

Tworzenie dokumentów PDF przy pomocy L^AT_EX-a

Spis treści

1. Wstęp	2
2. Droga do celu	2
3. PDF z pomocą <code>dvipdfm</code>	3
3.1. Specjalne polecenia rozszerzające możliwości T _E X-a	3
3.2. <code>dvipdfm</code> a fonty	4
3.3. Wstawianie plików graficznych	5
4. Destylowanie plików postscriptowych	5
4.1. Adobe Acrobat	6
4.1.1. Ogólne parametry destylacji	6
4.1.2. Parametry wpływające na kompresję grafiki	6
4.1.3. Parametry fontów	9
4.1.4. Parametry kolorów	10
4.2. Ghostscript	10
4.3. PStill	10
5. PDF _T E _X oraz PDF _L A _T E _X	11
5.1. Plik konfiguracyjny dla PDF _T E _X -a	11
5.2. Polecenia specjalne PDF _T E _X -a	12
5.2.1. Polecenia wpływające na wygląd dokumentu	12
5.2.2. Polecenia ustalające informacje o dokumencie	13
5.2.3. Polecenia sterujące fontami	14
5.2.4. Polecenia wspomagające obiekty specjalne (XObject)	15
5.2.5. Polecenia włączające pliki graficzne	15
5.2.6. Polecenia wstawiające notatki	18
5.2.7. Polecenia tworzące hiperłącza	18
5.2.8. Polecenia wspomagające zakładki	19
5.2.9. Polecenia tworzące wątki	20
5.2.10. Pozostałe polecenia	20
5.3. Fonty i PDF _T E _X	22
5.3.1. Fonty Type1	23
5.3.2. Fonty TTF	23
5.4. PDF _T E _X i grafika	23
6. Pakiety rozszerzające możliwości L ^A T _E X-a	23
6.1. <code>hyperref</code>	23
6.1.1. Konfiguracja i opcje pakietu	24
6.1.2. Informacje o dokumencie	27
6.1.3. Polecenia zdefiniowane w pakiecie	30
6.1.4. Tworzenie zakładek	31
6.1.5. Formularze	33
6.2. <code>pdfcrypt</code>	34
6.2.1. Konfiguracja i opcje pakietu	34
6.3. <code>pdfpages</code>	35
7. Prezentacje w PDF-ie	36
7.1. Tło	38
7.2. Atrybuty strony	38
7.2.1. Przejścia między stronami w prezentacji	39
7.3. <code>pdfscreen</code>	40
7.3.1. Opcje pakietu oraz plik konfiguracyjny	40

7.3.2. Panel nawigacyjny	41
7.3.3. Pozostałe właściwości pakietu	42
7.4. pdfslide	44
7.5. ifmslide	44
8. Dołączanie dźwięku oraz filmów video	44
9. Źródła	45
Bibliografia	46
O dokumencie	46
Revision Log for <code>pedeefy.tex</code>	46

1. Wstęp

W ostatnim czasie zauważyć można duży wzrost popularności formatu PDF czemu nie należy się zbytnio dziwić, biorąc pod uwagę oferowane przez ten format możliwości oraz łatwość wyświetlania i drukowania zawartości plików PDF pod (chyba) każdym systemem operacyjnym. Format PDF, w porównaniu do POSTSCRIPT-u, charakteryzuje się następującymi cechami:

- PDF został specjalnie zaprojektowany jako format służący dystrybucji i archiwizacji dokumentów, PS zaś jest językiem opisu strony, sterujący drukarkami; PDF jest formatem bardziej przenośnym,
- istnieje szereg darmowych programów umożliwiających wyświetlanie i drukowanie plików PDF: **Acrobat Reader**, **Ghostscript**, **xpdf**, programy te dostępne są na wiele platform i systemów operacyjnych,
- funkcja **Shrink to fit** w programie **Adobe Reader** pozwala na eleganckie rozwiązanie problemów z drukowaniem dokumentów, których wymiary nie są zgodne z normami ISO,
- format PDF posiada wbudowany algorytm kompresji, dzięki czemu dokumentów nie trzeba kompresować, np. przy użyciu **pkzip**,
- PDF pozwala na włączanie do dokumentu hipertekstowych odnośników do innych dokumentów lub zasobów sieciowych (URL),
- PDF umożliwia włączanie grafik (map bitowych) w formacie JPEG, dzięki czemu dokumenty wynikowe posiadają stosunkowo małe rozmiary, ponadto mapy bitowe mogą być resamplowane.

Co prawda o tworzeniu dokumentów PDF przy użyciu \LaTeX -a napisano już bardzo dużo, nie mniej jednak wydaje się słuszne zebranie tego wszystkiego w jednym miejscu tak, aby „było pod ręką”, aby można było sięgnąć i ewentualnie przeczytać.

2. Droga do celu

Sposobów na uzyskanie plików w formacie PDF jest co najmniej kilka:

1. $\text{tex} \rightarrow \text{dvi} \rightarrow \text{pdf}$
2. $\text{tex} \rightarrow \text{dvi} \rightarrow \text{ps} \rightarrow \text{pdf}$
3. $\text{tex} \rightarrow \text{pdf}$

przy czym każdy ze sposobów ma swoje wady i zalety.

Pierwszy z nich, korzystający z formatu DVI wymaga stosowania programu **dvipdfm**.

Druga metoda, korzystająca z formatu postscriptowego, oferuje największe możliwości, gdyż zezwala na korzystanie z wielu pakietów bazujących na poleceniach postscriptowych (np. **PSTricks**), jednocześnie jest „najtrudniejszą” z metod, gdyż wymaga destylowania pliku PS przy pomocy programów **Ghostscript** lub **Acrobat** firmy Adobe. Do destylacji można użyć także programu **PStall**, który dostępny jest na licencji shareware.

Trzeci – wydaje się, że najprostszy i najbardziej logiczny sposób – to korzystanie z programu $\text{PDF}\text{\LaTeX}$ i generowanie plików PDF bezpośrednio, tym bardziej, że rozwijane są specjalne pakiety rozszerzające możliwości \LaTeX -a.

opcja	opis
-c	ignoruje polecenia <code>\special</code> odnoszące się do koloru
-f nazwa_pliku	ustawia plik z mapami czcionek, domyślnie jest to plik <code>t1fonts.map</code>
-o	nazwa pliku wyjściowego, jeśli nie zostanie podana, program nada plikowi PDF taką samą nazwę jaką ma plik DVI
-l	Landscape mode
-m numer	ustawia dodatkowe powiększenie
-p format_papieru	ustawia format papieru (letter, legal, ledger, tabloid, a4, lub a3), domyślnie ustawiony jest format letter
-r rozdzielczość	ustawia rozdzielczość dla czcionek włączanych jako bitmapy, domyślnie jest to 600 dpi
-s strony	pozwala określić zakres stron do przedestylowania, np.: -s 1-3,5-6
-t	wbudowuje ministrony (thumbnails)
-d	usuwa pliki z ministronami po skończeniu przetwarzania
-x wymiar	ustawia przesunięcie w poziomie, domyślnie 1 cal
-y wymiar	ustawia przesunięcie w pionie, domyślnie 1 cal
-e	wyłącza częściowe wbudowywanie fontów, domyślnie jest to włączone
-z liczba	pozwala wybrać stopień kompresji w zakresie od 0 do 9, domyślnie jest 9
-v	tryb verbose
-vv	tryb „bardziej” verbose

Tabela 1: Opcje programu `dvipdfm`

3. PDF z pomocą `dvipdfm`

`dvipdfm` jest ciągle rozwijanym programem umożliwiającym tworzenie plików PDF bezpośrednio z formatu DVI. Autorem programu jest Mark A. Wicks, zaś sam program charakteryzuje się następującymi cechami:

- wspomaga tworzenie zakładek (*bookmarks*), odnośników i adnotacji, w tym formularzy oraz odsyłaczy hipertekstowych,
- obsługuje dostępne w \TeX -u polecenia `\special`: DVI, Hyper \TeX , TPIC, polecenia dotyczące koloru, `PSfile` i inne `\special-e` postscriptowe,
- pozwala włączać pliki METAPOST-owe oraz niektóre pliki postscriptowe, posiłkując się przy tym zewnętrznymi programami (np. `Ghostscript`),
- pozwala na wykorzystywanie prostych transformacji wprowadzanego materiału; może on być skalowany i obracany,
- pozwala włączać „mini-strony” (*thumbnails*), wykorzystując do ich utworzenia `Ghostscript`,
- umożliwia włączanie grafik w formatach PDF, PNG, JPEG, przy czym dla dokumentów PDF wstawiane są ich pierwsze strony, które dodatkowo mogą być przycinane poprzez odpowiedni `BoundingBox`,
- wspomaga częściowe wbudowywanie czcionek i algorytm kompresji `flate`,

Program wywoływany jest z linii poleceń, zaś poszczególne opcje przedstawione są w tabeli 1. Po skompilowaniu dokumentu `.tex` i otrzymaniu pliku DVI należy użyć `dvipdfm`, zaś opcje podać jako parametry wywołania, np.:

```
dvipdfm -p a4 -f pl.map plik.dvi
```

3.1. Specjalne polecenia rozszerzające możliwości \TeX -a

Program, będący w gruncie rzeczy interpretatorem formatu DVI, rozszerza zbiór poleceń specjalnych (`\special`) o polecenia, będące obiektami PDF¹, np.:

```
\special{pdf: out 1 << /Title (Jakis tytuł)
                        /Dest [ 1 0 R /FitH 234 ] >>}
```

¹ z dwoma wyjątkami: wymiarami oraz transformacjami

nazwa	opis
@catalog	odnośnik do katalogu dokumentu
@names	odnośnik do słownika /Names
@pages	odnośnik do głównego drzewa /Pages
@resources	odnośnik do słownika zasobów bieżącej strony
@thispage	odnośnik do bieżącej strony
@page n	odnośnik do strony n
@nextpage	odnośnik do strony po @thispage
@prevpage	odnośnik do strony przed @thispage
@ypos	liczba reprezentująca bieżącą współrzędną Y wyrażoną w punktach
@xpos	liczba reprezentująca bieżącą współrzędną X wyrażoną w punktach

Tabela 2: Predefiniowane nazwy w dvipdfm

Każde polecenie `\special` T_EX-a reprezentowane jest przez inne polecenie programu `dvipdfm`, przy czym musi się ono zaczynać od `pdf:`, w przeciwnym razie zostanie zignorowane przez `dvipdfm`. Składnia używanych obiektów PDF wewnątrz `\special`-i na ogół odpowiada specyfikacji formatu PDF, z małym wyjątkiem – rozwijaniem zmiennych. `dvipdfm` umożliwia zastępowanie obiektów PDF symbolicznymi nazwami zmiennych w postaci `@name`, przy czym `name` może zawierać dowolne znaki dozwolone w obiekcie `name` formatu PDF. Zdefiniowana przez użytkownika nazwa zaczynająca się od `@` rozwijana jest w odnośnik do zdefiniowanego przez użytkownika obiektu PDF. Oprócz nazw definiowanych przez użytkownika, kilka nazw jest predefiniowanych; wymienione są one w tabeli 2.

W dokumentacji [1] znajduje się opis wszystkich `\special`-i wprowadzanych przez program, poniżej opisano kilka z nich. Dla przykładu, dodanie informacji o dokumencie, które pojawią się w okienku General Info (rys. 1) programu Acrobat można zrealizować używając polecenia `\special docinfo:`

```
\special{pdf: docinfo << /Author (Grzegorz Sapijaszko)
                        /Title (Bardzo ciekawy dokument)
                        /Subject (Cokolwiek)
                        /Keywords (wazne,slova,kluczowe)
                        /Creator (LaTeX i inne programy)
                        /ModDate (D:20001124090000) >> }
```

przy czym `dvipdfm` sam wstawia pola `/Producer` oraz `/CreationDate`.

Innym przykładem jest notatka tekstowa, którą można wstawić poleceniem `\special annotate:`

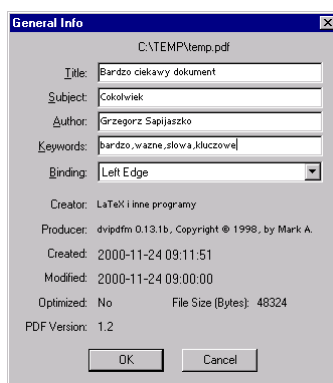
```
\special{pdf: annotate width 80pt height 36pt
  << /Type /Annot
    /Subtype /Text
    /T (Tytuł)
    /Contents (Notatka tekstowa)
    /Open true
    /C [1 0 1]
    /Border [0 0 1]
  >> }
```

3.2. dvipdfm a fonty

Należy zwrócić szczególną uwagę na tworzenie i używanie tablic kodowań, które różnią się od standardowych, wykorzystywanych przez `dvips` lub `PDFTEX`. Format pliku z tablicą jest następujący:

```
font_w_TeX-u   kodowanie   plik.afm   plik.pfb
```

przy czym domyślnie przyjmuje się, że nazwa pliku PFB jest taka sama jak nazwa pliku AFM, np. dla fontów PL:



Rysunek 1: Zawartość okna General Info uzyskana dzięki specjalnemu poleceniu `docinfo`.

```
plb10    plrm.enc  plb10
plbx10   plrm.enc  plbx10
...
plbsy10  plms.enc  plbsy10
...
% Antykwa Półtawskiego
antpr    antp.enc  antpr
antpri   antp.enc  antpri
...
```

3.3. Wstawianie plików graficznych

Program umożliwia wstawianie plików graficznych przy pomocy dwóch poleceń: `\epdf` oraz `\image`. Składnia polecenia `\epdf` jest następująca:

```
\special{pdf:epdf @nazwa
        yscale 0.50
        width 4.0in
        rotate 45
        (plik.pdf)
}
```

Polecenie to wstawia pierwszą stronę z pliku PDF podanego jako `plik.pdf`. Obiekt taki może być skalowany bądź obracany (poleceniami `scale`, `xscale`, `yscale` oraz `rotate`), wymiary można też podać przy użyciu opcji `width` i `height`. Obiekt ten przechowywany jest jako XObject i może być powtórnie wyświetlony przy użyciu polecenia `\usexobj` z podaniem nazwy `nazwa`.

