

Wielofunkcyjne narzędzie dla plików PDF

WIELKA MOC PDF

Aby okiełznać stosy papierów piętrzące się każdego dnia na naszych biurkach potrzebujemy katalogować, otrzymywać, kopiować, znakować, przeszukiwać i klasyfikować dokumenty. Istnieje narzędzie pozwalające wygrzebać się spod stosu papierzynek: pdftk – zbiór narzędzi do obróbki plików PDF.

BY STEFAN LAGOTZKI

Natywne narzędzia do pracy z plikami PDF dostępne dla Linuksa, jak GhostScript i Xpdf są bardzo użyteczne, jeśli ktoś chce pracować z interfejsami graficznymi i przetwarzać jednocześnie pojedyncze pliki. Jeśli jednak ktoś szuka szybszego sposobu, lub gdy chce zautomatyzować powtarzalne zadania, powinien spróbować pdftk. To wygodny i skuteczny program obsługiwany z wiersza poleceń do (wsadowego) przetwarzania plików PDF. Cytując Sida Stewarda, twórcę pdftk: „Jeśli PDF jest elektronicznym papierem, pdftk jest elektronicznym rozszyfrowaczem, dziurkaczem, bin-downicą, maszyną odszyfrowującą i okularami do odczytu tajnych danych”.

Instalacja i wykorzystanie

Najnowszą wersję pdftk można pobrać z jednej ze stron WWW Sida Stewarda [1]. Program jest dostępny na licencji GPL na platformy Linux, Max OS X (Panther), FreeBSD, Solaris i Windows. Pakiety instalacyjne specyficzne dla danej platformy okazały się zupełnie wystarczające na naszych platformach testowych (Debian i SuSe Linux).

Po pomyślnym zakończeniu instalacji można uruchomić pdftk z powłoki. Polecenie `pdftk -help` wypisze listę poleceń i opcji obsługiwanych przez program wraz z ich krótkim opisem. Podstawowa składnia programu do przetwarzania plików PDF jest następująca:

```
pdftk plik (i)_wejściowy (e) >
operacja [opcja] >
output plik_wyjściowy >
[hasła] [uprawnienia]
```

Pliki wejściowe muszą być w formacie PDF. Do niektórych operacji program wykorzystuje pliki tekstowe w określonym formacie. W wyniku działania programu powstają pliki PDF (jeden lub więcej), w niektórych przypadkach również pliki tekstowe.

Przygotowałem kilka przykładów, które zademonstrują najciekawsze zastosowania programu pdftk. Przykłady te w żadnym wypadku nie wyczerpują możliwości tego narzędzia.

Załączniki w dokumentach PDF

Do dokumentów PDF można dołączać pliki tak samo, jak to się robi z e-mailami.

Adobe Reader (w wersji 6. lub nowszej) zapisuje załączniki w strukturze dokumentu PDF. Dzięki pdftk można dołączać pliki do dokumentów PDF, można je też z nich wydobywać. Przed pojawieniem się Adobe Readera 7 był to jedyny sposób wydobywania załączników z dokumentów PDF dostępny dla użytkowników Linuksa.

Załączniki można wykorzystać do dołączania do dokumentów PDF kodów źródłowych lub fragmentów z literatury. Poniższy przykład demonstruje sposób dołączenia kodu źródłowego do dokumentu PDF. W tym celu wywołuje się następujące polecenie:

```
pdftk form.pdf attach_files >
form.tex output new.pdf
```

Aby dokonać tego samego, można wykorzystać pdfLaTEX i polecenie `attachfile`. Odbiorca takiego dokumentu może wyko-

