper*.

Instalowanie serwera

Aby zainstalować pozostałe skrypty Majordomo w katalogu */home/majordomo/runtime*, należy skorzystać z polecenia *make install*. Podczas instalacji powstają trzy podkatalogi (*Tools*, *bin* i *man*), ponadto na podstawie dostarczonego wraz z serwerem pliku *sample.cf* jest tworzony plik *majordomo.cf*.

Konieczne jest również dodanie ścieżki */home/majordomo/runtime/bin* do ścieżki przeszukiwania użytkownika *majordom* oraz każdego lokalnego użytkownika w grupie *majordom*, który będzie zajmował się administrowaniem serwerem.

Jeśli wiadomości wysyłane na listy dystrybucyjne mają być archiwizowane, konieczne jest zainstalowanie skryptu, zajmującego się tworzeniem archiwów. Dostępne są trzy takie skrypty: skrypt *archive.pl* tworzy archiwa w stylu LISTSERV, skrypt *archive2.pl* — bardziej wszechstronne archiwa, porządkowane według dat, a skrypt *archive_mh.pl* używa do zarządzania archiwum programu *rcvstore* z systemu pocztowego *mh*. Można zainstalować dowolny z tych trzech skryptów, używając odpowiednio poleceń *make install-archive*, *make install-archive2* i *make install-archive_mh*. W tym rozdziale przyjęto założenie, że został zainstalowany skrypt *archive2.pl*.

Instalowanie i testowanie programu wrapper

Aby zainstalować program *wrapper* jako program ustalający identyfikator użytkownika dla skryptów, należy jako użytkownik *root* wykonać polecenie *make install-wrapper*.

Następnie jako inny użytkownik — nie jako *root* ani *majordom* — należy wejść do katalogu *runtime* i uruchomić program *./wrapper config-test*. To polecenie sprawdzi aktualną konfigurację systemu serwera i poinformuje, czy zostały wykryte jakieś problemy, które należałoby naprawić. Przykład 10.1 pokazuje, jakie informacje może zwrócić to polecenie.

Przykład 10.1. Testowanie konfiguracji serwera Majordomo

```
% cd /home/majordomo/runtime
% ./wrapper config-test
--------------------- Obvious things: ---------------------
------------------ environment variables ------------------
   HOME=/home/majordomo/runtime
   LOGNAME=alansz
   MAJORDOMO_CF=/home/majordomo/runtime/majordomo.cf
   PATH=/bin:/usr/bin:/usr/ucb
   SHELL=/bin/csh
   USER=alansz
--------------------- euid/egid checks ---------------------
   effective user  = majordomo (uid 136)
   effective group = majordomo (gid 310)
```

```
---------------------- uid/gid checks ----------------------
   real      user  = majordomo (uid 136)
   real      group = majordomo (gid 310)
----------------------------  ----------------------------
  Non obvious things that cause headaches:
----------------------------  ----------------------------
Good: "require"d /home/majordomo/runtime/majordomo.cf okay.
Good: found ctime.pl okay.
Good: found majordomo_version.pl okay.
Good: found majordomo.pl okay.
Good: found shlock.pl okay.
Good: found config_parse.pl okay.
You're running Majordomo Version 1.94.4.
----> Majordomo home directory is /home/majordomo/runtime.
------------------- Include directories -------------------
  /home/majordomo/runtime
  /usr/local/lib/perl5/sun4-sunos/5.003
  /usr/local/lib/perl5
  /usr/local/lib/perl5/site_perl/sun4-sunos
  /usr/local/lib/perl5/site_perl
  /usr/local/lib/perl5/sun4-sunos
  .
---------------------------- Home ----------------------------
Good: changedir to /home/majordomo/runtime succeeded.
Good: Created a mock lock file.
---------------------- temp directory ----------------------
Good: Created a temp file in $TMPDIR (/usr/tmp).
---------------------- list directory ----------------------
Good: list directory /home/majordomo/runtime/lists has good permissions.
--------------------------- log ---------------------------
Good: logfile /home/majordomo/runtime/Log exists and is writeable.
------------------------- Mailers -------------------------
You have defined a mailer for delivery.
Attempting to verify that this is a valid mailer...looks okay.
You have defined a mailer for delivering administrative messages.
Attempting to verify that this is a valid mailer...looks okay.
------------------ Checking majordomo.cf ------------------
Checking to see if there are new variables that should be in
your majordomo.cf file...Nope, none that I see.
----------------------- end of tests -----------------------
Nothing bad found!  Majordomo _should_ work correctly.
```

Tworzenie struktury katalogów

Katalogi list zarządzanych przez Majordomo, archiwum i katalogi robocze wyciągów można umieścić w całkowicie dowolnym miejscu. Plik *majordomo.cf* informuje, gdzie można je znaleźć (więcej na ten temat poniżej, w podrozdziale poświęconym konfigurowaniu serwera). Konieczne jednak będzie utworzenie tych katalogów[1]. Wygodną strukturą katalogów jest umieszczenie w katalogu */home/majordomo* trzech następujących podkatalogów:

lists

Katalog ten będzie przechowywać pliki związane z listami. Każda lista posiada własny plik konfiguracyjny, plik subskrybentów, plik hasła oraz inne.

[1] Warto zauważyć, że choć program *wrapper* utworzył dla list domyślny katalog *lists* w katalogu macierzystym serwera `W_HOME`, nie ma obowiązku korzystania z tego katalogu.

archives

Katalog ten będzie zawierać pliki przechowywane dla każdej z list w archiwum.

