Narzędzia

nroff, troff – formatowanie dokumentacji w systemach uniksowych*

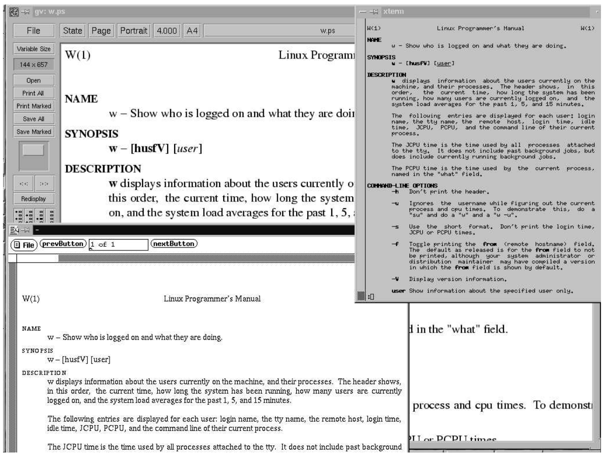
Piotr Bolek

We wszystkich systemach Unix „od zawsze” były dostępne wyrafinowane narzędzia do formatowania tekstu. Tradycyjnie dokumentacja do systemu dostępna jest „on-line” w postaci tzw. stron podręcznika ekranowego (*man pages*). Dokumentacja ta może być oglądana na zwykłych terminalach tekstowych za pomocą standardowego polecenia `man`. Strony podręcznika nie są zwykle dostępne wyłącznie w gotowej do obejrzenia postaci sformatowanej. Najczęściej są formatowane na żądanie w momencie wywołania polecenia `man`. Do formatowania dokumentacji jest używany program `nroff` – czyta on tekst źródłowy strony podręcznika i formatuje go w sposób dostosowany do możliwości tekstowych terminali lub drukarek wierszowych. Możliwe jest także sformatowanie tego samego dokumentu do druku na drukarkach laserowych. W tym celu należy użyć programu `troff`, który czytając te same pliki źródłowe przygotowuje tekst do prezentacji na urządzeniu o lepszych możliwościach.

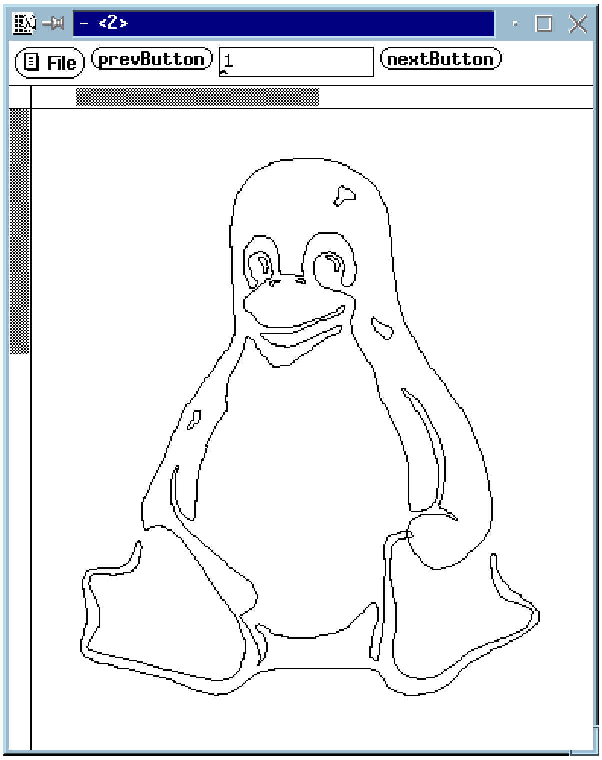
Uniksowe programy do formatowania

Sposób pracy programów służących do formatowania dokumentacji uniksowej może wydawać się niezwykły dla użytkowników edytorów WYSYWIG i systemów składu. Rodzina programów nroff/troff (do której zalicza się także bezpłatna wersja GNU – groff) pracuje bowiem w trybie wsadowym. Oznacza to, że w pliku źródłowym są „wymieszane” polecenia składu i tekst, stanowiący dane, na których działa program. Co jeszcze bardziej niezwykłe – efekt składu zależy nie tylko od ustawionych parametrów (wymiary strony, stopień pisma itd), ale także od urządzenia wyjściowego, na którym ma być prezentowana ostateczna sformatowana wersja pracy. Efekt prezentacji tej samej strony podręcznika w różny sposób można zobaczyć na rysunku 1.