Tabela 1: Operacje obsługiwane przez pdftk

Kod operacji	Działanie
<code>attach_files</code>	Dodaje pliki jako załączniki do dokumentu PDF. Dzięki temu można do pliku PDF dodać plik archiwum.
<code>background</code>	Dodaje znak wodny na każdej stronie dokumentu PDF. Pozwala również wstawić logo firmy itp.
<code>burst</code>	Dzieli dokument PDF na pojedyncze strony.
<code>cat</code>	Tworzy nowy dokument PDF z wielu plików lub stron pochodzących z innych dokumentów PDF.
<code>dump_data</code>	Wypisuje na standardowym wyjściu informacje o dokumencie PDF.
<code>dump_data_fields</code>	Wypisuje na standardowym wyjściu informacje o polach formularza dokumentu PDF.
<code>fill_form</code>	Wypełnia formularze PDF lub łączy dane formularza z dokumentem.
<code>unpack_files</code>	Wypakowuje załączniki z dokumentu PDF.
<code>update_info</code>	Aktualizuje metainformacje dokumentu PDF (np. autor, tytuł, temat).

rzystać pdftk do wypakowania załączników z dokumentu i zapisać je we wskazanym katalogu:

```
pdftk example_attachment.pdf >
unpack_files output Source
```

W tym przykładzie pdftk zapisuje załączniki w katalogu *Source*. Wykorzystanie katalogu ma sens szczególnie wtedy, gdy mamy do czynienia z dużą liczbą załączników.

Znaki wodne i kolory tła

Program pdftk do dodawania tzw. znaków wodnych w dokumencie wykorzystuje podobne podejście do pakietu *eso-pic* LaTeX-a. Służy do tego opcja *background*, którą można również zastosować do ustawienia tła dokumentu PDF.

Obraz ustawiany jako znak wodny musi być dokumentem PDF. Można utworzyć go z grafiki wektorowej lub napisać samodzielnie program w języku PostScript. Jeśli znak wodny jest innego rozmiaru niż dokument, pdftk przeskaluje go odpowiednio. Załóżmy, że chcemy wstawić napis DRAFT na każdej stronie dokumentu. W pierwszym kroku należy utworzyć odpowiedni dokument PDF ze znakiem wodnym o rozmiarze strony zgodnym z rozmiarem dokumentu, do którego będzie wstawiany, a następnie wywołać pdftk:

```
pdftk example.pdf background >
draft.pdf output draft1.pdf
```

Znak wodny wygląda jak wzór na papierze, który widać w miejscach nieprzykrytych przez obiekty dokumentu. Można również utworzyć na przykład siatkę, zapisując odpowiedni wzór w pliku EPS. Polecenia w języku PostScript realizujące to zadanie dla kartki papieru w formacie A4 są następujące:

```
%!PS-Adobe-2.0
%BoundingBox: 0 0 595 842
0.95 0.95 0.90 setrgbcolor
0 0 moveto 595 0 rlineto 0 842 >
rlineto -595 0 rlineto
closepath fill
showpage
```

Łatwo też zmienić kolor tła w kodzie EPS. Następnie ten dokument można przekonwertować do formatu PDF wykorzystując *epstopdf* i uruchomić pdftk ustawiając ten plik jako tło:

```
pdftk example.pdf background >
Bg.pdf output eg_color.pdf
```

Dzielenie i łączenie plików PDF

Operacja *burst* pozwala podzielić plik PDF na strony składowe. W tym celu należy podać programowi rdzeń nazwy plików poszczególnych stron oraz określić format numeracji:

```
pdftk example.pdf burst >
output Page%03d.pdf
pdftk example.pdf burst >
output ./Pages/Page%03d.pdf
```

W obydwu przykładach do nazwy strony zostanie dodany jej trzycyfrowy numer. W drugim przykładzie pdftk zapisze pliki stron w osobnym podkatalogu.

Operacja *cat* służy do łączenia kilku plików PDF, tworząc nowy dokument. Do określenia wielu plików można zastosować wzorce dopasowania.

```
pdftk example.pdf form.pdf >
attachment.pdf >
cat output example_concat.pdf
pdftk D=coversheet.pdf >
B=example.pdf >
cat D B1-4 output >
example_coversheet.pdf
```

Jak widać w drugim przykładzie, operacja *cat* może posłużyć również do zamiany kolejności fragmentów dokumentu i na przykład połączyć fragment jednego dokumentu z fragmentem innego, tworząc w ten sposób nowy dokument.