digests

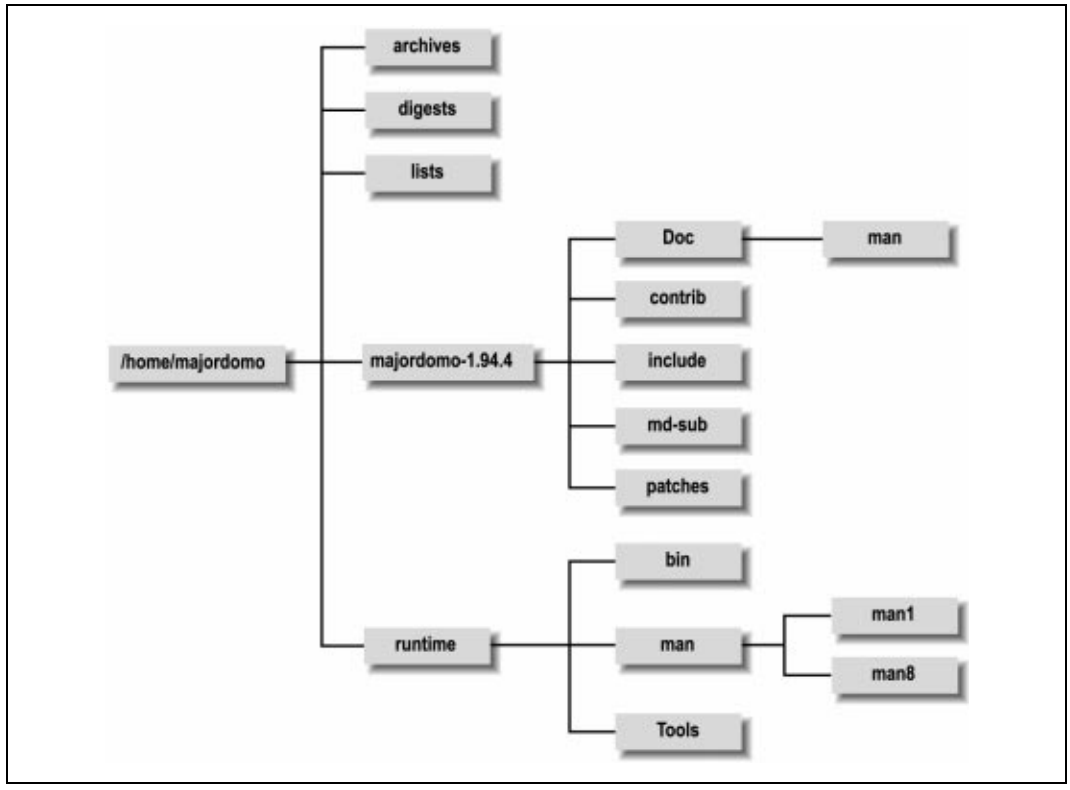
Pełni funkcję katalogu roboczego dla przygotowującego wyciągi programu *digest*.

Oto polecenia tworzące taką strukturę katalogów:

```
% cd /home/majordomo
% mkdir lists archives digests
% chmod 775 lists archives digests
% ls -lgdF lists archives digests
drwxrwxr-x  3 majordom majordom      512 Apr  1 14:26 archives/
drwxrwxr-x  3 majordom majordom      512 Apr  1 14:26 digests/
drwxrwxr-x  3 majordom majordom      512 Apr  1 14:26 lists/
```

Inną możliwością jest przechowywanie archiwów list jako podkatalogów katalogu *lists* o nazwach kończących się na *.archive* (archiwum) lub *.files* (pliki). Na przykład katalogiem archiwum listy dystrybucyjnej *zodiak* będzie katalog */home/majordomo/lists/zodiak.archive*. W tym przypadku nie ma potrzeby tworzenia osobnego katalogu *archives*.

Rysunek 10.1 pokazuje, jakie katalogi powinny znajdować się w katalogu */home/majordomo* po zakończeniu instalacji serwera.



Rysunek 10.1. Struktura katalogów serwera Majordomo

Katalog *runtime* (lub, bardziej ogólnie, katalog `W_HOME`) zawiera programy tworzące serwer Majordomo, plik konfiguracyjny serwera oraz wspomagające skrypty (których nazwy kończą się na *.pl*). Oto zestawienie najczęściej używanych programów serwera Majordomo:

bounce-remind

Program odpowiedzialny za zarządzanie subskrybentami, których adresy odbijają wysyłaną do nich pocztę.

digest

Ten program łączy wiadomości w wyciągi. Przywoływany jest spod aliasu pocztowego utworzonego dla listy oferującej wyciągi w programie *sendmail*.

majordomo

Serwer obsługujący polecenia wysyłane do Majordomo. Odpowiada na pocztę wysyłaną na adres *majordomo* (oraz, jeśli administrator tego sobie życzy, również na adresy z końcówką *-request* dla każdej z obsługiwanych list dystrybucyjnych) i przetwarza polecenia przesłane w wiadomości.

new-list

Zwraca wiadomość informującą, że lista jest ciągle w fazie tworzenia i nie jest jeszcze gotowa do rozsyłania wiadomości. Przywoływany jest za pośrednictwem aliasu pocztowego listy z pliku *aliases* programu *sendmail*, gdy lista niedawno powstała i nadal zbiera subskrybentów.

request-answer

Program ten zwraca wiadomość informującą, że wszelkie polecenia administracyjne należy wysyłać na adres *majordomo*`@host`, a nie `nazwa-listy-request@host`.

resend

Program ten jest odpowiedzialny za rozsyłanie poczty nadchodzącej na listę dystrybucyjną i opcjonalnie za dodawanie lub usuwanie specjalnych nagłówków, wykrywanie poleceń administracyjnych omyłkowo przesłanych na listę i zapobieganie temu, aby wiadomości nie przekroczyły określonej długości.

Katalog *runtime* zawiera również trzy podkatalogi: *bin*, *man* i *Tools*. Katalog *bin* przechowuje dwa skrypty Perla, dość przydatne właścicielom list (*approve* i *bounce*), oraz jeden skrypt dla administratora serwera (*medit*). Właściciele list moderowanych mogą przesyłać wiadomości potokiem do skryptu *approve*, aby ułatwić sobie aprobowanie wiadomości nadchodzących na listę. Skrypt *bounce* ułatwia właścicielom list wypisywanie z listy subskrybentów, którzy odbijają wysyłaną do nich pocztę. Skrypt *medit* służy do edytowania plików serwera Majordomo. Blokuje pliki na czas edycji, by zapobiec ich przypadkowemu użyciu przez Majordomo, gdy są w trakcie modyfikacji, uruchamia odpowiedni edytor używany przez administratora, a na koniec odblokowuje pliki.

Katalog *man* zawiera dwie strony podręcznika (sformatowane za pomocą *nroff*), które można albo zainstalować wśród stron podręcznika systemowego, przywoływanych za pomocą polecenia *man*, albo rozprowadzić wśród właścicieli list obsługiwanych przez Majordomo. Plik *man/man1/approve.1* jest stroną podręcznika opisującą skrypt *approve*, a plik *man/man8/majordomo.1* — starą stroną podręcznika opisującą sam serwer Majordomo.

Katalog *Tools* zawiera narzędzia, których serwer Majordomo używa do archiwizowania wiadomości, indeksowania ich oraz do tworzenia z nich wyciągów.