Polecenie `\image` różni się od polecenia `\epdf` jedynie tym, że pozwala wstawiać pliki graficzne typu PDF, JPEG oraz PNG. Podobnie jak w przypadku `\epdf` tak wstawione obiekty można powtórnie wyświetlić przy pomocy polecenia `\usexobj` `nazwa`.

Pliki graficzne można także wstawiać w sposób klasyczny, korzystając np. z pakietu `graphicx`:

```
\usepackage[dvipdfm]{graphicx}
```

W tym przypadku pliki EPS konwertowane są przy pomocy programu `Ghostscript`.

4. Destylowanie plików postscriptowych

Jak już wspomniano poprzednio, istnieje kilka możliwości utworzenia dokumentu w formacie PDF z pliku postscriptowego. Pierwsza z nich, to użycie komercyjnego pakietu `Acrobat` firmy Adobe Systems, w skład którego wchodzi `Distiller`. Druga możliwość, to użycie bezpłatnego programu `Ghostscript`, dostępnego chyba na wszystkie systemy operacyjne. Kolejnym programem konwertującym jest `PStall`.

4.1. Adobe Acrobat

Adobe Acrobat jest obecnie jednym z najlepszych programów do przetwarzania dokumentów PDF. W skład pakietu wchodzi między innymi program **Acrobat Distiller**, pozwalający na konwersję plików postscriptowych do formatu PDF. Parametry wpływające na właściwości destylacji opisane są w [2]; parametry te mogą być dodane do destylowanego pliku PS, dzięki czemu uzyskuje się dość dobrą kontrolę nad destylacją.

Parametry można podzielić na następujące grupy:

- ogólne parametry destylacji,
- parametry kompresji grafiki,
- parametry fontów,
- parametry kolorów,
- parametry wpływające na strukturę dokumentu.

Wybrane parametry, ich wartości i opis znajduje się w następnych rozdziałach. Szczegółowy opis podany jest w [2].

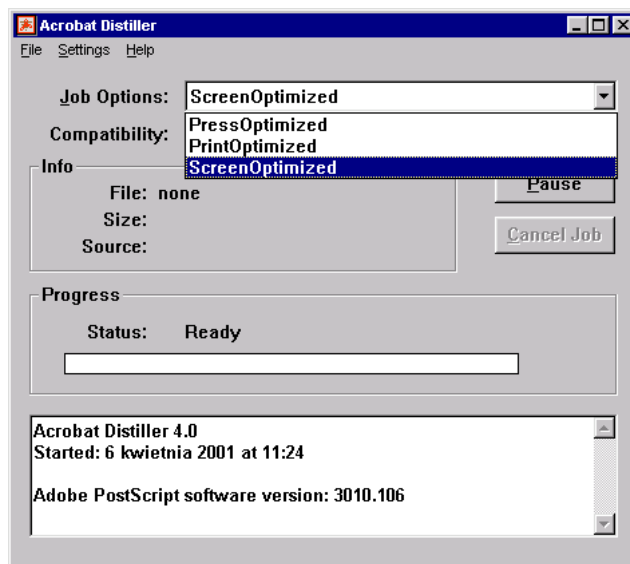
4.1.1. Ogólne parametry destylacji

- /ACSCII85EncodePages** – parametr ten, typu logicznego, określa czy obiekty binarne (np. zawartość strony, mapy bitowe, wbudowane fonty) mają być kodowane jako ASCII85 (jeśli wartość parametru jest **true**), dzięki czemu wynikowy plik PDF jest plikiem w „czystym” ASCII, czy też nie (**false**).
- /AutoRotatePages** – parametr ten pozwala Distillerowi próbować wyświetlać stronę (pionowo/poziomo) bazując na orientacji tekstu. Polecenie to brane jest pod uwagę o ile w dokumencie nie używane są polecenia **%%ViewingOrientation**, **%%PageOrientation** lub **%%Orientation**. Wartości tego parametru mogą być następujące: **/None** – wyłącza polecenie **/AutoRotatePages**, **/All** – sprawdza dominującą orientację tekstu na wszystkich stronach i obraca odpowiednio wszystkie strony oraz **/PageByPage** – sprawdza układ strona po stronie i obraca poszczególne strony; opcja ta może być użyteczna dla mieszanych dokumentów o układzie pionowym i poziomym.
- /Binding** – określa w jaki sposób wydrukowane strony będą bindowane, przy czym można podać **/Left** dla bindowania lewostronnego oraz **/Right** – dla prawostronnego.
- /CompatibilityLevel** – określa wersję formatu PDF, a wartość może wynosić 1.2 lub 1.3.
- /CompressPages** – parametr logiczny pozwalający włączyć (dla wartości **true**) lub wyłączyć (dla **false**) kompresję stron, przy czym kompresja jest typu Flate.
- /DefaultRenderingIntent** – parametr ustalający sposób renderowania dokumentu, przy czym dopuszczalne wartości to: **/Default**, **/Perceptual**, **/Saturation**, **/RelativeColorimetric** oraz **/AbsoluteColorimetric**.
- /DoThumbnails** – parametr włączający (dla wartości **true**) tworzenie ministron dla pliku PDF.
- /StartPage** oraz **/EndPage** – parametry, przy pomocy których można określić zakres stron podlegających destylacji. Domyślnie oba parametry mają wartość -1. Jeśli wartość **/StartPage** jest większa od 1, strony do **/StartPage - 1** nie są destylowane, a następna strona jest pierwszą stroną w wynikowym pliku PDF.
- /Optimize** – parametr pozwalający włączyć optymalizację pliku PDF.
- /UseFlateCompression** – parametr zmieniający algorytm kompresji; przy domyślnej wartości (**true**) używana jest kompresja ZIP. Jeśli wartość wynosi **false** oraz parametr **/CompressPages** ustawiony jest jako **true**, używana jest kompresja LZW bez względu na ustawienia parametru **/CompatibilityLevel**.

4.1.2. Parametry wpływające na kompresję grafiki

Aby zmniejszyć rozmiar wyjściowego pliku można wykorzystać fakt, iż Distiller może kompresować tekst oraz grafikę. W zależności od ustawień kompresja oraz resamplingowanie bitmap może radykalnie zmniejszyć wielkość pliku oraz czas potrzebny na wyświetlenie go na ekranie. Rysunek 2 pokazuje dostępne parametry kompresji z poziomu Distillera: wybór automatyczny, kompresja JPEG oraz ZIP.

Metoda kompresji JPEG powinna być stosowana do kolorowych map bitowych, np. fotografii. Metoda ta jest metodą stratną, tzn. aby osiągnąć określony poziom kompresji pewne dane są



Rysunek 2: Parametry destylacji dostępne z listy Job Options

wyrzucane z bitmapy, co prowadzi do zmniejszenia jakości obrazu, nie mniej jednak metoda ta próbuje zmniejszyć rozmiar wynikowego pliku z minimum strat w jakości obrazu. Z drugiej jednak strony, kompresja JPEG pozwala na uzyskanie mniejszych plików wynikowych niż przy stosowaniu kompresji ZIP.

Metoda kompresji ZIP jest użyteczna do kompresji map bitowych o dużych obszarach tego samego koloru (lub powtarzających się deseni), np. zrzutów ekranowych lub prostych grafik utworzonych w programach graficznych oraz skanowanych obrazów czarno-białych. Acrobat umożliwia cztero- i ośmiobitową kompresję ZIP.

Tabela 3 zawiera informacje o ustawieniach parametrów destylowania dla tych trzech opcji.

Parametry wpływające na kompresję kolorowych map bitowych przedstawione są poniżej.

- `/AntiAliasColorImages` – parametr ten dopuszcza (dla wartości `true`) wygładzanie krawędzi; domyślna wartość, `false`, nie zezwala na wygładzanie. Wygładzanie zwiększa ilość bitów dla każdej składowej koloru w mapach bitowych resamplowanych „w dół” tak, aby zachować pewne informacje, które w przeciwnym wypadku zostałyby stracone przy samplowaniu. Wygładzanie przeprowadzane jest tylko jeśli mapa bitowa jest aktualnie samplowana „w dół” oraz wartość parametru `/ColorImageDepth` jest większa niż ilość bitów na poszczególną składową koloru w mapie bitowej.
- `/AutoFilterColorImages` – parametr używany tylko wtedy, gdy parametr `/EncodeColorImages` ma wartość `true`. Jeśli wartość tego parametru wynosi `true`, dla zdjęć używana jest kompresja JPEG, dla grafik o wyraźnych przejściach między kolorami – kompresja Flate; jeśli `false` – metoda kompresji zależna jest od wartości parametru `/ColorImageFilter`.
- `/ColorACSDict` – parametr będący słownikiem (zbiorem) parametrów dla filtru JPEG wybranego w procesie autoselekcji.
- `/ColorImageDepth` – parametr określający ilość bitów dla każdej składowej koloru w samplowanych obrazkach (gdy `/DownsampleColorImages true`). Dozwolone wartości to: -1 (w tym przypadku ilość bitów zostaje taka sama jak w oryginalnym obrazku), 1, 2, 4 oraz 8 (dla 1, 2, 4 i 8 bitów na kanał).
- `/ColorImageDict` – słownik parametrów dla filtru JPEG. Na jakość kompresji JPEG można wpływać zmieniając wartość parametru `/QFactor`. Dla orientacji: `/QFactor = 0.25` odpowiada wysokiej kompresji, `/QFactor = 0.90` – kompresji średniej, zaś `/QFactor = 1.5` słabej kompresji. Efekty uzyskane przy stosowaniu różnych współczynników kompresji w programie Ghostscript przedstawia rys. 3.
- `/ColorImageDownsampleType` – parametr ten musi przyjmować jedną z następujących wartości:

Kompresja typu Automatic	
parametr	wartość
/AutoFilterColorImages	true
/AutoFilterGrayImages	true
/EncodeColorImages	true
/EncodeGrayImages	true
Kompresja typu JPEG	
parametr	wartość
/AutoFilterColorImages	false
/AutoFilterGrayImages	false
/EncodeColorImages	true
/EncodeGrayImages	true
/ColorImageFilter	/DCTEncode
/GrayImageFilter	/DCTEncode
Kompresja typu ZIP	
parametr	wartość
/AutoFilterColorImages	false
/AutoFilterGrayImages	false
/EncodeColorImages	true
/EncodeGrayImages	true
/EncodeMonolImages	true
/ColorImageFilter	/FlateEncode
/GrayImageFilter	/FlateEncode
/MonolImageFilter	/FlateEncode

Tabela 3: Domyślne opcje destylacji dla poszczególnych typów kompresji.

`/Average` – Distiller uśrednia grupy próbek, `/Bicubic` – używana jest interpolacja typu bicubic oraz `/Subsample`, w której brana jest wartość środkowej próbki z grupy próbek.
`/ColorImageDownsampleThreshold` – parametr ten, z zakresu 1.0 – 10, określa współczynnik rozdzielczości mapy bitowej do rozdzielczości urządzenia wyjściowego (np. monitora), powyżej którego sampłowanie będzie przeprowadzane. Dla przykładu, jeśli ustalone zostaną wartości dla parametrów:

```
/ColorImageResolution 72  
/ColorImageDownsampleThreshold 1.5
```

będą sampłowane tylko te mapy bitowe, których rozdzielczość przekracza $72 \times 1.5 = 108$ dpi.

`/ColorImageFilter` – parametr ten określa rodzaj filtru używanego do kompresji kolorowych map bitowych. Parametr ten jest ignorowany, jeśli `/AutoFilterColorImages` jest włączony lub gdy `/EncodeColorImages` ma wartość `false`. Dopuszczalne wartości to: `DCTEncode` (włącza kompresję JPEG), `FlateEncode` (włącza kompresję ZIP) oraz `LZWEncode` (włącza kompresję LZW).

Podana wartość `DCTEncode` zostanie użyta tylko wtedy, gdy obraz wyjściowy używa ośmiu bitów na kanał (tj. wartość parametru `/ColorImageDepth` wynosi 8 lub -1 i oryginalna mapa bitowa też używa 8 bitów na kanał). W przeciwnym wypadku używana jest opcja `FlateEncode` bez względu na wartość parametru `/ColorImageFilter`.

`/ColorImageResolution` – parametr określający minimalną rozdzielczość dla sampłowanych map bitowych. Parametr używany jest tylko gdy `/DownsampleColorImages` ma wartość `true`.

`/ConvertImagesToIndexed` – parametr pozwalający Distillerowi na konwersję map bitowych zawierających mniej niż 257 kolorów do indeksowanej przestrzeni kolorów.

`/DownsampleColorImages` – dla wartości `true` mapy bitowe są sampłowane „w dół” do rozdzielczości określonej przez `/ColorImageResolution`. Jeżeli parametr ma wartość `false`, sampłowanie „w dół” nie jest przeprowadzane, a rozdzielczość bitmapy w pliku PDF odpowiada tej, jaka była w źródłowym pliku PS.

`/EncodeColorImages` – parametr, który pozwala na włączenie kompresji określonej przez parametr `/ColorImageFilter`. Dla wartości `false` nie są stosowane żadne filtry kompresji.

Podobne parametry zdefiniowane zostały dla map bitowych zapisanych w odcieniach szarości oraz monochromatycznych map bitowych.

4.1.3. Parametry fontów

Distiller pozwala na kontrolę wbudowywania fontów (Type1, TrueType, multiple master) do pliku PDF co gwarantuje, że dokładnie takim samym fontem będzie renderowany ten plik w przeglądarce, niezależnie od tego, czy jest on zainstalowany w systemie, na którym PDF jest wyświetlany.

Jest to bardzo pożyteczna cecha formatu PDF nie mniej jednak należy zawsze sprawdzić w licencji na używany font czy zezwala ona na wbudowywanie fontów w pliki. Standardowe fonty rozprowadzane z dystrybucjami T_EX-a nie mają tych ograniczeń.