Język używany do definiowania poleceń sterujących jest bardzo bogaty i pozwala m.in. na definiowanie makrodefinicji, zakładanie pułapek (czyli definiowanie akcji, które będą wykonane w chwili osiągnięcia pewnego stanu procesu składania tekstu). Razem z podstawowymi programami dostępne są także preprocesory pozwalające w łatwy i wygodny sposób definiować tabele (`tbl`), formuły matematyczne (`eqn`), a nawet grafikę (`pic`, rys. 2).



Rys. 1. Różne sposoby prezentacji strony podręcznika



Rys. 2. Możliwości graficzne programu troff

Oczywiście wzory i grafika nie mogą być poprawnie wyświetlone na znakowym terminalu, ale

∗: Przedruk z *Magazynu Unix & Linux*, nr 9/1999.

tabele zwykle tak. Możliwości jakie dają preprocesory są naprawdę duże. Osobiście najbardziej spodobał mi się sposób definiowania tabel – i fakt, że jak raz przygotujemy tabelę w postaci źródłowej, to możemy bez konieczności jakichkolwiek modyfikacji pliku źródłowego otrzymać tabelę sformatowaną w PostScripcie i w ASCII (rys. 3). Kodu źródłowego tabeli nie podaję, ponieważ byłby on niezrozumiały dla większości czytelników, a omówienie języka używanego do kodowania tabel to materiał na oddzielny artykuł.

```
+----------+-------------------------------------+----------+
|          |               Poland                | European |
|Sources   |                                     | Union    |
|of energy +-------+-------+-------+-------+-------+----------+
|          | 1960  | 1970  | 1980  | 1990  | 1994  |   1994   |
+----------+-------+-------+-------+-------+-------+----------+
| Coal     | 95,3% | 83,6% | 78,5% | 77,3% | 73,7% |   19,5%  |
+----------+-------+-------+-------+-------+-------+----------+
| Oil      |  3,4% | 10,3% | 14,6% | 13,7% | 16,2% |   44,1%  |
+----------+-------+-------+-------+-------+-------+----------+
| Gas      |  1,2% |  5,9% |  6,7% |  8,8% |  8,5% |    20,3% |
+----------+-------+-------+-------+-------+-------+----------+
| Other*   |  0,1% |  0,2% |  0,2% |  0,2% |  1,6% |    16,2% |
+----------+-------+-------+-------+-------+-------+----------+
```

Sources of energy	Poland					European Union
	1960	1970	1980	1990	1994	1994
Coal	95,3%	83,6%	78,5%	77,3%	73,7%	19,5%
Oil	3,4%	10,3%	14,6%	13,7%	16,2%	44,1%
Gas	1,2%	5,9%	6,7%	8,8%	8,5%	20,3%
Other*	0,1%	0,2%	0,2%	0,2%	1,6%	16,2%

Rys. 3. Dwa sposoby prezentacji tej samej tabeli

Polecenia sterujące składem są dosyć niskiego poziomu: np. „wycentruj pięć kolejnych wierszy pliku wejściowego”, „zmień szerokość marginesów”, „pozostaw 5 pustych wierszy” itp.. We wszystkich systemach uniksowych dostępne są jednak także liczne pakiety makr wysokiego poziomu (me, mm, ms, man) pozwalające znakować tekst w sposób mniej lub bardziej strukturalny. Najczęściej używanym pakietem makr jest pakiet man służący do znakowania dokumentacji – uniksowych stron podręcznika ekranowego.