Tworzenie aliasów dla serwera Majordomo

Jako użytkownik *root* należy zmodyfikować plik *aliases* programu *sendmail*, dodając do niego następujące aliasy:

```
majordomo: "|/home/majordomo/runtime/wrapper majordomo"
owner-majordomo: alan,nobody
majordom: owner-majordomo
Majordomo-Owner: owner-majordomo
nobody: /dev/null
```

Alias *majordomo* jest sercem całego systemu. Uruchamia on program *wrapper*, który z kolei inicjuje pracę serwera rozpatrującego polecenia administracyjne *W_HOME/majordomo*. Alias *owner-majordomo* jest wykorzystywany w charakterze nagłówka From protokołu SMTP, nadawanego rozsyłanym wiadomościom, i jednocześnie jest adresem, na który wysyłane są wszelkie informacje o problemach z rozsyłaniem wiadomości.

Alias *Majordomo-Owner* jest wykorzystywany przez niektóre witryny zamiast aliasu *owner-majordomo*. Warto więc zdefiniować go na wypadek, gdyby ktoś chciał wysłać wiadomość właśnie na ten adres.

Alias *nobody* odrzuca wiadomości, zapisując je w urządzeniu */dev/null*. Jak się przekonamy w dalszej części rozdziału, alias ten jest dodawany do listy innych aliasów odbiorcy, aby „przekonać" program *sendmail*, że rozsyła wiadomość do wielu odbiorców. Jeśli z jakichś powodów nie można użyć *nobody*, należy wybrać inną nazwę aliasu i użyć jej wszędzie tam, gdzie w tej książce wspominany jest alias *nobody* serwera Majordomo.

Należy pamiętać, aby po dodaniu tych aliasów uruchomić program *newaliases*.

Konfigurowanie serwera

Oprócz tego, że każda lista Majordomo ma własny plik konfiguracyjny, również sam serwer Majordomo posiada własny plik konfiguracyjny *majordomo.cf*, który kontroluje globalne właściwości systemu serwera. Konieczne będzie wprowadzenie w pliku *majordomo.cf* (który można znaleźć w katalogu definiowanym za pomocą makra `W_HOME`) pewnych poprawek, podających lokalizację katalogów i plików serwera Majordomo. Plik *majordomo.cf* został napisany w Perlu, tak więc każdy wiersz (z wyjątkiem wierszy komentarzy) powinien kończyć się średnikiem (`;`), a ostatni wiersz musi mieć postać: `1;`.

Plik konfiguracyjny *majordomo.cf* jest opatrzony przejrzystymi komentarzami i jego zrozumienie nie sprawia żadnych problemów. Oto przykładowy fragment tego pliku:

```
# If you are using majordomo at the -request address, set the
# following variable to 1. This affects the welcome message that is
# sent to a new subscriber as well as the help text that is generated.
#
$majordomo_request = 0;

# If you have lists that have who turned off, but still allow which
# requests to work for subscribed members, and you don't want to have
# "which @" to act like a who, the variable $max_which_hits sets the
```

```
# number of hits that are allowed using which before an error is
# returned. Arguably this should be a per list settable number.
#
$max_which_hits = 5;
```

Zmiennej `$majordomo_request` należy przypisać wartość `1`, jeśli każda z zarządzanych przez serwer list powinna posiadać również alias (adres) z końcówką *-request*. Po włączeniu tej zmiennej adresy *-request*, zamiast uruchamiać skrypt *request-answer*, informujący użytkownika, że polecenia należy przesyłać na inny adres, przesyłać będą potokiem na adres *majordomo* wiadomości zawierające polecenia. Zmienna `$max_which_hits` zapobiega omijaniu przez użytkowników ograniczeń nałożonych na polecenie `who` poprzez zadawanie pytania, którzy użytkownicy mają w swoich adresach „małpę" (`@`) lub kropkę (`.`).

Warto uruchamiać po każdej modyfikacji pliku *majordomo.cf* polecenie *./wrapper config-test*, by upewnić się, że w wyniku modyfikacji nie pojawiły się w pliku konfiguracyjnym żadne błędy.

Dodawanie nowych list

Aby utworzyć prostą listę dystrybucyjną, należy wykonać następujące czynności:

1. Utwórz w katalogu *lists* plik subskrybentów i plik hasła. Plik subskrybentów powinien mieć taką samą nazwę jak lista, powinien być początkowo pusty i mieć przypisany tryb 664. Plik hasła musi mieć nazwę *`nazwa-listy.passwd`*. Powinien zawierać tylko hasło administracyjne listy i należy mu przypisać tryb 660. Oba pliki muszą być własnością użytkownika i grupy serwera Majordomo. Poniżej przedstawiono przykład, jak będą wyglądać odpowiednie polecenia używane do tworzenia plików listy dystrybucyjnej *kapelusznicy*, wspomnianej w rozdziale czwartym (poświęconym opisowi Majordomo z punktu widzenia zarządcy listy):

   ```
   % su - majordom
   % cd lists
   % touch kapelusznicy
   % chmod 644 kapelusznicy
   % echo "fez" > kapelusznicy.passwd
   % chmod 600 kapelusznicy.passwd
   ```

   Sprawdź dwukrotnie, czy ustawienia właściciela, grupy i uprawnień dla tych plików są prawidłowe.

2. Utwórz plik, który zawierać będzie argumenty wiersza poleceń dla programu *resend*. Umieszczenie argumentów dla *resend* w pliku sprawi, że użytkownikom trudniej będzie odkryć, jakiego aliasu lista używa dla poczty wychodzącej[2]. Plikowi temu można nadać dowolną nazwę, niemniej najwygodniejsza jest *`lista.resend`*. Tryb pliku należy ustawić na 644:

   ```
   % echo "-l kapelusznicy -h kapelusze.com kapelusznicy-doit,nobody"
     > kapelusznicy.resend
   % chmod 644 kapelusznicy.resend
   ```

[2] Użytkownik może próbować odkryć alias pocztowy listy dla poczty wychodzącej, łącząc się poprzez telnet z portem SMTP hosta serwera i przesyłając polecenie `EXPN` *`nazwa-listy`*. Jeśli użytkownik przekona się, że rozwinięciem aliasu jest program *resend* pobierający argumenty z pliku, nie będzie mógł ustalić, jakie argumenty są mu rzeczywiście przesyłane.

Argument `-l` podaje nazwę listy, podczas gdy argument `-h` — nazwę hosta, na którym działa lista. Ostatni argument jest aliasem lub aliasami (oddzielonymi przecinkami), poprzez które poczta listy jest wysyłana. Dodanie aliasu *nobody* sprawia, że program *sendmail* — „przekonany", że wysyła pocztę do kilku osób — nie będzie dodawał do wysyłanych wiadomości nagłówka Received podającego adres *kapelusznicy-doit*.