`/AlwaysEmbed` – parametr zawierający nazwę (lub ciąg nazw) fontów, które zawsze powinny być wbudowywane w plik PDF, przy czym każda nazwa musi być postscriptową nazwą fontu.

`/CannotEmbedFontPolicy` – parametr pozwalający na obsługę błędów w przypadku niemożliwości wbudowania fontu. Dopuszczalne wartości to: `/OK` – powodująca zignorowanie błędu jeśli font nie może zostać znaleziony lub wbudowany, `/Warning` – powodująca wyświetlenie ostrzeżenia i kontynuowanie pracy oraz `/Error` – powodująca zakończenie pracy przez Distillera w przypadku niemożliwości znalezienia bądź wbudowania fontu.

`/EmbedAllFonts` – parametr, który dla wartości `true` powoduje, iż wszystkie fonty (oprócz wymienionych w `/NeverEmbed`) są wbudowywane.

`/MaxSubsetPct` – parametr określający w procentach maksymalną ilość znaków w foncie, która może być użyta zanim cały font jest wbudowywany (zamiast podzbioru); parametr ten jest używany tylko gdy `/SubsetFonts` ma wartość `true`.

`/NeverEmbed` – nazwa lub ciąg nazw fontów, które nie mogą być wbudowywane.

`/SubsetFonts` – parametr pozwalający (dla wartości `true`) na wbudowywanie tylko używanego w dokumencie podzbioru znaków z fontu (zamiast całego fontu), co pozwala na zmniejszenie wielkości wynikowego pliku PDF. Jeżeli parametr ten zostanie włączony, decyzja o wbudowywaniu całego fontu lub tylko jego podzbioru zależy od ustawienia parametru `/MaxSubsetPct`.

4.1.4. Parametry kolorów

`/CalRGBProfile` – profil ICC, który jest używany przy dodawaniu lub konwersji plików graficznych RGB.
`/CalCMYKProfile` – profil ICC, który jest używany przy dodawaniu lub konwersji plików CMYK.
`/CalGrayProfile` – profil ICC, który jest używany przy dodawaniu lub konwersji plików w skali szarości.
`/ColorConversionStrategy` – nazwa profilu ICC używanego w strategii konwersji kolorów. Możliwe wartości to: `/LeaveColorUnchanged`, `/UseDeviceIndependentColorForImages`, `/sRGB` oraz `/UseDeviceIndependentColor`.
`/sRGBProfile` – profil ICC, który jest używany przy konwersji przestrzeni kolorów zależnej lub niezależnej sprzętowo do przestrzeni CalRGB (PDF ver. 1.2) lub sRGB (PDF ver. 1.3).
`/PreserveHalftoneInfo` – parametr decydujący o tym, czy informacje o półtonach są przekazywane do pliku PDF czy nie.
`/PreserveOverprintSettings` – parametr decydujący o przekazywaniu (lub nie) wartości operatora `setoverprint` do pliku PDF.
`/TransferFunctionInfo` – parametr ustalający nazwy funkcji transferu, wykorzystywanych przy naświetlaniu.

4.2. Ghostscript

Na wstępie uwaga: we wcześniejszych wersjach **Ghostscripta** (przed wersją 6) obsługa i wbudowywanie fontów Type 1 do dokumentów PDF nie była w pełni rozwiązana i wszystkie niestandardowe fonty Type 1 były konwertowane do fontów bitmapowych w rozdzielczości 720 pikseli. Należy zainstalować nowszą wersję programu **Ghostscript** w systemie.

Parametry destylowania wprowadzone przez Adobe wykorzystuje również **Ghostscript**, dzięki czemu stał się on dobrym, a przy tym darmowym, narzędziem konwertującym. **Ghostscript** rozprowadzany jest łącznie ze skryptami `ps2pdfxx` ułatwiającymi destylowanie, przy czym `ps2pdf12` tworzy pliki PDF w wersji 1.2, `ps2pdf13` pliki PDF w wersji 1.3, zaś `ps2pdf14` – w wersji 1.4. Skryptu używa się z linii poleceń:

```
ps2pdf [opcje] plik.[e]ps plik.pdf
```

przy czym jako **opcje** można podać każdą z opcji **Ghostscripta**. Z bogatej listy opcji bardziej przydatne mogą być następujące:

-rrozdzielczość – ustawia rozdzielczość dla wypełnień (*pattern fill*) oraz dla fontów konwertowanych do map biotowych;
-dProcessColorModel=przestrzeń_kolorów – ustala przestrzeń kolorów jaka ma być używana dla kolorów zależnych od urządzenia wyjściowego. Dopuszczalne wartości to: `/DeviceGray`, `/DeviceRGB` lub `DeviceCMYK`, domyślnie jest `/DeviceRGB`.

Co ważniejsze, aby ustawić parametry destylacji zgodne z **Acrobatem**, jako opcje można podać też `-dparametr=wartość` lub `-sparametr=ciąg`. **Ghostscript** rozpoznaje także postscriptowe operatory `setdistillerparams` oraz `currentdistillerparams`.

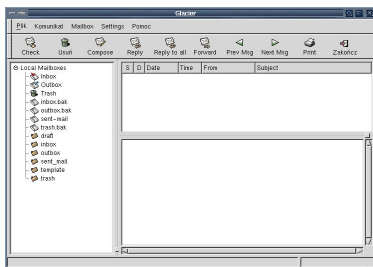
Jako opcję dla `ps2pdf` można podać też `-dPDFSETTINGS=konfiguracja`, dzięki czemu można wybrać jedno z prekonfigurowanych ustawień:

`/screen` – ustawia opcje podobne do opcji `Screen Optimized` w programie Adobe Acrobat,
`/printer` – ustawia opcje podobne do `Print Optimized`,
`/prepress` – ustawia opcje podobne do `Prepress Optimized`,
`/default` – ustawia domyślne opcje, które mogą być użyteczne dla wielu zastosowań.

Ciekawy artykuł o drukowaniu do postscriptu i destylowaniu dokumentów **Ghostscriptem** w środowisku Windows opublikował Zbigniew Okoń w [3].

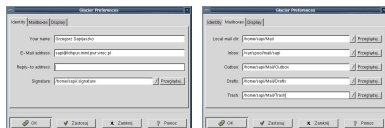
4.3. PStill

PStill jest sharewarowym konwerterem plików postscriptowych do formatu PDF napisanym przez Franka Siegerta, a dostępnym na wiele systemów operacyjnych: NeXTSTEP/OPENSTEP,

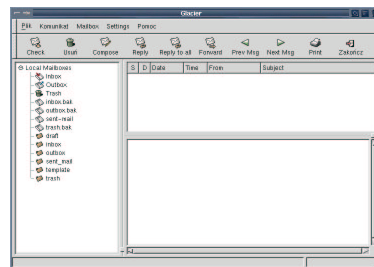


Rysunek 2: Główne okna programu Glacieur

Niestety, liczba dostępnych opcji jest bardzo ograniczona; daje się jedynie ustawić podstawowe dane użytkownika (imię i nazwisko, adres pocztowy, adres strony oraz plik z sygnaturą) oraz połączenie lokalnych skrzynek pocztowych. Możliwość ta przedstawiona są na rys. 3. W obecnej wersji program umożliwia jedynie czytanie listów, wybranie którejkolwiek możliwości tworzenia przesyłek kończy się niepowodzeniem.

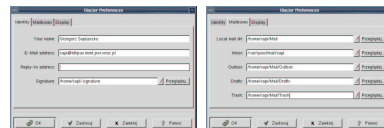


Rysunek 3: Dostępne opcje



Rysunek 2: Główne okna programu Glacieur

Niestety, liczba dostępnych opcji jest bardzo ograniczona; daje się jedynie ustawić podstawowe dane użytkownika (imię i nazwisko, adres pocztowy, adres strony oraz plik z sygnaturą) oraz połączenie lokalnych skrzynek pocztowych. Możliwość ta przedstawiona są na rys. 3. W obecnej wersji program umożliwia jedynie czytanie listów, wybranie którejkolwiek możliwości tworzenia przesyłek kończy się niepowodzeniem.



Rysunek 3: Dostępne opcje

2. AMC Mail Client

AMC Mail Client nie jest typowym programem pocztowym – można powiedzieć, że jest to graficzny interfejs próbujący połączyć możliwości oferowane przez fetchmail, sendmail oraz procmail. Dostępny jest ze strony <http://www.faceprint.com/~axion/index.html>.

2. AMC Mail Client

AMC Mail Client nie jest typowym programem pocztowym – można powiedzieć, że jest to graficzny interfejs próbujący połączyć możliwości oferowane przez fetchmail, sendmail oraz procmail. Dostępny jest ze strony <http://www.faceprint.com/~axion/index.html>.

2

2

Rysunek 3: Przykłady dokumentów destylowanych przy pomocy programu Ghostscript 6.50, z różnymi wartościami współczynnika QFactor

MacOS, Windows, Solaris oraz wszystkie systemy uniksowe. Charakteryzuje się ciekawymi własnościami:

- w systemach Windows, MacOS oraz NeXTSTEP/OPENSTEP program posiada graficzny interfejs użytkownika,
- istnieje możliwość konwersji w tle w oparciu o serwer (w systemach uniksowych oraz Windows) plików EPS i PS do formatu PDF,
- w środowisku Windows umożliwia konwersję fontów TTF do Type1.

5. PDF \TeX oraz PDF \LaTeX

PDF \TeX został opracowany przez Hàn Thé Thàhn-a jako wygodne narzędzie pozwalające na generowanie plików PDF bezpośrednio z \TeX -a, przy czym rezultat przetwarzania \TeX -em może zostać wzbogacony o efekty dostępne w formacie PDF. Jednocześnie PDF \TeX w wyniku kompilowania dokumentu może tworzyć pliki DVI – rodzaj pliku wynikowego zależy od ustawień konfiguracyjnych. Należy podkreślić, że przy domyślnych ustawieniach, zawartość plików PDF oraz DVI (treść, układ, fonty) otrzymanych w wyniku działania programu PDF \TeX jest identyczna (patrz też opis polecenia `\pdfadjustspacing` na str. 14).

PDF \LaTeX jest niczym innym, jak PDF \TeX -em z wczytanym formatem \LaTeX -a. Wszystkie polecenia specjalne wprowadzone przez PDF \TeX będą działały też w dokumencie \LaTeX -owym.

5.1. Plik konfiguracyjny dla PDF \TeX -a

PDF \TeX , podobnie jak i inne programy, po uruchomieniu, a przed kompilacją dokumentu, czyta zawartość pliku konfiguracyjnego `pdftex.cfg`. Plik ten natomiast nie jest czytany przy tworzeniu formatu dla PDF \TeX -a. W czasie generowania pliku z formatem wszystkie parametry konfiguracyjne są ustawiane domyślnie na 0, zarówno dla parametrów całkowitych jak i dla wymiarów. W standardowych dystrybucjach \TeX -a plik ten umieszczony jest w katalogu `/texmf/pdftex/config`.

Przykładowy plik konfiguracyjny przedstawiony jest poniżej.

```

pk_resolution 600
image_resolution 300
output_format 1
compress_level 9
decimal_digits 3
page_width 210 true mm
page_height 297 true mm
horigin 1 true in
vorigin 1 true in
map +pdftex.map
map +pl.map

```

gdzie poszczególne opcje oznaczają:

pk_resolution – określa w jakiej rozdzielczości powinny być generowane fonty bitmapowe,
image_resolution – określa rozdzielczość, z jaką mają być traktowane bitmapy włączane przez PDF_TE_X, domyślnie jest to 72 punkty na cal (dpi),
output_format – określa jakiego typu ma być plik wyjściowy: wartość dodatnia oznacza, że tworzone są pliki PDF, każda inna wartość – pliki DVI,
compress_level – określa stopień kompresji tekstu oraz włączanych grafik, przy czym wartość 0 określa brak kompresji, wartość 1 określa kompresję szybką, zaś 9 – najlepszą kompresję, przy czym stopień kompresji ma bezpośredni wpływ na szybkość kompilacji – im stopień wyższy tym kompilacja przebiega wolniej. Standardowo wartość ta powinna być ustawiona na 9, zaś wartość równa 0 dobra jest przy testowaniu poleceń odwołujących się do składni PDF-a. PDF_TE_X używa kompresji typu ZIP określonej w bibliotece **zlib**,
decimal_digits – określa ilość cyfr znaczących, występujących po znaku dziesiętnym w liczbach rzeczywistych opisujących wygląd strony. Dopuszczalna jest cyfra z przedziału od 1 do 5, przy czym im wyższa wartość, tym bardziej precyzyjny plik wyjściowy, ale tym samym rośnie wielkość pliku PDF oraz czas kompilacji. Parametr ten nie zmienia precyzji z jaką podawane są wartości w czystym kodzie PDF, używanych np. przez **\pdfliteral**,
page_width – oraz
page_height – określają wymiary medium wyjściowego (papieru, ekranu, itp.); jeżeli nie zostaną podane, szerokość strony wyznaczana jest jako: szerokość pudełka + 2 × (horigin + \hoffset),
horigin – oraz
vorigin – określają przesunięcie wyniku działania PDF_TE_X-a względem górnego lewego wierzchołka „papieru”,
map – określa plik, w którym zapisane jest mapowanie fontów. Plik z mapowaniem jest bardzo podobny do pliku mapowań używanego np. przez **dvips**. W pliku konfiguracyjnym można podać kilka plików z mapowaniami, jednak każdy z nich należy wpisać w oddzielnej linii, przy czym znak + umieszczony przed nazwą pliku oznacza, że zawartość pliku jest dodawana do wcześniej wczytanych mapowań, w przeciwnym razie plik ten zastępuje dotychczasowe mapowania. Jeśli żaden plik nie zostanie podany, PDF_TE_X domyślnie używa **psfonts.map**.
move_chars – określa, czy znaki z przedziału 0–31 powinny być przesunięte, czy też nie.

5.2. Polecenia specjalne PDF_TE_X-a

PDF_TE_X() wprowadza cały szereg nowych poleceń, które rozszerzają możliwości T_EX-a umożliwiając wykorzystanie specyficznych właściwości formatu PDF [4]. Każde polecenie, z wyłączeniem **\efcode** oraz rozszerzonego polecenia **\font**, rozpoczyna się od **pdf**.