Odczyt i aktualizacja metainformacji

Większość plików PDF zawiera metainformacje dotyczące szczegółów dokumentu, jak autor, temat, czy oprogramowanie wykorzystane do wygenerowania dokumentu. Program pdftk potrafi wypisać te informacje na standardowym wyjściu lub zapisać je w pliku:

```
pdftk example.pdf >
dump_data output info.txt
```

Powyższe wywołanie zapisuje metainformacje z dokumentu PDF do pliku tekstowego *info.txt*. Informacje składają się z pola nazwy klucza oraz jego wartości (Listing 1.). Przed przesłaniem lub zarchiwizowaniem pliku PDF warto zaktualizować jego metainformacje. Dzięki pdftk można

tego dokonać bez konieczności ponownego wygenerowania dokumentu PDF.

Aby zmodyfikować metainformacje dokumentu PDF należy stworzyć plik tekstowy definiujący te metadane. Plik powinien mieć następującą postać (nieco skrócona):

```
InfoKey: Creator
InfoValue: TeX
InfoKey: Corporation
InfoValue: Sample and Sons
```

W pliku wejściowym nie ma potrzeby zapisywania wszystkich metainformacji, które mają znaleźć się w dokumencie PDF. Wszystkie pola niezdefiniowane w pliku tekstowym zachowają oryginalną wartość w dokumencie PDF. Można również dodawać niestandardowe klucze (w naszym przykładzie jest to Corporation) i nadawać im wartości. Poniższe polecenie zmodyfikuje metainformacje w dokumencie PDF, wykorzystując definicję z pliku tekstowego:

```
pdftk example.pdf >
update_info info.txt >
output eg_meta.pdf
```

Wejściowy i wyjściowy dokument PDF nie mogą mieć tej samej nazwy. Innymi słowy, aby plik wynikowy miał tę samą nazwę co plik wejściowy, należy zmienić ją ręcznie lub wykorzystać odpowiedni skrypt.

Wypełnianie formularzy PDF

Dokumenty PDF mogą zawierać formularze. Firma Adobe opracowała własny otwarty format FDF na potrzeby danych

Listing 1: Typical PDF Meta-Data

```
InfoKey: Title
InfoValue: Praca z pdftk
InfoKey: Subject
InfoValue: Praktyczne przykłady zastosowań pdftk
InfoKey: Keywords
InfoValue: pdftk, iText, Open Source applications
InfoKey: Author
InfoValue: Stefan Lagotzki
InfoKey: City
InfoValue: Dresden
```

Listing 2: Przykład pliku w formacie FDF

```
%PDF-1.2
1 0 obj <<
  /FDF << /Fields [
    << /V (Dresden)/T (city) >>
    << /V (Stefan Lagotzki)/T
      (author)>>
  ]/F (form.pdf) >>
>>
endobj
trailer
<<
  /Root 1 0 R
>>
%%EOF
```

formularza dokumentów PDF. Listing 2 przedstawia przykład krótkiego pliku FDF.

Na Listingu 2 *T* określa tytuł, *V*to wartość pola formularza. Po zdefiniowaniu wartości formularza można wypełnić je w dokumencie PDF. Można również zdecydować, które dane w formularzu mają być modyfikowalne, a które trwale połączone z dokumentem:

```
pdftk form.pdf fill_form >
eg.fdf output edit.pdf
pdftk form.pdf fill_form >
eg.fdf output end.pdf flatten
```

Pierwszy przykład powoduje wypełnienie formularza, pozostawiając go w trybie edytowalnym. Drugi przykład, dzięki opcji *flatten* powoduje, że wartości formularza są wypełniane i łączone z dokumentem PDF w sposób trwały.

Opcja wypełniania formularzy PDF pozwala wykorzystać formularze PDF na stronie WWW. Użytkownik wypełnia formularz za pomocą przeglądarki WWW na

zwykłej stronie HTML. Następnie skrypt PHP lub Perla wypełni dokument PDF odpowiednimi wartościami za pomocą *pdftk*. Następnie tak wypełniony dokument PDF może być wysłany do użytkownika e-mailem.