3. Dodaj aliasy pocztowe dla tworzonej listy do pliku *aliases* programu *sendmail*. Oto zestaw odpowiednich aliasów dla listy *kapelusznicy*, działającej na hoście *kapelusze.com*:

```
owner-kapelusznicy: sombrero@casa.mx,nobody
kapelusznicy:
  "|/home/majordomo/runtime/wrapper resend
   @/home/majordomo/lists/kapelusznicy.resend"
kapelusznicy-approval: owner-kapelusznicy
kapelusznicy-doit: :include:/home/majordomo/lists/kapelusznicy
owner-kapelusznicy-doit: owner-kapelusznicy
kapelusznicy-request:
  "|/home/majordomo/runtime/wrapper request-answer kapelusznicy"
owner-kapelusznicy-request: owner-kapelusznicy
```

Zostały tu utworzone następujące aliasy:

owner-kapelusznicy

Ten alias pełni funkcję adresu właściciela listy. Na ten adres program *sendmail* wysyła komunikaty o błędach. Będzie on również używany jako nagłówek From protokołu SMTP, nadawany wiadomościom wysyłanym na listę. Dołączenie aliasu *nobody* zapobiega umieszczeniu przez wersję 8 programu *sendmail* w nagłówku From protokołu SMPT adresu *sombrero@casa.mx* zamiast *owner-kapelusznicy@kapelusze.com*. Wiele z pozostałych aliasów odwołuje się do tego aliasu. W szczególności każdy alias, który generuje pocztę, posiada wewnątrz alias *owner-* wskazujący na alias *owner-kapelusznicy*.

kapelusznicy

Alias ten przesyła potokiem wiadomości nadchodzące na listę do programu *wrapper*. Ustawia on właściwe identyfikatory użytkownika i grupy oraz uruchamia program *resend*, który z kolei odpowiednio przetwarza wiadomość przed rozesłaniem. Wiersz `@/home/majordomo/lists/kapelusznicy.resend` poleca programowi *resend* pobrać swoje argumenty wiersza poleceń z pliku */home/majordomo/lists/kapelusznicy.resend*, o czym wspomniano powyżej.

kapelusznicy-approval

Ten alias podaje adresy osoby lub grupy ludzi, którzy aprobować będą wszelkie prośby o zapisanie na listę zamkniętą, prośby o zapisanie innej osoby oraz wiadomości wysyłane na listę moderowaną. Będzie to zazwyczaj adres właściciela listy (w naszym przykładzie: *owner-kapelusznicy*).

kapelusznicy-doit

To jest rzeczywista lista subskrybentów. Na ten alias program *resend* będzie przesyłał wiadomości rozsyłane na listę. Bardziej klasyczne nazwy tego aliasu to *kapelusznicy-outgoing* lub na przykład *kapelusznicy-dist*, niemniej większość właścicieli list wolałoby, aby ich subskrybenci (a także inne osoby) nie byli w stanie ustalić, jak naprawdę brzmi ten alias, i nie wysyłali wiadomości wprost do niego, omijając program *resend*.

kapelusznicy-request

Wiele z internetowych list dystrybucyjnych (szczególnie tych tworzonych ręcznie) używa *nazwa-listy-request* jako adresu do wysyłania poleceń. Majordomo oferuje dwa sposoby radzenia sobie z poleceniami wysyłanymi na adres *nazwa-listy-request*. Metoda pokazana w przykładzie powyżej instruuje program *wrapper*, aby uruchomił skrypt *request-answer*, który odeśle użytkownikowi wiadomość z informacją, że polecenia należy wysyłać na adres *majordomo@hats.com*. Można również nakazać Majordomo, by akceptował polecenia wysyłane na adres *kapelusznicy-request* używając aliasu w następującej postaci:

```
kapelusznicy-request:
  "|/home/majordomo/runtime/wrapper majordomo -l kapelusznicy"
```

Opcja `-l` podaje nazwę listy. Jeśli zdecydujemy w ten sposób, że serwer ma odpowiadać również na polecenia wysłane na adres *-request*, należy zmiennej `$majordomo_request` w pliku *majordomo.cf* przypisać wartość `1`. W wyniku tej zmiany, wprowadzonej w pliku konfiguracyjnym, wszystkie obsługiwane przez serwer listy wykorzystywać będą jako adres do wysyłania poleceń adres z końcówką *-request*. Innymi słowy, nie ma możliwości korzystania z obu tych technik jednocześnie. Kiedy zmiennej `$majordomo_request` zostanie przypisana wartość 1, wszystkie pliki pomocy i inne automatyczne porady będą zalecać użytkownikom, by ci wysyłali polecenia na adres listy z dodaną końcówką *-request*.

4. Poinstruuj właściciela listy, jak korzystać z poleceń `config`, `newintro` i `newinfo`, by mógł on skonfigurować swoją listę, przygotować wiadomość powitalną i plik informacyjny. Dla listy *kapelusznicy* jej właściciel *sombrero@casa.mx* powinien przesłać na adres *majordomo@kapelusze.com* następujące polecenia:

```
config kapelusznicy hasło
newintro kapelusznicy hasło
Tekst wiadomości powitalnej
EOF
newinfo kapelusznicy hasło
Zawartość pliku informacyjnego
EOF
```

Polecenie `config` tworzy domyślny plik konfiguracyjny listy dystrybucyjnej i przesyła go pocztą elektroniczną właścicielowi listy, który może go odpowiednio zmodyfikować i zainstalować za pomocą polecenia `newconfig`. Szczegółowe informacje na ten temat można znaleźć w rozdziale czwartym. Polecenia `newintro` i `newinfo` tworzą pliki *nazwa-listy.intro* i *nazwa-listy.info* z zawartością podaną przez właściciela listy. Każdy z tych plików musi być zakończony wierszem zaczynającym się od trzech dużych liter EOF, sygnalizujących koniec pliku.

Po wykonaniu powyższych czynności lista jest już gotowa i właściciel listy posiada kopię jej pliku konfiguracyjnego. Administrator serwera nie musi wykonywać żadnych działań, aby przygotować listę moderowaną. Właściciel listy może sam ustalić, czy i w jaki sposób lista będzie moderowana, edytując jej plik konfiguracyjny. Zazwyczaj warto jeszcze przesłać właścicielowi listy kopię pliku *Doc/list-owner-info*, zawierającego wiele użytecznych informacji, które mogą przydać się początkującym właścicielom list.