5.2.1. Polecenia wpływające na wygląd dokumentu

Pierwszą grupą poleceń są polecenia, dzięki którym można sterować wyglądem dokumentu, przy czym polecenia te w większości odpowiadają opcjom z pliku konfiguracyjnego **pdftex.cfg**.

\pdfoutput (liczba całkowita) – polecenie to określa, jaki rodzaj pliku ma być wygenerowany przez PDF_TE_X: DVI lub PDF. Polecenie to odpowiada opcji **output_format** z pliku konfiguracyjnego; wartość dodatnia oznacza, że wyjściowym plikiem będzie PDF, każda inna wartość spowoduje utworzenie pliku DVI. Polecenie to nie może być użyte po wysłaniu pierwszej strony,

tn. musi być użyte zanim PDF_TE_X wyśle pierwszą stronę jeśli plikiem wynikowym ma być PDF. Jest to też jedyny parametr wymagany aby otrzymać plik PDF, pozostałe polecenia są opcjonalne.

- `\pdfcompresslevel` (liczba całkowita) – polecenie to określa stopień kompresji pliku PDF, odpowiada ono opcji `compress_level`, zaś dopuszczalne wartości zawierają się w przedziale 0 – 9.
- `\pdfdecimaldigits` (liczba całkowita) – polecenie to określa z jaką dokładnością podawane będą liczby rzeczywiste zapisywane w kodzie PDF., odpowiada ono opcji `decimal_digits`.
- `\pdfmovechars` (liczba całkowita) – polecenie to określa czy PDF_TE_X powinien próbować przenieść znaki z zakresu 0–31 na wyższe pozycje, czy też nie. Wartość 1 oznacza, że polecenie to będzie działało tylko dla tych fontów, które zawierają znaki o kodzie niższym niż 128, np. dla fontu Computer Modern Roman. Ustawiając wartość na 2 lub wyższą, wymusza się na PDF_TE_X-u próbę przeniesienia znaków z dolnego zakresu na wolne miejsca w wektorze kodowania nawet w przypadku, gdy font zawiera znaki o kodzie większym lub równym 128.
- `\pdfpkresolution` (liczba całkowita) – polecenie to określa domyślną rozdzielczość z jaką będą generowane fonty PK, wbudowywane w plik PDF. Należy jednak zauważyć, że fonty bitmapowe nie są najlepiej wyświetlane przez przeglądarki, z tego też względu zaleca się używanie fontów wektorowych wszędzie tam, gdzie jest to możliwe.
- `\pdfpagewidth` (wymiar) – polecenie to określa szerokość strony (papieru, ekranu lub innego medium wyjściowego) tworzonego dokumentu. PDF_TE_X czyta ten parametr gdy rozpoczyna wysyłanie zawartości strony; jeżeli wartość ta nie zostanie określona, szerokość strony obliczana jest jako: szerokość wysyłanego pudełka + 2 × (`\horigin` + `\hoffset`). Podobnie jak i w następnym poleceniu, wartość ta zastępuje wartość ustawioną w pliku konfiguracyjnym.
- `\pdfpageheight` (wymiar) – polecenie to określa wysokość strony tworzonego dokumentu.
- `\pdfhorigin` (wymiar) – polecenie służy do określenia poziomego przesunięcia wysyłanego pudełka względem górnego lewego narożnika strony. Wartość jednego cala odpowiada standardowemu przesunięciu T_EX-a.
- `\pdfvorigin` (wymiar) – polecenie to określa przesunięcie w pionie względem górnego lewego narożnika strony. Należy pamiętać, iż środek systemu współrzędnych używanego przez T_EX-a zaczyna się w górnym, podczas gdy PDF-a w dolnym lewym narożniku strony.
- `\pdfpagesattr` (atrybuty) – polecenie to pozwala określić atrybuty dla wszystkich stron dokumentu. Przykładowymi atrybutami mogą być: `/MediaBox` – prostokąt określający wymiary strony, `/CropBox` – prostokąt określający wymiary obszaru strony, który będzie widzialny (możliwy do wydrukowania) lub `/Rotate` – kąt w stopniach o jaki powinna być obrócona strona gdy jest wyświetlana lub drukowana, np.:

```
\pdfpagesattr {  
  /Rotate 90  
  /MediaBox [0 0 612 792]}
```

Strona obracana jest zgodnie z kierunkiem wskazówek zegara, zaś wartość kąta musi być wielokrotnością 90 stopni (lub równa zero). PDF_TE_X rozwija listę atrybutów po zakończeniu tworzenia dokumentu PDF i dodaje powstały ciąg znaków do drzewa obiektów Pages.

- `\pdfpageattr` (atrybuty) – polecenie podobne do `\pdfpagesattr` odnoszące się jednak do pojedynczej strony. Może być użyte do nadpisania atrybutów ustalonych przez `\pdfpagesattr` dla indywidualnych stron. Atrybuty umieszczone w tym poleceniu są rozwijane gdy PDF_TE_X kończy wysyłać stronę, zaś ich zawartość dodawana jest do atrybutów danej strony.

5.2.2. Polecenia ustalające informacje o dokumencie

Te polecenia pozwalają użytkownikowi na dodanie (lub zmianę) informacji o dokumencie zapisanych w pliku PDF. Między innymi na podstawie tych informacji pliki PDF są indeksowane. Informacje te dostępne są w programie Acrobat po wybraniu opcji Document Info z menu File (patrz też rys. 1).

- `\pdfinfo` `<tekst>` – polecenie to pozwala dodać informacje o dokumencie; `<tekst>` jest zbiorem par klucz-wartość, gdzie poszczególne słowa kluczowe rozpoczynają się od /, zaś wartości, będące w gruncie rzeczy ciągiem znaków, ujęte są w nawiasy okrągłe. Wszystkie słowa kluczowe są

opcjonalne, tzn. mogą ale nie muszą wystąpić. Poszczególne słowa to: `/Author`, `/CreationDate` (domyślnie wstawiana jest bieżąca data), `/ModDate`, `/Creator` (domyślnie jest `TEX`), `/Producer` (wartość domyślna to `PDFTEX`), `/Title`, `/Subject`, oraz `/Keywords`.

Słowa kluczowe `/CreationDate` oraz `/ModDate` wyrażone są jako ciąg `D:YYYYMMDDhhmmss`, gdzie `YYYY` określa rok, `MM` – miesiąc, `DD` – dzień, `hh` oznacza godzinę, `mm` – minuty oraz `ss` – sekundy.

Polecenie `\pdfinfo` powinno wystąpić w dokumencie tylko raz, jeśli pojawi się kilka razy, wzięte pod uwagę będzie tylko pierwsze wystąpienie. Podobnie, jeżeli w poleceniu jakaś para klucz–wartość wystąpi więcej niż jeden raz, pod uwagę wzięta będzie ta wartość, która pojawiła się jako pierwsza. Przykładowe polecenie może wyglądać następująco:

```
\pdfinfo {
  /Author (Grzegorz Sapijaszko)
  /Title (Tworzenie plikow pdf)
  /Keywords (pdf,LaTeX,pdfTeX)
}
```

`\pdfcatalog <tekst> openaction <specyfikacja>` – polecenie to wstawia odpowiednie wartości do katalogu dokumentu, kluczami w tym przypadku są `/URI`, określający podstawowy URL dokumentu oraz `/PageMode`, określający w jaki sposób **Acrobat** ma wyświetlać dokument zaraz po wczytaniu. Dopuszczalne wartości dla `/PageMode` przedstawiono poniżej:

<code>/UseNone</code>	wyświetlana jest tylko strona
<code>/UseOutlines</code>	wyświetlane są zakładki
<code>/UseThumbs</code>	wyświetlane są ministrony (<i>thumbnails</i>)
<code>/FullScreen</code>	dokument wyświetlany jest na całym ekranie

W trybie `/FullScreen` na ekranie monitora wyświetlany jest tylko dokument, nie są wyświetlane kontrolki okna, menu ani inne okna.

Dodatkowa opcja polecenia `\pdfcatalog, openaction`, może służyć do wywołania określonej akcji przy otwieraniu dokumentu, przy czym składnia jest taka sama jak przy tworzeniu odsyłaczy wewnątrz dokumentu (patrz rozdz. 5.2.7).

`\pdfnames <tekst>` – to polecenie wstawia `tekst` do wektora `/Names`, przy czym `tekst` musi odpowiadać specyfikacji przedstawionej w [5], w przeciwnym wypadku dokument PDF może być uszkodzony.

5.2.3. Polecenia sterujące fontami

Program `PDFTEX` rozszerza standardowe możliwości `TEX`-a w zakresie obsługi fontów.

`\font <specyfikacja> stretch <liczba> shrink <liczba> step <liczba>` – polecenie pozwalające na zdefiniowanie fontu, np.:

```
\font\jakisfont=jakisplik at 10pt stretch 30 shrink 20 step 10
```

Te dodatkowe opcje, `stretch`, `shrink` oraz `step` umożliwiają `PDFTEX`-owi dostosowanie fontu, którym składany jest dany fragment, do specyficznych sytuacji. Inaczej mówiąc dla `TEX`-a znaczy to mniej więcej tyle: „jeśli coś idzie źle, możesz rozciągnąć (`stretch`) znaki w foncie maksymalnie o 3% lub pomniejszyć (`shrink`) je o 2%”. `PDFTEX` używa wewnętrznych struktur danych o stałych szerokościach, każda dodatkowa szerokość oznacza także dodatkowy font. Z przyczyn praktycznych `PDFTEX` używa dyskretnych wartości, w tym przypadku 1% – znaczy to, że dla fontu `jakisplik` może być użytych maksymalnie 6 różnych odmian. Jeżeli nie jest podana wartość `step`, używana jest podziałka 0.5%.

Nazwy dodatkowych fontów tworzone są od nazwy fontu podstawowego, np. `jakisplik+10` lub `jakisplik-15`. Dodatkowo muszą być dostępne pliki TFM o takich nazwach i zawierające font odpowiednio przeskalowany – każdy przeskalowany font musi posiadać odpowiadający mu plik TFM. Jeżeli plik TFM nie zostanie znaleziony, `PDFTEX` spróbuje utworzyć taki plik wykorzystując w tym celu program `mktextfm` o ile będzie on dostępny.

`\pdfadjustspacing <liczba całkowita>` – polecenie to pozwala wpływać na `PDFTEX`-a przy formatowaniu rozdziałów.

Wynik działania PDF \TeX -a jest identyczny z wynikiem działania \TeX -a, gdyż standardowo optymalizacja opisana w poprzednim punkcie jest wyłączona – *engine* \TeX -a pozostaje nie zmienione. Po ustawieniu wartości `\pdfadjustspacing` równej 1, rozciąganie ustalone przez parametr `stretch` dokonywane jest po sformatowaniu akapitu przez \TeX -a, dzięki czemu łamanie linii jest identyczne z łamaniem dokonanym przez \TeX -a.

W przypadku ustawienia tej wartości na 2 zmiany szerokości fontów będące wynikiem rozciągania bądź zwięzania brane są pod uwagę podczas formatowania akapitu. W tym przypadku łamanie linii może różnić się od standardowego łamania \TeX -em w wyniku czego akapity mogą być dłuższe lub krótsze.

Obie możliwości wykorzystują fonty TFM odpowiadające ustawieniom skalowania opisanym w poprzednim punkcie.

`\efcode <liczba całkowita>` – polecenie to pozwala zdefiniować zdolność rozciągania się poszczególnych znaków, np.:

```
\efcode'A=2500
\efcode'O=0
```

co oznacza, że A może rozciągnąć się 2.5 raza zaś O nie może być wogóle rozciągane. Domyślną wartością jest 1000 co odpowiada 100%.

`\pdffontname ` – to polecenie pozwala określić numer jaki został nadany danemu fontowi w pliku PDF. Font wewnątrz pliku PDF zapisywany jest jako ciąg `/F<numer>` `<numer obiektu>` 0 R, np.:

```
/Font << /F19 16 0 R >>
```

Dla fontu w podanym przykładzie polecenie `\pdffontname` zwróciłoby liczbę 19.

`\pdffontobjnum ` – polecenie to, podobne do `\pdffontname`, zwraca `<numer obiektu>` dla użytego fontu.

Używając tych dwóch poleceń można uzyskać informacje o wszystkich fontach użytych w danym dokumencie.

`\pdfincludechars <tekst>` – polecenie pozwalające wymusić na PDF \TeX -u włączenie znaków podanych jako `<tekst>` do tworzonego dokumentu PDF. Domyślnie PDF \TeX włącza do dokumentu tylko używane znaki. Przykładowo, polecenie to może wyglądać następująco:

```
\pdfincludechars\font{ABCDEF}
```

5.2.4. Polecenia wspomagające obiekty specjalne (XObject)

Kolejne trzy polecenia wspomagają ciekawą właściwość formatu PDF, jaką jest zdolność do wielokrotnego użycia tego samego elementu w dokumencie. W praktyce wygląda to tak, iż najpierw tworzony jest forma obiektu (XObject) w PDF-ie. Zawartość tego obiektu odpowiada zawartości \TeX -owego pudełka – może zawierać grafikę a także odnośniki do innych elementów XObject. Po utworzeniu, forma XObject może być wielokrotnie używana; zamiast wstawiać dany obiekt ponownie podaje się odnośnik do numeru tego elementu. Właściwość ta może być bardzo przydatna przy tworzeniu dokumentów, zawierających dużą ilość takich samych elementów.

`\pdfxform <attributy> <zasoby> <numer pudełka>` – polecenie to tworzy formę XObject odpowiadającą zawartości pudełka podanego jako `<numer pudełka>`. Pudełko to może zawierać inne obiekty, formy XObject lub grafikę, nie może natomiast zawierać notatek tekstowych, gdyż te tworzone są na osobnej warstwie oraz miejsce umieszczenia notatek na stronie nie zależy od położenia innych elementów.