Struktura stron podręcznika

Strony podręcznika winny być przygotowane w odpowiedni sposób. Wszystkie polecenia tradycyjnie zaczynają się od kropki i muszą być umieszczone na początku wiersza Każda strona składa się z tytułu oraz kolejnych sekcji. Pierwszym poleceniem (poza ewentualnymi komentarzami) powinna być definicja tytułu strony:

```
|.TH| <tytuł> <rozdział> <data> \
                 <źródło> <podręcznik>
```

gdzie:

- <tytuł> – tytuł strony – najczęściej nazwa polecenia, funkcji systemowej, czy pliku konfiguracyjnego, pisana wielkimi literami (np. MAN, PERL, SYSLOG.CONF)
- <rozdział> – jest numerem rozdziału, w którym jest umieszczona strona. Tradycyjny podział na rozdziały jest następujący:

Nr	Tytuł rozdziału	Opis
1	Commands	Polecenia, które mogą być używane przez użytkowników z poziomu shella
2	System calls	Funkcje systemowe udostępniane przez jądro systemu (np. semctl, fork)
3	Library call	Funkcje biblioteczne udostępniane przez bibliotekę języka C (np. qsort, printf) lub inne biblioteki dostępne w systemie (np. biblioteka wątków)
4	Special files	Opis plików specjalnych dostępnych w katalogu /dev
5	File formats	Opis różnych plików konfiguracyjnych systemu Unix (np. passwd, wtmp)
6	Games	Gry
7	Macro packages and convenctions	Opis pakietów makr (np. man) oraz różnych konwencji systemowych

W niektórych systemach (np. Solaris) podział ten jest inny, być może bardziej szczegółowy.

- <data> – data ostatniej modyfikacji, należy ją aktualizować po każdej zmianie treści strony.
- <źródło> – pochodzenie polecenia lub funkcji systemowej, np. GNU, NET-2, Linux 0.99.13 (wersja jądra systemu Linux).
- <podręcznik> – tytuł podręcznika np. „Tk Library Procedures”

Nie wszyscy twórcy programów i autorzy dokumentacji stosują się ściśle do konwencji i czasami używają polecenia .TH nieco inaczej. Oto kilka przykładów wziętych z systemu Linux.

```
.TH KILL 1 "14 October 1994" "Linux \
    Utilities" "Linux Programmer's  \
    Manual"
.TH PS 1 "3 Sep 1997" "Cohesive     \
    Systems" "Linux Programmer's    \
    Manual"
.TH LS 1 "GNU File Utilities" "FSF"
.TH cpp 1 "30apr1993" "GNU Tools"   \
    "GNU Tools"
.TH GAWK 1 "Dec 19 1996" "Free      \
    Software Foundation" "Utility   \
    Commands"
```

```
.TH XTERM 1 "Release 6.4" "X Version\
   11"
.TH UMASK 2 "24 July 93" "Linux"    \
   "Linux Programmer's Manual"
.TH Tk_WindowId 3 "" Tk "Tk Library \
   Procedures"
.TH E2FSCK 8 "April 1997" "E2fsprogs\
   version 1.10"
.TH POSTGRESQL UNIX 12/08/96        \
   PostgreSQL PostgreSQL
```

Jak widać nie wszyscy umieszczają datę, nie zawsze też jest podana nazwa podręcznika. W praktyce podanie w argumentach polecenia `.TH` informacji niepełnych lub niezgodnych z konwencją nie powoduje fatalnych skutków – co najwyżej niektóre pola w nagłówkach lub stopkach strony podręcznika będą puste.

Kolejne części opisu polecenia na stronie podręcznika definiowane są jako sekcje za pomocą polecenia `.SH` <nazwa>. Do tradycyjnych pojawiających się w większości stron nagłówków sekcji należą:

- `NAME` – jedyna wymagana sekcja. Powinna zaczynać się od jednowierszowego opisu polecenia lub funkcji. Wiersz ten jest używany przez program makewhatis do konstruowania bazy danych wykorzystywanej przez polecenia apropos i whatis. Musi on zawierać nazwę i krótki opis rozdzielone parą znaków `\-` (ukośnik i minus), np.:

```
.SH NAME
gcc, g++ \- GNU project C and C++ \
Compiler (v2.7)

.SH NAME
gawk \- pattern scanning and      \
processing language

.SH NAME
umask \- set file creation mask

.SH NAME
xterm \- terminal emulator for X
```