Listy oferujące wyciągi

Majordomo może przygotować dodatkową bliźniaczą listę, oferującą wiadomości zebrane w wyciągi. Ażeby zilustrować poszczególne etapy tworzenia listy oferującej wyciągi, posłużę się przykładem list *kapelusznicy* i *kapelusznicy-digest*. Oto co należy kolejno zrobić:

1. Upewnij się, że program *digest* został zainstalowany w katalogu zawierającym plik konfiguracyjny serwera *majordomo.cf*. Domyślnie program *digest* znajduje się w podkatalogu *Tools/*, należy więc użyć następujących poleceń:

```
% cd /home/majordomo/runtime
% ln Tools/digest digest
```

2. Utwórz zwykłą listę (nie rozsyłającą wyciągów) oraz następnie listę rozsyłającą wyciągi, jak to zostało opisane wcześniej w podrozdziale na temat tworzenia list.

3. Zmodyfikuj alias poczty wychodzącej dla zwykłej listy, dołączając do niego odwołanie do programu *digest*. Upewnij się, że alias pocztowy listy dla listy oferującej wyciągi wskazuje zwykłą listę. Oto pełny zestaw odpowiednich aliasów:

```
kapelusznicy-doit: :include:/home/majordomo/lists/kapelusznicy,
  "|/home/majordomo/runtime/wrapper digest -r -C
   -l kapelusznicy-digest kapelusznicy-digest-outgoing"
kapelusznicy:
  "|/home/majordomo/runtime/wrapper resend
   @/home/majordomo/lists/kapelusznicy.resend"
owner-kapelusznicy: sombrero@casa.mx,nobody
kapelusznicy-approval: owner-kapelusznicy
owner-kapelusznicy-doit: owner-kapelusznicy
kapelusznicy-request:
  "/home/majordomo/runtime/wrapper request-answer kapelusznicy"
owner-kapelusznicy-request: owner-kapelusznicy
kapelusznicy-digest: kapelusznicy
kapelusznicy-digest-doit:
  :include:/home/majordomo/lists/kapelusznicy-digest
owner-kapelusznicy-digest: owner-kapelusznicy
kapelusznicy-digest-approval: owner-kapelusznicy
owner-kapelusznicy-digest-doit: owner-kapelusznicy
kapelusznicy-digest-request:
  "/home/majordomo/runtime/wrapper
   request-answer kapelusznicy-digest"
owner-kapelusznicy-digest-request: owner-kapelusznicy
```

   Warto pamiętać, że poszczególne aliasy w pliku */etc/aliases* nie mogą rozciągać się na więcej niż jeden wiersz. Tutaj zostały rozpisane w kilku wierszach z przyczyn estetycznych (dla większej przejrzystości).

4. Utwórz katalog roboczy dla programu *digest*, definiowany w zmiennej `$digest_work_dir` w pliku konfiguracyjnym serwera:

```
% cd /home/majordomo
% mkdir digests/kapelusznicy-digest
% chmod 770 digests/kapelusznicy-digest
% chgrp majordomo digests/kapelusznicy-digest
```

5. Jeśli chcesz archiwizować wyciągi, utwórz (opierając się na wartościach przypisanych odpowiednim zmiennym w pliku *majordomo.cf*) katalog archiwum dla tych wyciągów o nazwie: `$filedir`*/kapelusznicy-digest*`$filedir_suffix`[3]:

```
% cd /home/majordomo
% mkdir archives/kapelusznicy-digest
% chmod 770 archives/kapelusznicy-digest
% chgrp majordomo archives/kapelusznicy-digest
```

   Jeśli nie zamierzasz przechowywać w archiwum wyciągów z wiadomości, możesz przygotować zadanie *cron*, które będzie co jakiś czas oczyszczało archiwa:

```
cd /home/majordomo/archives &&
  find *-digest/. -type f -mtime +1 -exec rm {} \;
```

6. Utwórz plik *config* dla listy oferującej wyciągi, kopiując plik konfiguracyjny zwykłej listy. Następnie zmodyfikuj ten plik, wprowadzając takie oto zmiany:

   - Zmień `digest_name` z *kapelusznicy* na *kapelusznicy-digest*.
   - Przypisz dyrektywie `maxlenght` maksymalną długość (w bajtach), którą wyciąg może osiągnąć, zanim zostanie automatycznie rozesłany.
   - Przypisz dyrektywie `reply-to` listę *kapelusznicy* (a nie *kapelusznicy-digest*), aby wszelkie odpowiedzi trafiały na listę nieoferującą wyciągów.
   - Zmień dyrektywę `sender` z *owner-kapelusznicy* na *owner-kapelusznicy-digest*.
   - Sprawdź i ewentualnie popraw dyrektywy `message_headers`, `message_fronter` i `message_footer`. Nagłówki i stopki dodawane do wiadomości (oraz nagłówki pocztowe) funkcjonują inaczej w przypadku list oferujących wyciągi.

Wyciąg jest rozsyłany do subskrybentów, gdy osiągnie rozmiar (w bajtach) większy niż zdefiniowany w `maxlenght` lub będzie posiadał więcej wierszy, niż to zostało określone w dyrektywie `digest_maxlines`, lub gdy będzie miał więcej dni niż wartość podana w `digest_maxdays`. Właściciel listy może również wysłać wyciąg w dowolnym momencie, przesyłając na adres *majordomo* polecenie `mkdigest` *`nazwa-listy hasło`*.

Jeśli wyciągi mają być rozsyłane o określonych porach (na przykład raz dziennie), w aliasie pocztowym kierującym do programu *digest* opcję `-r` można zastąpić opcją `-R`, co spowoduje, że kolejne wiadomości będą dodawane do wyciągu, nie powodując jednak wysłania go:

```
kapelusznicy-doit: :include:/home/majordomo/lists/kapelusznicy,
  "|/home/majordomo/runtime/wrapper digest -R -C
   -l kapelusznicy-digest kapelusznicy-digest-outgoing"
```

Następnie należy przygotować zadanie *cron*, które będzie rozsyłało regularnie wyciągi (jeśli jakiś będzie gotowy do rozesłania), przywołując program *digest* opcją `-p`. Poniżej znajduje się przykład, jak rozsyłać wyciągi co tydzień:

```
0  8  *  *  1 /home/majordomo/runtime/wrapper digest -p -C -l nazwa-listy
```

[3] W przypadku serwera Majordomo 1.93 trzeba utworzyć ten katalog, nawet jeśli nie zamierzamy zachowywać wyciągów w archiwum. Lepiej jednak unowocześnić serwer do wersji 1.94.4.

Opcja `-p` poleca programowi *digest* rozesłać wyciąg, jeśli to tylko możliwe. Opcja `-C` wskazuje, że informacje konfiguracyjne powinny być odczytywane z pliku *config* listy, a opcja `-l nazwa-listy` określa, o którą listę chodzi.