`\pdfrefxform <liczba całkowita>` – polecenie to powoduje zapisanie formy w pliku wynikowym. Utworzona forma przechowywana jest w pamięci, a zapisywana w dokumencie PDF dopiero wtedy, gdy jej numer jest wywołany przez polecenie `\pdfrefxform` lub gdy polecenie `\pdfrefxform` poprzedzone jest poleceniem `\immediate`.

`\pdflastxform` – polecenie udostępnia numer ostatniej utworzonej formy.

5.2.5. Polecenia włączające pliki graficzne

Format PDF przewiduje możliwość wbudowywania grafiki oraz obiektów tekstowych (form XObject). W PDF \TeX -u umożliwiają to polecenia opisane w rozdziale 5.2.4, nie mniej jednak

Subject: bardzo proszę o pomoc
 From: "katya" <dali@promail.pl>
 prosze was powiedzcie mi jak opisać tylko bardzo
 mądrze i rzeczowo co to jest "małpa"... próbowałam
 już swoimi słowami ale chyba nie bardzo mądrze...
 prosze pomóżcie.

28.07.1999, pl.comp.mail, *Remi Schleicher*:

Znaczek @ (at) - jest najczęściej zwany „małpą”. Oddziela nazwę konta użytkownika od nazwy hosta. Jego kształt pochodzi od angielskiego słowa *at*, czasami, choć dość rzadko można spotkać czyjś adres e-mail zapisany w postaci user(at)host.docelowy, w takim wypadku Twój adres wyglądałby dali(at)promail.pl.

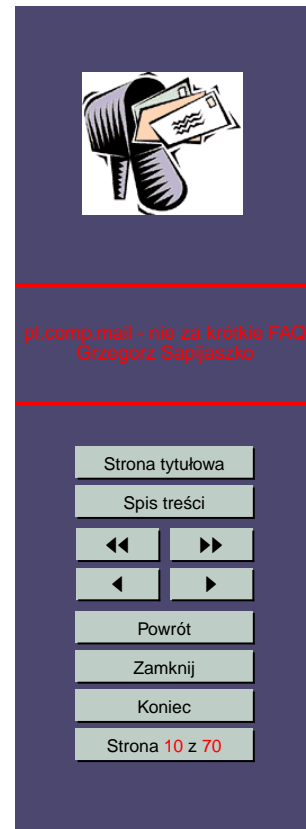
From news@black.gliwice.pl
 może głupie, ale... jak się czyta
 (fonetycznie) - "et" "at" "ejti" ?

01.08.1999, pl.comp.mail, *Remi Schleicher*:

Fonetycznie? AFAIK „et” → patrz dowolny słownik j. angielskiego. Adres zapisany jako user@host czytamy „user at host” – IMHO.

6.11.1999, prywatnie, *Ziemek Borowski*:

W typografii @ nazywa się comercial at.



Rysunek 4: Przykład ilustracji utworzonej z pliku PDF poleceniem `\pdfximage` z podaniem numeru strony.

bitmapy zostały potraktowane jako specjalny rodzaj formy `XObject` i do ich obsługi utworzone zostały osobne polecenia.

`\pdfximage <reguły> <atrybuty> <strona> <plik>` – Polecenie to pozwala na wstawienie pliku graficznego bezpośrednio do PDF-a; formaty obsługiwane przez PDF_TE_X-a (podane poniżej) rozpoznawane są po rozszerzeniach plików:

PDF .pdf, .PDF
 PNG .png, .PNG
 TIFF .tif, .TIF, .tiff, .TIFF
 JPEG .jpg, .JPG, .jpeg, .JPEG

Wymiary rysunku mogą być określone przez `<reguły>`, domyślnymi wartościami są: 0 dla „depth” oraz bieżąca szerokość i wysokość rysunku. Jeżeli wszystkie wymiary zostaną podane, rysunek zostanie odpowiednio przeskalowany, jeżeli podane zostaną tylko niektóre z nich, podana wartość została zachowana, a pozostałe wymiary rysunku zostaną przeskalowane z zachowaniem proporcji szerokość : (wysokość + *depth*), gdzie *depth* traktowane jest jako zero. Jeżeli żaden wymiar nie zostanie podany, rysunek jest wstawiany „jak jest”.

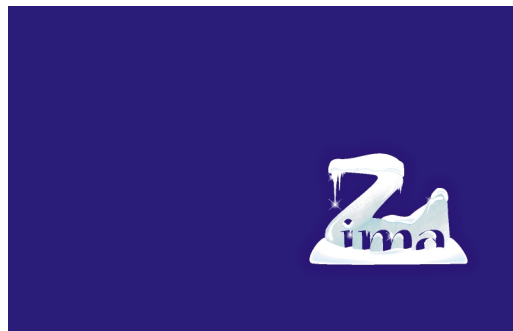
Pliki PDF mogą zawierać wiele stron; przy wstawianiu takiego pliku można odwołać się do konkretnej strony, np.:

```
\pdfximage page 10 {emailfaq.pdf}
\pdfrefximage\pdflastximage
```

Utworzoną w ten sposób ilustrację przedstawia rys. 4. Należy jednak pamiętać, że w formacie PDF numeracja stron zaczyna się od 1. Poleceniem `\pdflastximagepages` można określić ilość stron we wstawionym ostatnio dokumencie PDF.

`\pdfimageresolution <liczba całkowita>` – polecenie to określa rozdzielczość wstawianych rysunków, przy czym rozdzielczość podawana jest w punktach na cal. Niektóre rysunki zawierają


```
\pdfximage
width \textwidth {zima.jpg}
\pdfrefximage\pdflastximage
```



Rysunek 5: Rysunek o ustalonej szerokości (`\textwidth`), którego wysokość została przeskalowana tak, aby zachować stosunek boków zgodny z oryginalnym.

```
\pdfximage
height 2.5cm {zima.jpg}
\pdfrefximage\pdflastximage
```



Rysunek 6: Rysunek o ustalonej wysokości (2.5 cm), którego szerokość została przeskalowana tak, aby zachować stosunek boków zgodny z oryginalnym.

informację o rozdzielczości – w takim przypadku rysunek zostanie przeskalowany tak, aby uzyskać wymaganą rozdzielczość.

`\pdfrefximage <numer_obiektu>` – polecenie pozwala zapisać do pliku rysunek utworzony poleceniem `\pdfximage`.

Podobnie jak w przypadku `\pdfobj`, rysunek utworzony przy pomocy `\pdfximage` jest przetrzymywany w pamięci, a do wynikowego dokumentu PDF wstawiany jest tylko w dwu przypadkach: gdy rysunek jest wywołany poleceniem `\pdfrefximage <numer_obiektu>` lub gdy polecenie `\pdfximage` poprzedzone jest poleceniem `\immediate`.

Należy także zauważyć, że `\immediate\pdfximage` nie powoduje wyświetlenia rysunku, a jedynie gwarantuje zapisanie rysunku w pliku wynikowym. Aby wyświetlić rysunek należy użyć polecenia `\pdfrefximage <numer_obiektu>` w wymaganym miejscu.

`\pdflastximage` – polecenie to określa numer ostatnio tworzonych rysunku.

Wymiary rysunku można określić umieszczając polecenia `\pdfrefximage` w pudełku i sprawdzenie wymiarów pudełka:

```
\setbox0=\hbox{\pdfximage{plik.png}\pdfrefximage\pdflastximage}
szerokość = \the\wd0
wysokość = \the\ht0
```

przy czym `\wd0` i `\ht0` podane są w punktach.

Przykłady wstawiania plików graficznych przy wykorzystaniu PDF_TE_X-a pokazano na rysunkach 5–8.

Należy zwrócić uwagę, iż powyższe polecenia są „wewnętrznymi” poleceniami PDF_TE_X-a – korzystając z L^AT_EX-a rysunki można wstawiać (i jest to dużo prostsze) przy pomocy pakietu `graphicx`, np.:

```
\usepackage[pdfTeX]{graphicx}
...
\includegraphics[width=\textwidth]{obrazek}
```

pozostawiając całą resztę PDF_TE_X-owi.

```
\pdfximage
width 3cm height 3cm
{zima.jpg}
\pdfrefximage\pdflastximage
```



Rysunek 7: Rysunek o ustalonych wymiarach (3×3 cm), w którym nie został zachowany oryginalny stosunek boków.

```
\pdfimageresolution=72
\pdfximage {zima.jpg}
\setbox0=\hbox{\pdfrefximage\pdflastximage}
\dimen0=.06\wd0
```



```
\pdfximage width \dimen0 {pic.pdf}
\pdfrefximage \pdflastximage
```

Rysunek 8: Ilustracja wstawiona w rozdzielczości 1200 dpi

5.2.6. Polecenia wstawiające notatki

Format PDF pozwala na tworzenie notatek kilku podstawowych typów: tekstową, hipertekstową, dźwiękową lub film wideo. Więcej na ten temat napisano w rozdziale 8.

`\pdfannot <reguły> <tekst>` – polecenie to wstawia notatkę, której wymiary określone są jako `<reguły>`, zaś treść zdefiniowana jest przez `<tekst>`. W przypadku notatki dźwiękowej lub wideo, domyślnymi rozmiarami są wymiary pliku graficznego lub filmu.

`\pdflastannot` – to polecenie zwraca numer ostatniej utworzonej notatki.

Oba polecenia pozwalają użytkownikowi na utworzenie dowolnej notatki, która nie może zostać utworzona przy pomocy `\pdfstartlink` (zobacz też rozdz. 5.2.7).

5.2.7. Polecenia tworzące hiperłącza

Hiperłącze jest jednym z typów notatki, zdefiniowanej trzema poleceniami: `\pdfdest` oraz `\pdfstartlink` i `\pdfstoplink`.

`\pdfdest <specyfikacja>` – polecenie to tworzy obszar docelowy dla hiperłącza lub dla zakładki (*bookmark*), łącze jest identyfikowane przez numer lub nazwę symboliczną. Jednocześnie musi być podany sposób jak ma być wyświetlany ten obszar, przy czym jako `<specyfikacja>` dopuszczalne są:

<code>fit</code>	dopasowuje stronę do okna przeglądarki
<code>fith</code>	dopasowuje szerokość strony do okna
<code>fitv</code>	dopasowuje wysokość strony do okna
<code>fitb</code>	dopasowuje obszar określony przez Bounding Box
<code>fitbh</code>	dopasowuje szerokość obszaru Bounding Box
<code>fitbv</code>	dopasowuje wysokość obszaru Bounding Box
<code>fitr <spec></code>	dopasowuje do obszaru podanego jako <code><spec></code>

Dla przykładu:

```
\pdfdest name{obszar1} fitb
```

`\pdfstartlink <reguły> <atrybuty> <akcja>` – polecenie to rozpoczyna tworzenie łącza, gdzie wymiary mogą być podane jako `<reguły>`, atrybuty dodatkowe jako `<atrybuty>`. `<akcja>` określa zachowanie się przeglądarki po wskazaniu łącza; dopuszczalne akcje i ich opcje wymienione są w tab. 4. Przykładowe łącze może wyglądać w sposób następujący:

```
\pdfstartlink
height 10pt depth 2pt
```

```

attr {
  /C [1 0 0]
  /Border [0 0 2]}
goto page 1 {/Fit}
Przykładowe łącze
\pdfendlink

```

w którym wymiary określone są jako **height** oraz **depth**, dodatkowe atrybuty jako **attr** (w tym przypadku kolor i szerokość obramowania), akcja wskazuje na skok do pierwszej strony dokumentu i dopasowanie jej do rozmiaru okna.

Wszystkie niestandardowe akcje mogą być zdefiniowane przez użytkownika, np. możliwe jest zdefiniowanie odnośnika do [strony internetowej](#) :

```

\pdfstartlink
user{
  /Subtype /Link
  /A <<
    /Type /Action
    /S /URI
    /URI (http://www.gust.org.pl/)
  >>}
\textcolor{blue}{strony internetowej}
\pdfendlink

```

Obramowanie dookoła łącza może być usunięte przez zdefiniowanie atrybutu `attr { /Border [0 0 0]}`.

Innym przykładem może być odnośnik do [poprzedniego widoku](#) w oknie Readera:

```

\pdfstartlink
attr{ /Border [0 0 0]}
user {
  /Subtype /Link
  /A <<
    /S /Named
    /N /GoBack
  >>}
\textcolor{cyan}{poprzedniego widoku}
\pdfendlink

```

Należy zaznaczyć, że `\pdfstartlink` nie może zaczynać paragrafu, musi być poprzedzone tekstem lub jednym z poleceń: `\indent`, `\noindent` lub `\leavevmode`, w przeciwnym wypadku `\pdfstartlink` zakończy poprzedni rozdział i będzie sygnalizować błąd.

`\pdfendlink` – polecenie to kończy tworzenie hiperłącza rozpoczęte poleceniem `\pdfstartlink`. Tekst zawarty pomiędzy tymi dwoma poleceniami traktowany jest jako część linku, przy czym jeżeli tekst nie zmieści się w jednej linii (lub na stronie), PDF_{TEX} rozbije taki link na kilka pojedynczych wskazujących ten sam obiekt docelowy.

`\pdflinkmargin wymiar` – polecenie to określa dodatkowy margines dla ramki otaczającej hiperłącze.