Hasła i uprawnienia do dokumentu PDF

Dokumenty PDF można zabezpieczać za pomocą hasła użytkownika lub właściciela. Program *pdftk* daje możliwość zdefiniowania każdego z tych haseł jak również uprawnień do dokumentu PDF. Poniższy przykład ustanawia obydwa hasła:

```
pdftk file.pdf output >
file_new.pdf owner_pw >
Lie5quai user_pw phupaefu
```

Hasła z tego przykładu są wygenerowane za pomocą programu *pwgen*. W rzeczywistości należy oczywiście zastosować własne ciągi znaków.

Właściciel dokumentu PDF może określać uprawnienia do pliku. Tabela 2 zawiera listę uprawnień, które można ustawić za pomocą *pdftk*. Poniższy przykład tworzy dokument PDF, który ma ustawione uprawnienia wyłącznie do druku. Drugi wiersz tworzy dokument PDF, który można drukować oraz kopiować do schowka jego zawartość.

```
pdftk example.pdf output >
file_new.pdf owner_pw >
Lie5quai user_pw phupaefu >
allow printing
pdftk example.pdf output >
file_new.pdf owner_pw >
Lie5quai user_pw phupaefu >
allow printing CopyContents
```

Dokumenty PDF mogą być szyfrowane na kilku poziomach kryptograficznych.

Aby zaszyfrować dokument za pomocą *pdftk*, można zastosować jedną z następujących opcji: *encrypt_40bit* lub *encrypt_128bit*. Trzeba również określić hasło. Przetwarzając kilka plików opcje kryptograficzne i hasła można określać dla każdego indywidualnie. W poniższym przykładzie tylko plik A będzie zabezpieczony hasłem.

```
pdftk A=file_new.pdf >
B=eg_color.pdf input_pw >
A=Lie5quai cat output >
egl_pw.pdf user_pw Abraxas
```

Powyższy przykład wykorzystuje plik *file_new.pdf*, któremu wcześniej odebraliśmy uprawnienia modyfikacji, aby wykonać tę operację musimy więc podać hasło jego właściciela.

Podsumowanie

Program *pdftk* jest użytecznym, wielofunkcyjnym narzędziem do przetwarzania plików PDF. Czytelnicy zainteresowani głębszymi tajnikami manipulacji dokumentami PDF powinni zapoznać się z książką *PDF Hacks* [2] autorstwa Sida Stewarda.

Program *pdftk* jest napisany w C++ i wykorzystuje bibliotekę *iText* [3], która z kolei jest napisana w Javie. Cały program został skompilowany i skonsolidowany za pomocą narzędzi wchodzących w skład GNU Compiler Collection [4], zatem jest łatwo przenośny i rozszerzalny. Strona domowa *pdftk* zawiera odnośniki do portów tego programu.

Praca nad programem nadal trwa. Jego autor Sid Steward na liście dyskusyjnej *comp.text.pdf* oraz na własnym forum dotyczącym formatu PDF [1] chętnie odpowiada na pytania dotyczące *pdftk* i programowania narzędzi do obróbki formatu PDF. ■

Tabela 2: Uprawnienia do dokumentów PDF

Opcja	Działanie
Printing	Dokument można drukować w pełnej jakości.
DegradedPrinting	Dokument można drukować w trybie obniżonej jakości.
ModifyContents	Można modyfikować zawartość dokumentu.
Assembly	Dokument PDF może być łączony z innymi dokumentami PDF.
CopyContents	Tekst i obrazy z dokumentu mogą być kopiowane za pośrednictwem schowka.
ModifyAnnotations	Można modyfikować i tworzyć komentarze.
FillIn	Można wypełniać formularze w dokumencie PDF.
AllFeatures	Użytkownicy mają wszystkie uprawnienia do dokumentu.

Dodatkowe informacje

[1] Sid Steward: *pdftk*; Version 1.12 (listopad 2004): <http://www.accesspdf.com/pdftk/>

[2] Sid Steward, *PDF Hacks*; O'Reilly, 2004.

[3] Bruno Lowagie, Paulo Soares: *iText-Library*; wersja 1.3.3 (sierpień 2005): <http://itext.sourceforge.net>

[4] GNU Compiler Collection: <http://gcc.gnu.org>