Samotna lista oferująca wyciągi

Listy rozsyłające wyciągi towarzyszą zazwyczaj zwykłym listom nieoferującym wyciągów. Co jednak uczynić, aby utworzyć listę dystrybucyjną Majordomo, która będzie dostępna tylko w formie wyciągów i nie będzie moderowana, bo nie mamy ochoty robić wyciągów ręcznie? Pokażę to na przykładzie listy *kapelusznicy-digest*, tym razem jednak bez towarzyszącej jej listy *kapelusznicy*.

Aby utworzyć taką listę, należy wykonać instrukcje, które opisywały tworzenie listy oferującej wyciągi, towarzyszącej zwykłej liście Majordomo. Jednakże dyrektywie `reply-to` w pliku konfiguracyjnym listy *kapelusznicy-digest.config* powinna zostać przypisana wartość *kapelusznicy-digest*, a dyrektywie `sender` — wartość *owner-kapelusznicy-digest*. Nie należy używać dyrektyw nagłówka i stopki wiadomości: `message_fronter` i `message_footer`, ponieważ będą dodawane do każdej wiadomości przesyłanej w wyciągu oraz do samej wiadomości wyciągu, powtarzając się w ten sposób wielokrotnie. W katalogu *lists* należy utworzyć plik *kapelusznicy-digest.resend*, zawierający wiersz:

```
-l kapelusznicy -digest -h hats.com kapelusznicy-to-digest
```

Na koniec do pliku aliasów pocztowych agenta MTA trzeba dodać następujące aliasy:

```
kapelusznicy-digest:
 "|/home/majordomo/runtime/wrapper resend @kapelusznicy-digest.resend"
kapelusznicy-to-digest:
 "|/home/majordomo/runtime/wrapper digest -r -C
  -l kapelusznicy-digest kapelusznicy-digest-outgoing"
kapelusznicy-digest-outgoing:
 :include:/home/majordomo/lists/kapelusznicy-digest
kapelusznicy-digest-request:
 "|/home/majordomo/runtime/wrapper request-answer kapelusznicy-digest"
owner-kapelusznicy-digest: sombrero@casa.mx
kapelusznicy-digest-owner: owner-kapelusznicy-digest
kapelusznicy-digest-approval: owner-kapelusznicy-digest
```

Aby polecenie `mkdigest` działało prawidłowo, alias pocztowy ukrywający listę subskrybentów musi nosić nazwę *kapelusznicy-digest-outgoing*.

Tworzenie list oferujących tylko wyciągi jest znacznie prostsze w serwerze SmartList. Z kolei w serwerach Listproc i LISTSERV Lite nie można tego zrobić bez moderowania lub wyłączenia polecenia `set`.

Archiwa

Serwer Majordomo może archiwizować pliki tekstowe i wiadomości wysyłane na listę. Mimo że nie oferuje haseł ograniczających dostęp do archiwum, umożliwia ograniczenie korzystania z poleceń `get` i `index` tylko do subskrybentów listy.

Majordomo dopuszcza znaczną swobodę w zakresie wyboru miejsca lokalizacji archiwów. Zmienne `$filedir` i `$filedir_suffix` z pliku *runtime/majordomo.cf* pozwalają określić lokalizację archiwów. Archiwum określonej listy znajdować się będzie w katalogu `$filedir/`*`nazwa-listy`*`$filedir_suffix`. Poniżej przedstawiono trzy miejsca, w których można zlokalizować archiwa:

Podkatalogi katalogu lists

Aby umieścić archiwa w podkatalogach katalogu *lists*, który zawiera informacje na temat list, należy zmodyfikować plik konfiguracyjny serwera *runtime/majordomo.cf* w następujący sposób:

```
$filedir = "$listdir";
$filedir_suffix = ".archive";
```

Archiwa będą teraz przechowywane w katalogach *lists/`nazwa-listy`.archive*.

W katalogu archives

Aby umieścić archiwa w katalogu o nazwie *archives* (archiwa), należy utworzyć odpowiedni katalog */home/majordomo/archives* i zmodyfikować plik *runtime/majordomo.cf* w następujący sposób:

```
$filedir = "/home/majordomo/archives";
$filedir_suffix = "";
```

Po tej modyfikacji archiwa będą przechowywane w katalogach *archives/`nazwa-listy`*.

W anonimowym katalogu FTP

Jeśli użytkownik *majordom* ma uprawnienia do tworzenia katalogów w obszarze dostępnym dla anonimowego użytkownika FTP, można tam właśnie przechowywać archiwa list. Jeśli na przykład anonimowy katalog FTP dla archiwów nosi nazwę */home/ftp/pub/list-archives*, to plik *runtime/majordomo.cf* należy zmodyfikować w następujący sposób:

```
$filedir = "/home/ftp/pub/list-archives";
$filedir_suffix = "";
```

Archiwa będą teraz przechowywane w katalogach */home/ftp/pub/list-archives/`nazwa-listy`*.

Kolejną istotną zmienną w pliku *runtime/majordomo.cf* jest `$index_command`, definiująca polecenie służące do przeglądania zawartości katalogu archiwum. Domyślnie zmiennej `$index_command` jest przypisana wartość */bin/ls -lRL*. Polecenie *ls* służy w tym przypadku do wyświetlania szczegółowej listy, podającej zawartość katalogu archiwum i jego podkatalogów. Aby zmienić format wyświetlania zawartości archiwum, należy przypisać zmiennej `$index_command` inne polecenie (lub program), które zdoła na przykład utworzyć indeks zawartości.

Automatyczne archiwizowanie list

Jeśli lista przygotowuje wyciągi, są one automatycznie umieszczane w katalogu archiwum listy i opatrywane odpowiednim numerem tomu i wyciągu w tomie. Majordomo może również automatycznie archiwizować osobne wiadomości, zapisując je w pojedynczym pliku archiwum, który zawierać będzie zebrane pliki z całego roku, miesiąca lub dnia. Program *archive2.pl* dodaje do archiwum wszystkie rozsyłane przez listę wiadomości. Może również przetwarzać plik skrzynki pocztowej i zapisywać w archiwum każdą wiadomość przechowywaną w tym pliku.