5.2.8. Polecenia wspomagające zakładki

`\pdfoutline <akcja> count <liczba całkowita> <tekst>` – polecenie powoduje utworzenie zakładki (*bookmark*). Parametr podany jako `<akcja>` definiuje polecenie jakie ma być wykonane po kliknięciu danej zakładki, zaś składnia tego polecenia jest taka sama jak w przypadku `\pdfstartlink` (patrz str. 18). Liczba `<count>` podaje ilość zakładek niższego poziomu umieszczonych bezpośrednio pod bieżącą zakładką. Jeżeli wartość ta wynosi 0 lub nie występuje, oznacza to, że nie ma zakładek niższego poziomu. Jeżeli wartość podana w `<count>` jest wartością ujemną, wszystkie zakładki niższego poziomu mają być zwinięte. `<tekst>` jest tekstem, jaki pojawi się jako treść zakładki. Należy przy tym zauważyć, że `<tekst>` ograniczony jest do znaków zgodnych z PDFDocEncoding. Dla przykładu:

akcja	parametr	opis
goto	file{plik.pdf}	opcjonalna nazwa pliku, do którego odnosi się łącze (może występować w akcjach goto lub thread). Jeżeli identyfikatorem akcji jest nazwa to w pliku plik.pdf powinien znajdować się obszar (lub wątek) o podanej nazwie utworzony przy pomocy \pdfdest; jeżeli identyfikatorem jest numer, to w przypadku akcji goto oznacza on numer strony (dokument jest wyświetlany tak jakby było podane fitb), w przypadku akcji thread – oznacza on numer wątku (pierwszy wątek w dokumencie ma numer 0).
	num 1	powoduje przejście do obszaru określonego numerem 1
	name{nazwa}	powoduje przejście do obszaru określonego nazwą nazwa
	page 1 {/Fit}	powoduje przejście do pierwszej strony oraz dopasowanie całej strony do okna przeglądarki
thread	file{plik.pdf}	opcjonalna nazwa pliku
	num 1	powoduje przejście do wątku o identyfikatorze 1
	name{thread2}	powoduje przejście do wątku o nazwie thread2
user		akcje definiowane są przez użytkownika

Tabela 4: Dopuszczalne akcje w hipertłaczach PDF_TE_X-a

```

\pdfoutline goto num 1 count 3 {Zakładka 1}
  \pdfoutline goto num 1 count -2 {Zakładka 1.1}
    \pdfoutline goto num 1 {Zakładka 1.1.1}
    \pdfoutline goto num 1 {Zakładka 1.1.2}
  \pdfoutline goto num 1 {Zakładka 1.2}
  \pdfoutline goto num 1 {Zakładka 1.3}
\pdfoutline goto page 1 {/Fit} {Zakładka 2}

```

5.2.9. Polecenia tworzące wątki

`\pdfthread` <warunki> <atrybuty> <numer> – polecenie to definiuje wątek artykułu. Wątki o tym samym numerze rozrzucone po całym dokumencie są łączone w jeden.

`\pdfthreadmargin` <wymiar> – polecenie to definiuje dodatkowy margines dodawany do wymiarów wątku.

5.2.10. Pozostałe polecenia

`\pdfliteral` <tekst> – polecenie to, podobnie jak polecenie `\special` w TeX-u powoduje wstawienie kodu PDF, podanego jako <tekst> do pliku. Polecenie to może służyć do przekształceń koloru lub tekstu. Transformacje dokonywane są przy pomocy macierzy transformujących – ogólnie mówiąc macierz zapisana jest jako 6 cyfr, po których musi wystąpić operator `cm`. Na ilustracjach 9–11 przedstawiono przykłady zastosowania polecenia `\pdfliteral` do przekształcania tekstu.

Definiujemy kilka rzeczy:

```

\font\f=plr10 at 20pt
\setbox0=\hbox{\f Tekst obrócony}
\setbox1=\hbox{\f Tekst przeskalowany}
\setbox2=\hbox{\f Tekst pochyły}

\newdimen\d
\newbox\tmpbox
\def\avoidboxdimen#1{%
  \setbox\tmpbox=\hbox{\box#1}%
  \wd\tmpbox=0pt
  \ht\tmpbox=0pt
  \dp\tmpbox=0pt
  \box\tmpbox}

```

Rysunek 9: Obrót uzyskany przy pomocy `\pdfliteral`

Tekst przeskalowany

Rysunek 10: Skalowanie tekstu dokonane przy pomocy `\pdfliteral`

Obrót o kąt t w kierunku przeciwnym do kierunku ruchu wskazówek zegara podany jako macierz $\begin{pmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{pmatrix}$ przedstawia rys. 9.

```
\vskip\wd0
\leftline{\hskip\ht0\hskip\dp0%
\pdfliteral{q 0 1 -1 0 0 0 cm}%
\avoidboxdimen 0
\pdfliteral{Q}}
```

Skalowanie zapisuje się jako $S_x \ 0 \ 0 \ S_y \ 0 \ 0$, a wynik przedstawiony jest na rys. 10:

```
\d=\ht1 \advance\d by \dp1
\vskip3\d
\pdfliteral{q 1.9 0 0 3 0 0 cm}
\avoidboxdimen 1
\pdfliteral{Q}
```

Pochylenie osi X o kąt u oraz osi Y o kąt v dokonuje się przy pomocy macierzy $\begin{pmatrix} 1 & \tan(u) \\ \tan(v) & 1 \end{pmatrix}$ (rys. 11).

```
\d=\ht2 \advance\d by \dp2
\vskip\d
\d=0.57735\wd2 %tan(30) = 0.57735
\pdfliteral{q 1 -0.57735 0 1 0 0 cm}%
\avoidboxdimen 2
\pdfliteral{Q}
\vskip\d
```

`\pdfobj <specyfikacja> <tekst>` – polecenie to tworzy obiekt w dokumencie PDF, który dodawany jest do pliku jako `1 0 obj << ... >> endobj`. Obiekt ten przetrzymywany jest w pamięci, a do pliku zapisywany jest jedynie gdy jest on wstawiony przez polecenie `\pdfrefobject`.

`\pdfrefobj <liczba całkowita>` – polecenie to pozwala na wstawienie do pliku PDF obiektu utworzonego przy pomocy `\pdfobj`.

`\pdflastobj` – to polecenie zwraca numer ostatnio wstawionego obiektu.

Tekst pochyły

Rysunek 11: Pochylenie tekstu dokonane przy pomocy `\pdfliteral`

Jedną z najczęściej poruszanych kwestii związanych z tworzeniem „poprawnych” plików PDF jest jakość wyświetlanych na ekranie monitora fontów. O ile format PDF pozwala na używanie w pliku zarówno fontów wektorowych (Type1, True Type) jak i fontów bitmapowych (Type3), zalecane jest stosowanie wyłącznie fontów wektorowych. Fonty bitmapowe, o ile przygotowane zostaną z dużą rozdzielczością (600 dpi), dobrze będą wyglądały na wydruku, jednak czytanie z ekranu dokumentu z takimi fontami będzie utrudnione. Dodatkową negatywną cechą jest fakt, iż pliki PDF zawierające fonty bitmapowe są objętościowo większe od równoważnych im plików z fontami wektorowymi.

Jedną z najczęściej poruszanych kwestii związanych z tworzeniem „poprawnych” plików PDF jest jakość wyświetlanych na ekranie monitora fontów. O ile format PDF pozwala na używanie w pliku zarówno fontów wektorowych (Type1, True Type) jak i fontów bitmapowych (Type3), zalecane jest stosowanie wyłącznie fontów wektorowych. Fonty bitmapowe, o ile przygotowane zostaną z dużą rozdzielczością (600 dpi), dobrze będą wyglądały na wydruku, jednak czytanie z ekranu dokumentu z takimi fontami będzie utrudnione. Dodatkową negatywną cechą jest fakt, iż pliki PDF zawierające fonty bitmapowe są objętościowo większe od równoważnych im plików z fontami wektorowymi.

Rysunek 12: Różnice w wyglądzie dokumentów PDF z wbudowanymi fontami wektorowymi i bitmapowymi.

`\pdftexversion` – polecenie to zwraca numer PDF \TeX -a pomnożony przez 100, np. wersja, którą kompilowany jest ten dokument nosi numer 14.

`\pdftexrevision` – polecenie to zwraca numer rewizji PDF \TeX -a, np. dla wersji 14h jest to h.

5.3. Fonty i PDF \TeX

Jedną z najczęściej poruszanych kwestii związanych z tworzeniem „poprawnych” plików PDF jest jakość wyświetlanych na ekranie monitora fontów. O ile format PDF pozwala na używanie w pliku zarówno fontów wektorowych (Type1, True Type) jak i fontów bitmapowych (Type3), zalecane jest stosowanie wyłącznie fontów wektorowych. Fonty bitmapowe, o ile przygotowane zostaną z dużą rozdzielczością (600 dpi), dobrze będą wyglądały na wydruku, jednak czytanie z ekranu dokumentu z takimi fontami będzie utrudnione. Dodatkową negatywną cechą jest fakt, iż pliki PDF zawierające fonty bitmapowe są objętościowo większe od równoważnych im plików z fontami wektorowymi. Różnice w wyglądzie przedstawia rys. 12.

Podczas włączania grafiki w formacie PDF zawierającej fonty, PDF \TeX próbuje zastąpić wbudowane w plik graficzny fonty ich zasobami (*by its resources*), aby niepotrzebnie nie powielać plików z fontami. Jeżeli postscriptowa nazwa fontu wbudowanego w plik z grafiką zostanie znaleziona w pliku `.map`, PDF \TeX zakłada, że plik z fontem (`.pfb`) znajduje się na dysku. Dlatego też font nie jest kopiowany z pliku z grafiką w momencie wstawiania tego pliku, a zostanie on włączony na końcu. Dzięki temu font, który występuje kilka razy w różnych włączanych plikach z grafiką, włączany jest tylko raz. Jeżeli nazwa postscriptowa nie zostanie znaleziona w plikach `.map`, cały font jest kopiowany w momencie włączania grafiki PDF.

Instalację fontów Type1 oraz TrueType opisali też Tomasz Przechlewski i Stanisław Wawrykiewicz w [6].

5.3.1. Fonty Type1

W większości dystrybucji \TeX -a rozprowadzany jest zestaw fontów Type1, umożliwiający składanie poprawnych dokumentów PDF. Dostępne są zarówno standardowe fonty Computer Modern (CM) jak i fonty Times/Helvetica/Courier firmy Adobe rozprowadzane wraz z **Acrobat Readerem** lub odpowiadające im fonty URW++ Nimbus rozprowadzane wraz z **Ghostscriptem**. Dodatkowo, w katalogu `/fonts` na serwerach CTAN dostępne są inne fonty Type1, w tym fonty PL (są to zmodyfikowane fonty CM z polskimi diakrytykami), czy też spolonizowane fonty URW: QuasiBokman, QuasiChancery, QuasiCourier, QuasiSwiss, QuasiPalatino oraz QuasiTimes, czy wreszcie rodzime polskie fonty: Antykwa Toruńska i Antykwa Półtawskiego. Najnowsze wersje tych fontów można znaleźć pod adresem <http://www.gust.org.pl/fonty/index.html>.

Instalacja fontów sprowadza się do skopiowania plików do katalogu `/fonts/` w drzewie katalogów `texmf`, odświeżenie baz danych \TeX -a oraz dodanie pliku z mapowaniem fontów w pliku `pdftex.cfg`.

5.3.2. Fonty TTF

Możliwość wykorzystania fontów TrueType w (pdf) \LaTeX -u oraz procedurę instalowania opisał Damir Rakityansky w [7].

5.4. PDF \TeX i grafika

Jak już wspomniano wcześniej, PDF \TeX (oraz PDF \LaTeX) pozwala na wstawianie plików graficznych w formatach: PDF, PNG, TIFF oraz JPEG. Format EPS, który jest standardowym formatem plików graficznych używanych przez \LaTeX -a, nie jest niestety obsługiwany przez PDF \TeX -a, z tego też powodu pliki EPS muszą być przekonwertowane do innego formatu, np. do formatu PDF. Można tego dokonać używając **Ghostscripta**, **Distillera** lub innego programu konwertującego. Rozmiar pliku PDF określany jest, o ile to możliwe, na podstawie **CropBox**, jeśli nie – na podstawie **MediaBox**. Aby uzyskać właściwe rozmiary **MediaBox** należy dokonać transformacji pliku EPS tak, aby punkt początkowy znajdował się dokładnie w punkcie o współrzędnych (0, 0) oraz aby rozmiar strony odpowiadał wartościom podanym w **BoundingBox**. Do tego celu można wykorzystać skrypt **epstopdf** autorstwa Sebastiana Rahtza.

Na temat różnych aspektów wstawiania plików graficznych do publikacji tworzonych w \LaTeX -u można przeczytać w [8].

Generalnie należy zwrócić uwagę, aby używać odpowiednich formatów dla odpowiednich grafik: rysunki wektorowe, o ile jest to możliwe, należy konwertować do formatu PDF, zrzuty ekranowe najlepiej prezentują się w niestratnym formacie PNG, zaś fotografie o dużej ilości kolorów – w formacie JPEG, który stosuje kompresję stratną.

6. Pakiety rozszerzające możliwości \LaTeX -a

Powstało kilka rozbudowanych pakietów, pozwalających wzbogacić dokument PDF, np. pakiet **hyperref** o elementy hipertekstowe, pakiety **pdfscreen** oraz **pdfslide** – umożliwiając w miarę łatwe utworzenie prezentacji w PDF-ie.

6.1. hyperref

Pakiet **hyperref**, którego autorami są Sebastian Rahtz oraz Heiko Oberdiek, opisany w [9] oraz w [10] rozszerza funkcjonalność wszystkich hipertekstowych poleceń \LaTeX -a: spisu treści, przypisów, bibliografii itp. wstawiając polecenia `\special`, które, interpretowane przez sterowniki, zamieniane są na hipertekstowe odnośniki. Pakiet ten umożliwia także tworzenie własnych odnośników, np. do innych dokumentów lub do zasobów sieciowych.

hyperref może być skonfigurowany do pracy z wieloma sterownikami, zarówno obsługującymi polecenia `\special` z Hyper \TeX -a jak i z rozumiejącymi postscriptowe polecenia `\special` czy też prymitywy PDF \TeX -a. Obsługiwane sterowniki zostały wymienione poniżej:

hypertex – opcja dla sterowników zgodnych ze specyfikacją Hyper \TeX -a, np.: **xdvi**, **OzTeX**, **dvips** z opcją `-z`, oraz **Textures**,

`dvipdfm` – zapisuje `\special`-e rozumiane przez konwerter `dvipdfm`,
`dvips` – zapisuje `\special`-e rozumiane przez `dvips`,
`dvipsone` – zapisuje `\special`-e rozumiane przez `dvipsone`,
`dviwindo` – zapisuje `\special`-e interpretowane przez przeglądarkę DVI firmy Yandy&Yandy,
`pdftex` – zapisuje polecenia dla PDF_{TEX}-a,
`ps2pdf` – specjalny sterownik przeznaczony dla dokumentów, które będą przetwarzane przez Ghostscript w wersji wcześniejszej niż 5.21.
`vtex` – ta opcja zapisuje polecenia `\special` zrozumiałe dla systemu V_{TEX} firmy MicroPress.