- `SYNOPSIS` – składnia polecenia (funkcji) i wszystkie dopuszczalne argumenty. Przy opisie składni korzysta się z następujących konwencji:
  - nazwy poleceń, argumentów i opcji są pogrubione,
  - argumenty, których wartość może podać użytkownik oznacza się kursywą,
  - argumenty opcjonalne są umieszczane w nawiasach kwadratowych (`[]`)
  - do rozdzielenia argumentów, które mogą być używane zamiennie używa się pionowej kreski (|),
  - argument, po którym umieszczony jest znak wielokropka (...) może być powtarzany,
  - w przypadku funkcji bibliotecznej, umieszcza się tutaj deklarację funkcji, poprzedzoną w razie konieczności niezbędnymi deklaracjami danych oraz dyrektywami `#include`.
- `DESCRIPTION` – opis działania polecenia lub funkcji. Nie podaje się tutaj zwykle szczegółowego opisu implementacji, a jedynie opis funkcjonalny, z punktu widzenia użytkownika. Odnośniki do części opisującej składnię są formatowane w identyczny sposób jak w części `SYNOPSIS`.
- `OPTIONS` – opis wszystkich opcji i ich wpływu na działanie polecenia.
- `FILES` – lista plików, związanych z poleceniem lub funkcją biblioteczną.
- `SEE ALSO` – lista innych stron podręcznika związanych tematycznie z bieżącą stroną. Zgodnie z konwencją podaje się pogrubioną nazwę strony i numer rozdziału podręcznika. Tutaj podaje się także odnośniki do innego rodzaju dokumentacji.
- `DIAGNOSTICS` – lista komunikatów diagnostycznych z opisami.
- `BUGS` – opis ograniczeń, znanych błędów i problemów związanych z poleceniem lub funkcją.
- `AUTHORS` – historia programu, kontakt do autorów.

Często używane są także inne nazwy sekcji. Nie ma tu żadnych formalnych ograniczeń.

Zmiana kroju pisma

Polecenia służące do zmiany kroju pisma są następujące: `.B`, `.I` zmieniają krój pisma odpowiednio na pogrubiony i kursywę. Zmiana kroju dotyczy tekstu podanego jako argument polecenia tzn. do końca wiersza wejściowego. Przy prezentacji dokumentu na terminalach znakowych, gdzie nie ma możliwości zmiany kroju pisma, pogrubienie jest zastąpione podświetleniem znaku, a kursywa podkreśleniem.

Pozostałe polecenia do zmiany kroju pisma na stronach podręcznika są nieco bardzie skomplikowane. Sklejają one słowa podane jako argumenty dokonując jednocześnie zmiany kroju pisma. Polecenia te to `.BI`, `.IB`, `.BR`, `.RB`. Powodują one, że kolejne argumenty (tradycyjnie do 6, w wersji groff bez ograniczeń), są na przemian odpowiednio: pogrubione-pochylone, pochylone-pogrubione, pogrubione-normalne, normalne-pogrubione. Najłatwiej zrozumieć działanie tych poleceń na przykładach, następująca sekwencja wejściowa:

```
.BI jeden dwa trzy cztery
.IB JEDEN DWA TRZY CZTERY
.BR jeden dwa trzy cztery
.RB JEDEN DWA TRZY CZTERY
```

będzie po przetworzeniu do przez troffa wyglądała następująco:

jedendwatrzycztery JEDENDWATRZYCZTERY jedendwatrzycztery JEDENDWATRZYCZTERY

Zastosowanie takich makr jest być może nieoczywiste, ale wystarczy zajrzeć do źródeł stron manuala w swoim systemie, aby zrozumieć do czego mogą się one przydać.