Program *archive2.pl* nie jest instalowany wraz z pozostałą częścią serwera Majordomo. Można znaleźć go w katalogu narzędzi pod nazwą *runtime/Tools/archiwe2.pl*. Aby korzystać z niego w codziennej pracy, najprawdopodobniej trzeba go przenieść do katalogu *runtime*:

```
% cd runtime
% cp Tools/archive2.pl archive
% chown majordom.majordom archive
% chmod 755 archive
```

Składnia uruchamiania programu *archive2.pl* jest następująca:

```
archive2.pl -f bazowa-nazwa opcja-wejścia [opcja-daty] [plik(i)]
```

Wiadomości, które mają zostać zarchiwizowane, mogą zostać podane jako nazwy plików w wierszu poleceń lub za pośrednictwem standardowego strumienia wejścia. Jeśli *opcją-wejścia* będzie `-u`, program zakłada, że danymi wejściowymi dla niego będzie jedna lub więcej wiadomości w formacie UNIX-owej skrzynki pocztowej. Jeśli zostanie podana opcja `-a`, program zakłada, że otrzyma pojedynczą wiadomość e-mail. Na podstawie *bazowa-nazwa* tworzona jest nazwa pliku zapisywanego w archiwum. Rzeczywista nazwa tego pliku zależeć będzie jeszcze od *opcji-daty*, jeśli jakaś zostanie podana:

`-d` Pliki archiwalne tworzone są codziennie i opatrywane nazwą: *bazowa-nazwa.RRMMDD*.

`-m` Pliki archiwalne są tworzone co miesiąc i opatrywane nazwą: *bazowa-nazwa.RRMM*.

`-y` Pliki archiwalne są tworzone co rok i opatrywane nazwą: *bazowa-nazwa.RR*.

A jeśli nie zostanie podana żadna *opcja-daty*, plik archiwum otrzymuje nazwę taką jak *nazwa-bazowa*.

Aby utworzyć listę automatycznie archiwizowaną, należy odpowiednio zmienić aliasy pocztowe w pliku *aliases* lokalnego agenta MTA, umieszczając w nich odwołania do programu *archive2*.pl. Poniżej podano przykład, jak przygotować aliasy umożliwiające archiwizowanie listy *pluszowe-lalki@pluszaki.com*, tworząc miesięczne pliki archiwalne o nazwach w rodzaju *pluszowe-lalki.9601*.

```
pluszowe-lalki:\
 "|/home/majordomo/runtime/wrapper resend -l
  pluszowe-lalki pluszowe-lalki-outgoing"
pluszowe-lalki-outgoing: \
  :include:/home/majordomo/lists/pluszowe-lalki,pluszowe-lalki-archive2.pl
pluszowe-lalki-request: \
  "|/home/majordomo/runtime/wrapper request-answer pluszowe-lalki"
owner-pluszowe-lalki: misio
pluszowe-lalki-owner: owner-pluszowe-lalki
```

```
pluszowe-lalki-approval: owner-pluszowe-lalki
pluszowe-lalki-archive2.pl: "|/home/majordomo/runtime/wrapper archive2.pl
  -f /home/majordomo/lists/pluszowe-lalki-archive/pluszowe-lalki
  -m -a"
```

Program *archive2.pl* może być również wykorzystywany do dodawania do archiwum określonych wiadomości e-mail lub zawartości skrzynek pocztowych, jeśli jako *opcja-wejścia* zostanie podana opcja `-u`.

Zapisywanie w archiwum innych plików

Można również umieścić w archiwum Majordomo dowolny inny plik tekstowy, po prostu kopiując go do katalogu archiwum i upewniając się, że jest własnością użytkownika i grupy serwera Majordomo i może być przez nich odczytywany.

Majordomo nie może przesyłać plików binarnych w odpowiedzi na polecenia żądające udostępnienia zawartości archiwum. Aby udostępnić użytkownikom również pliki binarne, trzeba je najpierw zakodować za pomocą programu *uuencode* i ów zakodowany plik zapisać w archiwum. Można też skorzystać z programu *ftpmail*.

Jeśli archiwa Majordomo są zlokalizowane w obszarze dostępnym dla anonimowego użytkownika FTP, użytkownicy będą mogli pobierać je za pomocą FTP. Serwery *ftpmail* umożliwiają pobieranie plików z anonimowych witryn FTP za pośrednictwem poczty elektronicznej. W serwer Majordomo wbudowana jest obsługa *ftpmail*. Może on przesyłać polecenia pobierania plików do serwera *ftpmail*, który zwróci wtedy użytkownikowi odpowiedni (już zakodowany) plik.

Aby zamiast ze standardowego sposobu pobierania plików z archiwum Majordomo korzystać z *ftpmail*, należy upewnić się, że archiwa Majordomo są dostępne dla anonimowego użytkownika FTP, i usunąć w pliku *runtime/majordomo.cf* znaki komentarza sprzed następujących wierszy:

```
$ftpmail_address = "ftpmail@decwrl.dec.com";
$ftpmail_location = "FTP.$whereami";
```

W zmiennej `$ftpmail_address` należy zamiast `$whereami` podać adres witryny FTP położonej blisko naszego serwera. Listę działających w sieci serwerów FTP-MAIL można znaleźć w witrynie WWW: *http://src.doc.ic.ac.uk/ftpmail-servers.html*.

Codzienna eksploatacja serwera i rozwiązywanie problemów

Majordomo nie wymaga od administratora zbyt wiele pracy. Większość obowiązków spada na właścicieli list. Niemniej jest kilka spraw, o których administrator serwera powinien wiedzieć.

Lista bounces

Majordomo rozwiązuje problem odbijanych wiadomości w dość nietypowy sposób. Nie usuwa automatycznie użytkowników, których adresy odbijają wysyłaną do nich pocztę. Gdy natomiast

właściciel listy usunie adres odbijający pocztę, adres ten może zostać umieszczony na specjalnej liście dystrybucyjnej *bounces*. Program *bounce-remind*, zazwyczaj uruchamiany w godzinach nocnych, próbuje skontaktować się z usuniętymi użytkownikami i poinformować ich, w jaki sposób mogą się ponownie zapisać na swoją listę.