Aby otrzymać poprawne PDF-y stosując sterownik `dvips` lub `dvipsone`, pliki postscriptowe muszą być destylowane przy użyciu Acrobat-a firmy Adobe.

Do współpracy z PDF_{TEX}-em pakiet ten wywołuje się umieszczając go w prambule dokumentu (`\usepackage[pdftex]{hyperref}`). Jako, że `hyperref` przeddefiniowuje wiele poleceń _{TEX}-owych, wskazane jest, aby wywoływany był jako ostatni – w przeciwnym razie może powodować błędy.

6.1.1. Konfiguracja i opcje pakietu

Wszystkie opcje pakietu można ustawić wykorzystując pary „klucz=wartość”, przy czym mogą być one ustawione jako argumenty polecenia `\usepackage` albo przy użyciu polecenia `\hypersetup`. Dodatkowo, przy ładowaniu pakietu, czytany jest plik `hyperref.cfg` (o ile taki istnieje) – w pliku tym można umieścić te opcje, które będą zawsze wykorzystywane i nie będą się specjalnie zmieniały. Dodatkowo, opcje dla konkretnego dokumentu mogą być ustalone w sposób następujący:

- wczytywane są opcje zawarte w pliku `hyperref.cfg`, np. autor dokumentu, wygląd hyperlinków, tryb wyświetlania:

```
\hypersetup{
  pdfauthor={Grzegorz Sapijaszko},
  colorlinks=true,
  pdfpagemode=FullScreen}
```

- globalne opcje klasy dokumentu przenoszone są na opcje pakietu `hyperref`:

```
\documentclass[draft]{mwart}
\usepackage{hyperref}
```

równoważne jest opcji:

```
\documentclass[draft]{mwart}
\usepackage[draft]{hyperref}
```

- opcje użyte w poleceniu `\usepackage` nadpisują opcje z pliku `hyperref.cfg`:

```
\usepackage[colorlink=false,
  pdftitle={Moj tytuł}]{hyperref}
```

Pierwsza grupa opcji, opisana w tab. 5, definiuje ogólne zachowanie się pakietu oraz wymiary dokumentu.

Kolejną grupę stanowią opcje konfiguracyjne wygląd i zachowanie się odsyłaczy hipertekstowych. Opcje te opisane są w tab. 6.

Opcje wybierające konkretny sterownik opisane są w tabeli 7. Stosując sterownik `dviwindo` może się okazać koniecznym przeddefiniowanie polecenia `\wwwbrowser` (domyślnie jest ono zdefiniowane dla programu Netscape jako `c:\netscape\netscape`), tak aby wskazywało ono na przeglądarkę, którą należy uruchomić. W przypadku Internet Explorer-a może to być:

```
\renewcommand{\wwwbrowser}{C:\string\Program\space
  Files\string\Plus!\string\Microsoft\space
  Internet\string\iexplore.exe}
```

Polecenie to można dodać do preambuły dokumentu lub zapisać w pliku `hyperref.cfg`.

Kolejna grupa opcji, wpływających głównie na wygląd różnorodnych odnośników w tekście, opisana jest w tab. 8.

opcja	typ	wartość domyślna	opis
draft	boolean	false	wszystkie opcje hipertekstowe zostają wyłączone
debug	boolean	false	w pliku log zapisywane są dodatkowe informacje diagnostyczne
a4paper	boolean	true	ustawia wymiary papieru na 210mm × 297mm
a5paper	boolean	false	ustawia wymiary papieru na 148mm × 210mm
b5paper	boolean	false	ustawia wymiary papieru na 176mm × 250mm
letterpaper	boolean	false	ustawia wymiary papieru na 8.5in × 11in
legalpaper	boolean	false	ustawia wymiary papieru na 8.5in × 14in
executivepaper	boolean	false	ustawia wymiary papieru na 7.25in × 10.5in

Tabela 5: Ogólne opcje pakietu hyperref

opcja	typ	wartość domyślna	opis
raiselinks	boolean	true	sterownik hypertext oblicza wysokość odsyłacza jako wysokość podstawowej linii tekstu, ta opcja przeddefiniowuje polecenie \special tak, aby uwzględniało ono rzeczywistą wysokość odsyłacza (odsyłacz taki może np. zawierać grafikę)
breaklinks	boolean	false	pozwala na łamanie odsyłaczy pomiędzy liniami. Jeżeli użyta jest sterownik pdftex, opcja ta przyjmuje wartość true – w takim przypadku odsyłacze łamane są pomiędzy liniami tworząc różne linki wskazujące na ten sam element docelowy.
pageanchor	boolean	true	określa tworzenie kotw dla każdej strony. Jeżeli opcja ta będzie miała wartość false, spis treści nie będzie zawierał hipertekstowych odnośników.
plainpages	boolean	true	wymusza nazywanie kotw dla poszczególnych stron zgodnie z numerem strony
nesting	boolean	false	pozwala na zagnieżdżanie odnośników; obecnie żaden sterownik nie wspomaga tego typu zagnieżdżeń

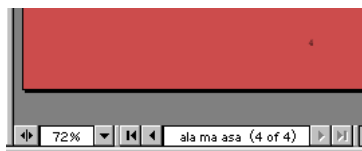
Tabela 6: Opcje wpływające na wygląd odsyłaczy hipertekstowych

opcja	typ	opis
pdftex	boolean	wybiera sterownik dla PDF _{TEX} -a
dvipdf	boolean	wybiera sterownik dla dvipdf
nativepdf	boolean	jest to alias dla opcji dvips
pdfmark	boolean	podobnie jak nativepdf jest to alias dla dvips
dvips	boolean	wybiera sterownik dla dvips
hypertext	boolean	ustawia hyperref do współpracy ze sterownikami zgodnymi z Hyper _{TEX} -em
dviwindo	boolean	wybiera sterownik dla przeglądarki dviwindo
dvipsone	boolean	wybiera sterownik dla dvipsone
vtex	boolean	wybiera sterownik dla programu V _{TEX} firmy MicroPress, przy czym tryb (HTML lub PDF) jest rozpoznawany automatycznie
latex2html	boolean	przeddefiniowuje makropolecenia tak, aby były kompatybilne z programem latex2html
ps2pdf	boolean	przeddefiniowuje polecenia tak aby były zgodne z Ghostscript-em

Tabela 7: Możliwe do wyboru sterowniki

opcja	typ	wartość domyślna	opis
extension	tekst		pozwała na zdefiniowanie rozszerzenia (np.: .dvi) jakie będzie dawane do nazwy pliku tworzącej odnośnik hipertekstowy przy użyciu pakietu <code>xr</code>
hyperfigures	boolean		
hyperfootnotes	boolean		
backref	boolean	false	dodaje odnośnik powrotny będący numerem rozdziału do każdej pozycji bibliografii; po kliknięciu takiego odnośnika wyświetlona zostanie strona, na której dana pozycja bibliograficzna została przywołana. Opcja ta działa tylko w przypadku, jeżeli po każdym poleceniu <code>\bibitem</code> zostanie wstawiona pusta linia
pagebackref	boolean	false	dodaje odnośnik powrotny będący numerem strony do każdej pozycji bibliografii
hyperindex	boolean	false	tworzy hipertekstowe odnośniki dla każdej pozycji skorowidza
colorlinks	boolean	false	koloruje tekst każdego odnośnika i elementu docelowego, przy czym kolor zależy od typu elementu: inny może być kolor odnośników cytowań, a inny adresów URL
linkcolor	kolor	red	kolor dla odnośników wewnątrz dokumentu; domyślnie jest to kolor czerwony
anchorcolor	kolor	black	kolor dla miejsc docelowych (kotw) wskazywanych przez odnośniki, domyślnie jest to kolor czarny
citecolor	kolor	green	kolor dla cytowań bibliografii, domyślnie jest to kolor zielony
filecolor	kolor	magenta	kolor dla adresów URL odnoszących się do lokalnych plików, domyślnie jest to kolor magenta
menucolor	kolor	red	kolor dla pozycji z menu Acrobat-a, domyślnie jest to kolor czerwony
pagecolor	kolor	red	kolor odsyłaczy do innych stron w dokumencie, domyślnie jest to kolor czerwony
urlcolor	kolor	cyan	kolor dla adresów internetowych, domyślnie jest to kolor cyan

Tabela 8: Opcje rozszerzające możliwości pakietu `hyperref`



Rysunek 13: Nazwa strony uzyskana przy pomocy polecenia `\thispdfpagelabel`

Opcja `pdfpagelabels` pozwala na numerowanie w dokumencie PDF poszczególnych stron zgodnie z ich numeracją logiczną (np. *ii*) niezależnie od ich fizycznej numeracji (np. str. 2 wstępu). Dodatkowo poleceniem `\thispdfpagelabel{}` można ustawić nazwę (numer) strony, np.:

```
\usepackage[pdfpagelabels=true]{hyperref}
...
\begin{document}
...
\thispdfpagelabel{ala ma asa}
\end{document}
```

co spowoduje utworzenie nazwy pokazanej na rys. 13.

Opcje specyficzne dla sposobu wyświetlania dokumentu PDF oraz tworzenia zakładek przedstawia tabela 9.

Należy podkreślić, że kolory obramowań podawane są jako trzy liczby z zakresu 0–1 jako kolor RGB. Nie można używać nazw kolorów zdefiniowanych w \TeX -u.

6.1.2. Informacje o dokumencie

Pakiet `hyperref` wprowadza też szereg opcji, przy pomocy których można zmieniać zawartość pól informacyjnych dokumentu. Opcje te, opisane w tab. 10, mogą być dodane jako opcjonalny argument polecenia `\usepackage`, nie mniej jednak \LaTeX może rozwijać te opcje przed wczytaniem ich przez pakiet. Z tego też względu lepszym sposobem na zdefiniowanie tych opcji jest użycie polecenia `\hypersetup`:

```
\usepackage{hyperref}
\hypersetup{
  pdftitle={Tworzenie dokumentów PDF przy użyciu LaTeX-a},
  pdfauthor={\textcopyright Grzegorz Sapijaszko}
  ...
}
```

Inną możliwość stwarza wykorzystanie danych wykorzystywanych w poleceniu `\maketitle` (zaczepnięte z [10]):

```
\newcommand\org@maketitle{}
\let\org@maketitle\maketitle
\def\maketitle{%
  \hypersetup{
    pdftitle={\@title},
    pdfauthor={\@author}
  }%
  \org@maketitle
}
```

W wersji 6.71r pakietu `hyperref` wprowadzono nową właściwość – dodanie `pdfusetitle=true` spowoduje przepisanie zawartości poleceń `\author{}` oraz `\title{}` do pól informacyjnych dokumentu PDF bez potrzeby wpisywania ich osobno (jako `pdftitle`, `pdfauthor`).

Preambuła przykładowego dokumentu \LaTeX owego, wykorzystującego pakiet `hyperref` może wyglądać następująco:

```
\documentclass[10pt]{mwart}
```

opcja	typ	wartość myślna	do-	opis
bookmarks	boolean	false		opcja ta określa czy mają być tworzone zakładki <i>bookmarks</i> , tworzenie zakładek wymaga dwukrotnego przebiegu $\text{PDF}\text{\LaTeX}$ -a. Czasami potrzebna jest ręczna ingerencja do pliku z tworzonymi zakładkami (plik .out) – aby $\text{PDF}\text{\LaTeX}$ nie nadpisywał tego pliku przy kolejnym przebiegu należy dopisać na początku tego pliku linijkę <code>\let\WriteBookmarks\relax</code> .
bookmarksnumbered	boolean	false		decyduje o dodawaniu numerów rozdziałów do zakładek
bookmarksopen	boolean	false		decyduje o tym, czy zakładki mają być rozwinięte czy nie
bookmarksopenlevel	liczba	\maxdimen		określa do jakiego poziomu zakładki mają być rozwinięte
bookmarkstype	nazwa	toc		określa rozszerzenie pliku z zakładkami jaki ma być przetwarzany
unicode	boolean	false		określa, czy tekst zakładek ma być kodowany zgodnie z <code>UNICODE</code> czy z <code>PDFDocEncoding</code>
pdfhighlight	nazwa	/I		określa zachowanie się przycisków: wartość <code>/I</code> stanowi skrót od <code>Inverse</code> (jest to wartość domyślna), pozostałe wartości to: <code>/N</code> (brak jakiegokolwiek efektu) , <code>/O</code> (outline) oraz <code>/P</code> (inset highlighting)
citebordercolor	kolor RGB	0 1 0		określa kolor obramowania wokół cytowania, domyślnie jest to kolor zielony
filebordercolor	kolor RGB	0 .5 .5		określa kolor obramowania wokół odnośników do plików, domyślnie jest to kolor teal
linkbordercolor	kolor RGB	1 0 0		określa kolor obramowania dookoła standardowych odnośników, domyślnie jest to kolor czerwony
menubordercolor	kolor RGB	1 0 0		określa kolor obramowania dookoła odnośników do poleceń menu Acrobat-a, domyślnie jest to kolor czerwony
pagebordercolor	kolor RGB	1 1 0		określa kolor obramowania wokół odnośników do stron, domyślnie jest to kolor żółty
urlbordercolor	kolor RGB	0 1 1		określa kolor obramowania dookoła odnośników do adresów <code>URL</code> , domyślnie jest to kolor cyan
pdfborder		0 0 1		określa styl ramki dookoła odnośnika, domyślnie jest to ramka o grubości linii równej 1 pt; należy zauważyć, że opcja <code>colorlinks</code> może zlikwidować to obramowanie