Polecenia formatujące

Pozostałe dostępne polecenia pozwalają wpływać na sposób formatowania strony podręcznika. Są one bardzo proste i jest ich stosunkowo niewiele:

- `.LP`, `.PP`, `.P` – rozpoczęcie nowego akapitu – w wierszu gdzie pojawi się jedno z tych poleceń nie może być żadnego tekstu.
- `.HP` – rozpoczęcie „wiszącego” bloku tekstu – wszystkie wiersze są wcięte z lewej strony względem pierwszego wiersza bloku tekstu. Opcjonalny argument określa wielkość wcięcia.
- `.RS` – początek bloku wciętego z lewej strony. Opcjonalny argument określa wielkość wcięcia.
- `.RE` – koniec wcięcia z lewej strony.
- `.IP` – akapit z „wiszącą” etykietą, etykieta podana w tym samym wierszu co polecenia. Drugi opcjonalny argument określa wielkość wcięcia.
- `.TP` – jak wyżej, ale etykieta w następnym wierszu. Opcjonalny argument w tym samym wierszu co polecenie pozwala określić wielkość wcięcia. To i poprzednie polecenie są używane przy opisach opcji.
- `.SS` – drugi poziom nagłówków sekcji.

Działanie powyższych poleceń najlepiej można zobaczyć na małym przykładzie. Dla skrócenia zapisu źródła na początku jest zdefiniowane makro `.tekst`, które używane jest później zamiast jawnego wpisywania tekstu do wypełnienia.

```
.de tekst
Tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst.
..
.TH TEST 1 1999-06-29 "Magazyn Linux \
& Unix" "Test Manual"
.nh
.SH NAME
test \- testowanie makr podręcznikowych
.SH TESTY
.PP
Zwykły akapit.
.tekst
.tekst
.HP
Akapit z zawieszeniem.
.tekst
.IP ***>
Krótka wisząca etykieta
.tekst
.IP "Długa wisząca etykieta"
.tekst
.TP
Wisząca etykieta definiowana w oddzielnym \
wierszu.
.tekst
.PP
.RS
Wcięty akapit.
.tekst
.PP
Drugi wcięty akapit.
.tekst
.RE
.PP
Koniec wcięcia
```

Efekt przetworzenia na terminalu tekstowym będzie następujący:

```
TEST(1)                 Test Manual                 TEST(1)


NAME
       test - testowanie makr podręcznikowych

TESTY
       Zwykły  akapit.  Tekst tekst tekst tekst tekst tekst tekst
       tekst tekst tekst tekst tekst  tekst  tekst  tekst  tekst.
       Tekst  tekst  tekst  tekst  tekst  tekst tekst tekst tekst
       tekst tekst tekst tekst tekst tekst tekst.

       Akapit z zawieszeniem.  Tekst  tekst  tekst  tekst  tekst
              tekst  tekst  tekst  tekst  tekst tekst tekst tekst
              tekst tekst tekst.

       ***>   Krótka wisząca etykieta  Tekst  tekst  tekst  tekst
              tekst  tekst  tekst  tekst  tekst tekst tekst tekst
              tekst tekst tekst tekst.

       Długa wisząca etykieta
              Tekst tekst tekst tekst  tekst  tekst  tekst  tekst
              tekst tekst tekst tekst tekst tekst tekst tekst.

       Wisząca etykieta definiowana w oddzielnym wierszu.
              Tekst  tekst  tekst  tekst  tekst tekst tekst tekst
              tekst tekst tekst tekst tekst tekst tekst tekst.

              Wcięty akapit.  Tekst tekst tekst tekst tekst tekst
              tekst  tekst  tekst  tekst  tekst tekst tekst tekst
              tekst tekst.

              Drugi wcięty akapit.  Tekst tekst tekst tekst tekst
              tekst  tekst  tekst  tekst  tekst tekst tekst tekst
              tekst tekst tekst.

       Koniec wcięcia
Magazyn Linux & Unix      1999-06-29                          1
```

Wszystkich, którzy chcą się więcej dowiedzieć na temat możliwości formatowania tekstów z wykorzystaniem narzędzi uniksowych muszę odesłać do innych źródeł. Zestawienie wszystkich poleceń programów `nroff` i `troff` można znaleźć między innymi w książce „UNIX in a Nutshell” wydawnictwa O’Reilly. Książka była przetłumaczona na język polski pod tytułem „UNIX podręcznik użytkownika”.

◇ Piotr Bolek
PBolek@GUST.org.pl