Aby tego rodzaju obsługa problemu odbijanej poczty była w ogóle możliwa, konieczne jest utworzenie listy *bounces*. W celu jej uruchomienia:

1. Utwórz listę *bounces* w taki sam sposób, jak zwykłą listę dystrybucyjną.
2. W pliku *bounces.config*:
   - Przypisz dyrektywie `precedence` wartość `bulk`, aby zapobiec zwracaniu poczty wysyłanej przez tę listę.
   - Przypisz dyrektywom `reply_to` i `sender` alias *nobody*, aby wszystkie odpowiedzi mogły być po cichu ignorowane.
   - Przypisz dyrektywie `strip` wartość `no`. Komentarze na liście subskrybentów są wykorzystywane przez program *bounce-remind* do ustalania, z której listy subskrybent został usunięty.
   - Przypisz dyrektywie `noadwertise` szablon `/./` (prawy ukośnik, kropka, prawy ukośnik). To wyrażenie regularne odpowiada wszystkim możliwym adresom, tak więc lista nie będzie się pojawiać w odpowiedzi na polecenia `lists` wysyłane przez użytkowników.
3. Sprawdź skrypt *bounce-remind*, aby upewnić się, że podana w pierwszym wierszu ścieżka do Perla jest poprawna.
4. Utwórz nową pozycję w pliku *crontab*, która uruchamiać będzie skrypt *bounce-remind* w godzinach nocnych. Skrypt *bounce-remind* poszukuje ścieżki do konfiguracyjnego pliku *majordomo.cf* w zmiennej środowiskowej `MAJORDOMO_CF`. Jeśli jednak zadanie *cron* nie ustawia tej zmiennej środowiskowej, to nowa pozycja w pliku *crontab* powinna być następująca:

```
30  3  *  *  *  MAJORDOMO_CF=/home/majordomo/runtime/majordomo.cf
                 /home/majordomo/runtime/wrapper bounce-remind
```

5. Roześlij wszystkim właścicielom list kopie skryptu *bounce* i hasło listy *bounces*, aby mogli dodać je do swoich plików *.majordomo*. Skrypt *bounce* i plik *.majordomo* zostały omówione w rozdziale 4.

Gdy program *bounce-remind* zostanie już w ten sposób zainstalowany, będzie wysyłał na każdy z adresów znajdujących się na liście *bounces* wiadomość podobną do tej:

```
To: Bounces@kapelusze.com
From: nobody@kapelusze.com
Subject: Bouncing email from mailing lists at kapelusze.com
Reply-To: nobody@kapelusze.com

Your address has been moved to Bounces@kapelusze.com
from some other mailing list at kapelusze.com
because email to you was bouncing.

Here are the addresses currently on Bounces@kapelusze.com
so that you can see which of your addresses is among them.
The comment for each address shows the date it was moved,
and the first list it was removed from.  If you were on
multiple lists here, you may have been removed from them
```

```
as well, but only the first list you were removed from
will show up in the comment below.

  użytkownik@jakiśtam.edu (960610 kapelusznicy)
   ...

If the problem has been fixed, you can get off of
Bounces and back on to the other list by sending the
following to majordomo@kapelusze.com:

    subscribe your_list
    unsubscribe bounces

To subscribe or unsubscribe an address other than where you're
sending the command from, append the other address to the end
of the "subscribe" or "unsubscribe" command (for example,
"subscribe your_list foo@bar.com").

You'll need to access the mailing list archives if you want to catch
up on whatever you missed while you were off the main list.

If you don't want to keep getting these reminders every day, but
don't want to resubscribe to the list, send just the "unsubscribe"
command shown above.

If you need to contact a human being regarding this, send a message
to owner-majordomo@kapelusze.com
```

Jeśli wiadomość ta dotrze do subskrybenta, będzie on wiedział, co się stało i co powinien zrobić, aby powrócić na swoją listę. Albo administrator, albo właściciele list powinni co jakiś czas sprawdzać plik subskrybentów listy *bounces* (wysyłając polecenie `who bounces`) i usuwać subskrybentów, którzy znajdują się na tej liście na przykład od miesiąca lub dłużej.

Tymczasowe wyłączanie serwera Majordomo

Programy tworzące serwer Majordomo są przywoływane zawsze wtedy, gdy na któryś z zarządzanych przez Majordomo adresów nadchodzi jakaś poczta. Komplikuje to obsługę plików serwera Majordomo. Jeśli popełnimy błąd, nadchodząca poczta może zostać zwrócona nadawcy z informacją, że nie można jej dostarczyć. Większości problemów można jednak uniknąć dzięki edytowaniu plików Majordomo skryptu *medit*. Czasami jednak zachodzi potrzeba tymczasowego wyłączenia serwera Majordomo. Poniżej znajduje się instrukcja, jak to zrobić:

1. Utwórz w katalogu *runtime* skrypt shella *majordomo.hold*:

   ```
   #!/bin/sh
   exit 75
   ```

2. Uczyń ten skrypt plikiem wykonywalnym:

   ```
   % chmod 755 majordomo.hold
   ```

3. Gdy już jesteś gotowy, by wyłączyć Majordomo, szybko zamień ze sobą skrypty *majordomo* i *majordomo.hold*:

   ```
   % cp majordomo majordomo.tmp && cp majordomo.hold majordomo
   ```

   (Operator `&&` shella pozwala na jednoczesne wpisanie dwu poleceń bez konieczności wykonywania najpierw jednej z nich. W tym przypadku drugie polecenie kopiowania *cp* zostanie wykonane tylko wtedy, gdy pierwsze zakończy się powodzeniem). Teraz gdy

do *majordomo* nadejdzie jakaś poczta, zostanie uruchomiony prosty skrypt przygotowany przed chwilą, który po uruchomieniu zwróci kod zamknięcia 75, co spowoduje, że program *sendmail* przeniesie wiadomość do kolejki pocztowej i podejmie próbę dostarczenia jej później. (Specjalne kody zamknięcia, takie jak kod 75, zostały opisane w systemowym pliku */usr/include/sysexits.h*)

4. Gdy już wszelkie niezbędne „prace konserwacyjne" zostaną zakończone, można będzie uruchomić serwer, przywracając na właściwe miejsce prawdziwy skrypt *majordomo*:

```
% cp majordomo.tmp majordomo
```

W podobny sposób, zamieniając *resend* z *majordomo.hold*, można przejściowo wyłączyć wszystkie adresy list dystrybucyjnych.

Warto również pamiętać, że program *sendmail* może przekonwertować ścieżki do poleceń uruchamianych przez zatrzymane w kolejce wiadomości w ten sposób, że będą zawierać tylko małe litery. Na przykład jeśli alias archiwum będzie miał taką postać:

```
kapelusznicy-archive: "|/usr/local/majordomo/wrapper archive
  -f /usr/local/majordomo/Archives/kapelusznicy/kapelusznicy -m -a"
```

Wielka litera `A` w podanej ścieżce może zostać zamieniona na małe `a`, gdy wiadomość zostanie zatrzymana w kolejce i po ponownym wysłaniu wiadomości skrypt *archive* nie zdoła odnaleźć katalogu archiwum. Rozwiązaniem tego problemu jest albo dokonanie odpowiednich zmian w pliku konfiguracyjnym *sendmail.cf* programu *sendmail* (aby program nie zmieniał wielkości liter), albo dodanie powiązania symbolicznego do katalogu wykorzystującego w nazwie małe litery (w tym przypadku powiązanie *archives* z *Archives*), albo używanie w nazwach plików i katalogów tylko małych liter.