Tabela 9: Opcje specyficzne dla wyświetlania dokumentu

opcja	typ	wartość domyślna	opis
baseurl	URL		określa podstawowy URL dla dokumentu
pdfpagemode	tekst	None	określa, w jaki sposób dokument powinien być otworzony przez Acrobat-a; możliwe wartości to: None, UseThumbs – widoczne są miniatry, UseOutlines – widoczne są zakładki oraz FullScreen – dokument otwierany jest na całym ekranie. Jeżeli żaden tryb nie zostanie podany, a ustawiona będzie opcja bookmarks, dokument zostanie wyświetlony tak, jakby ustawione zostało UseOutlines
pdftitle	tekst		określa zawartość pola Title (tytuł dokumentu) widoczne w oknie General Info
pdfauthor	tekst		określa zawartość pola Author (autora dokumentu)
pdfsubject	tekst		określa zawartość pola Subject (temat)
pdfcreator	tekst		określa zawartość pola Creator
pdfproducer	tekst		określa zawartość pola Producer
pdfkeywords	tekst		określa zawartość pola Keywords
pdfview	tekst	FitBH	określa domyślny sposób wyświetlania dokumentu PDF po naciśnięciu linku, domyślnie szerokość wyświetlanego obszaru dopasowana jest do rozmiarów okna
pdfstartpage	tekst	1	określa na której stronie ma zostać otwarty dokument
pdfstartview	tekst	FitB	określa początkowy sposób wyświetlania dokumentu, domyślnie wyświetlany obszar dokumentu zostaje dopasowany do okna
pdfpagescrop	n n n n		określa domyślny obszar /CropBox dla stron w dokumencie, podany jako cztery liczby

Tabela 10: Opcje specyficzne dla wyświetlania dokumentu

```

\usepackage[OT4]{polski}
\usepackage[latin2]{inputenc}
\usepackage[pdftex]{color,graphicx}
\usepackage[pdftex,colorlinks=true,urlcolor=blue,pdfusetitle=true]{hyperref}
\hypersetup{
  baseurl={http://www.immt.pwr.wroc.pl/\%7sapi/},%
  pdfkeywords={pdf, latex, tex}
}

\title={Tworzenie dokumentów PDF przy pomocy LaTeX-a}
\author={Grzegorz Sapijaszko}

\begin{document}
...
\end{document}

```

Pakiet `hyperref` nie zezwala na zmianę informacji wstawianych do pól informujących o dacie utworzenia i modyfikacji pliku PDF. PDF_{TEX} wpisuje do tych pól bieżącą datę, nie mniej jednak przy pomocy polecenia `\pdfinfo` można wstawić dowolną wartość zarówno dla `/CreationDate` jak i `/ModDate`, co pokazuje poniższy przykład:

```

\pdfinfo{/CreationDate (D:19990901000000-01'00')}
\begin{group}
\def\twodigits#1{\ifnum#1<10 0\fi\the#1}%
\count0=\time \divide\count0 by 60
\edef\x{\twodigits{\count0}}%
\multiply\count0 by 60
\count1=\time \advance\count1 by -\count0
\edef\x{\x\twodigits{\count1}}%
\edef\x{/ModDate (D:\the\year
\twodigits\month \twodigits\day \x 00-01'00'))%

```

```
\expandafter\endgroup
\expandafter\pdfinfo\expandafter{\x}%
```

6.1.3. Polecenia zdefiniowane w pakiecie

Pakiet `hyperref` wprowadza dodatkowe makropolecenia, które mogą być użyte do wprowadzania hipertekstowych odnośników do adresów sieciowych (URL) lub do innych części dokumentu:

`\hyperbaseurl{URL}` – pozwala ustawić „podstawowy” adres URL, który będzie dodawany do innych podawanych URL-i,
`\href{URL}{text}` – obszar oznaczony jako `text` jest hipertekstowym odnośnikiem do adresu wpisanego jako URL, przy czym znaki specjalne `#` lub `~` występujące w adresach sieciowych powinny być podawane „w sposób normalny”, tzn. nie powinny być poprzedzone żadnym innym znakiem, np. `\`:

```
\href{http://www.immt.pwr.wroc.pl/~sapi/faq/
      email_faq.pdf#wstep}{Wstęp}
```

`\hyperimage{URL}` – wstawia rysunek podany w URL,
`\hyperdef{category}{name}{text}` – w dokumencie zaznaczony zostaje obszar oznaczony jako `text` oraz nadana mu zostaje nazwa `category.name`,
`\hyperref{URL}{category}{name}{text}` – wstawiany jest odnośnik hipertekstowy `text` do dokumentu podanego jako `URL#category.name`,
`\hyperlink{name}{text}` – oraz
`\hypertarget{name}{text}` – te dwa polecenia pozwalają na tworzenie wewnętrznych odnośników w dokumencie. Odnośniki `\hypertarget` są dostępne „z zewnątrz”, np. z dowolnego dokumentu HTML jako `URL#name`.

Ponadto pakiet wspomaga polecenia tworzące odnośniki hipertekstowe wprowadzane przez pakiety `html.sty` oraz `url.sty`, np. `\htmladdnormallink{}`, `\url{}`, które można wykorzystać w następujący sposób:

więcej informacji o \LaTeX -u można znaleźć pod adresem
`\url{http://www.gust.org.pl}`

co spowoduje utworzenie odnośnika hipertekstowego: „więcej informacji o \LaTeX -u można znaleźć pod adresem <http://www.gust.org.pl>”.

Pakiet `hyperref` pozwala także na dostęp do wielu funkcji menu programów `Acrobat Reader` lub `Exchange`, dzięki czemu możliwe jest tworzenie dokumentów interaktywnych. Do tego celu przewidziane jest polecenie:

```
\Acrobatmenu{opcja_menu}{tekst}
```

w którym `opcja_menu` to jedno z poleceń `Acrobat`-a, zaś `tekst` służy do stworzenia przycisku, którego naciśnięcie powoduje wykonanie polecenia menu.

Możliwe do wykorzystania polecenia oraz sposób ich dostępu wymienione są w tabeli 11, poniżej podany został przykład wykorzystania:

```
\documentclass[a4paper,12pt,oneside]{article}
...
\usepackage{fancyhdr}
\usepackage[pdftex,...]{hyperref}

\pagestyle{fancy} \cfoot{\BelkaNawigacyjna}

\newcommand{\BelkaNawigacyjna}{
  \Acrobatmenu{PrevPage}{poprzednia}~
  \Acrobatmenu{NextPage}{następna}~
  \Acrobatmenu{FirstPage}{pierwsza}~
  \Acrobatmenu{LastPage}{ostatnia}~
  \Acrobatmenu{GoBack}{cofnij}~
}
```

Fragment prezentacji

[poprzednia](#) [następna](#) [pierwsza](#) [ostatnia](#) [cofnij](#) [zakończ](#)

Rysunek 14: Przykład wykorzystania opcji pakietu `hyperref` do stworzenia interaktywnego menu w dokumencie PDF.

```
\Acrobatmenu{Quit}{zakończ}
}
```

który powoduje utworzenie interaktywnego menu na dole każdej strony podobnego do pokazanego na rysunku 14.

6.1.4. Tworzenie zakładek

Opcje umożliwiające tworzenie oraz wpływające na wygląd zakładek opisane są w tab. 9. Zakładki tworzone są automatycznie dla poszczególnych poleceń `\part`, `\chapter`, `\section`, ... o ile tylko opcja `bookmarks` nie jest wyłączona. Właściwości zakładek (ilość poziomów) mogą być kontrolowane przy pomocy dwóch liczników: `\tocdepth`, który definiuje, do którego poziomu rozdziały pojawiają się w spisie treści (i jednocześnie, w zakładkach) oraz `\secnumdepth`, który definiuje, do którego poziomu rozdziały są numerowane. Jeżeli używana jest opcja `bookmarksnumbered`, licznik ten używany jest również dla zakładek. Jak już wspomniano, zakładki zapisywane są w pliku `.out`, którego zawartość podobna jest do poniższej:

```
\BOOKMARK [1] [-]{section.1}{Wstep}{}
\BOOKMARK [1] [-]{section.2}{Droga do celu}{}
\BOOKMARK [1] [-]{section.3}{pdf z pomoca dvipdfm}{}
\BOOKMARK [2] [-]{subsection.3.1}{Specjalne polecenia}{section.3}
\BOOKMARK [2] [-]{subsection.3.2}{dvipdfm a fonty}{section.3}
...
\BOOKMARK [1] [-]{section.4}{Destylowanie plikow}{}
\BOOKMARK [2] [-]{subsection.4.1}{Adobe Acrobat}{section.4}
\BOOKMARK [2] [-]{subsection.4.2}{Ghostscript}{section.4}
...
```

Poszczególne zakładki tworzą strukturę drzewiastą, przy czym rozdziały pośrednie (dla przykładu: `\section` między `\chapter` a `\subsection`) nie powinny być pominięte, np. dla struktury przedstawionej na rys. 15, \LaTeX tworzy poprawny spis treści, ale zakładka dla rozdziału zatytułowanego „zły rozdział” (`\subsection`) umieszczona jest w złym miejscu (bezpośrednio pod zakładką dla `\chapter`).

Dla rozdziałów nienumerowanych (np. `\section*{}`) polecenie `\addcontentsline` powoduje dodanie takiego rozdziału do spisu treści a także powoduje utworzenie odpowiedniej zakładki. Mechanizm ten działa w standardowych klasach dokumentów (`book`, `article`, itd.). W klasach Marcina Wolińskiego (`mwbk`, `mwart`, itd.) rozdziały nienumerowane z założenia (tzn. bez potrzeby stosowania polecenia `\addcontentsline`) trafiają do spisu treści, pojawiają się też odpowiednie zakładki. Problem polega jednak na tym, że zakładki te odwołują się do niewłaściwych miejsc w dokumencie. Rozwiązaniem jest umieszczenie `\phantomsection` bezpośrednio przed rozdziałem „z gwiazdką”, dla przykładu:

```
\documentclass{mwart}
\usepackage[pdftex]{hyperref}

\begin{document}
```

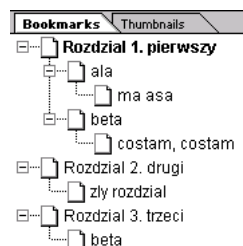
```

\chapter{pierwszy}
\section{ala}
\subsection{ma asa}
\section{beta}
\subsection{cośtam, cośtam}

\chapter{drugi}
\subsection{zły rozdział}

\chapter{trzeci}
\section{beta}

```



Rysunek 15: Niepoprawna struktura logiczna dokumentu powodująca złe umiejscowienie zakładki.

```

\section{ala}
\section{beta}
\subsection{podrozdział}
\subsection{jeszcze jeden}

\phantomsection
\subsection*{rozdział nienumerowany}
\end{document}

```

Do tworzenia zakładek można wykorzystać też polecenie `\pdfbookmark`:

```
\pdfbookmark[poziom]{tekst zakładki}{dowolny\_tekst}
```

Polecenie to tworzy znacznik o nazwie `dowolny_tekst` i dodaje zakładkę wskazującą na ten znacznik:

```

\pdfbookmark[0]{Strona tytułowa}{tytuł}
\pdfbookmark[1]{Wstęp}{wstęp}

```

Ponadto polecenie `\pdfbookmark` może wskazywać dowolne miejsce w dokumencie — wystarczy w określonym miejscu tekstu zdefiniować obszar do którego będzie odnosiła się zakładka, np. `\hypertarget{tutaj.1}{}{}`, a zakładkę zdefiniować następująco:

```

\begingroup
\makeatletter
\def\hyper@anchorstart #1\hyper@anchorend{%
\pdfbookmark[1]{Zakładka do okreslonego miejsca}{tutaj}%
\endgroup

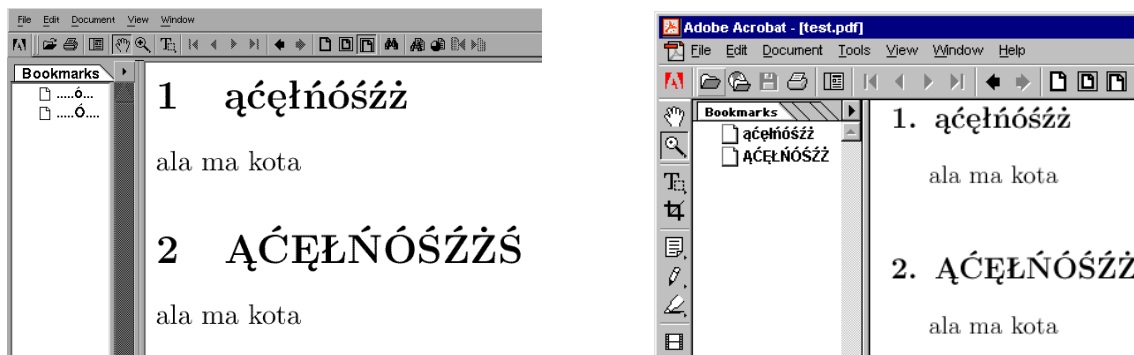
```

Poszczególnym poziomom rozdziałów odpowiadają poszczególne poziomy zakładki, przy czym jest to związane z klasą dokumentu:

	<code>\part</code>	<code>\chapter</code>	<code>\section</code>	<code>\subsection</code>	...
book, report	-1	0	1	2	...
article	0		1	2	...

Numer poziomu wprowadzonej ostatnim poleceniem zakładki zawarty jest w makropoleceniu `\Hy@currentbookmarklevel`. Właściwość tą wykorzystują polecenia `\currentpdfbookmark` oraz `\subpdfbookmark` pozwalające na wprowadzenie zakładek bez podania numeru poziomu jako argumentu; do określenia poziomu wykorzystują one tą przechowywaną wartość. Przy pomocy tych poleceń możliwe jest tworzenie zakładek tego samego lub niższego poziomu bez zwracania uwagi na numer poziomu.

Możliwości zakładek utworzonych przy pomocy pakietu `hyperref` ograniczają się do wskazywania określonego miejsca wewnątrz dokumentu. Generalnie jednak, zakładka może wskazywać inny dokument, wywoływać funkcję `Acrobat Reader`a, odtwarzać dźwięk, itp. Do tego celu można użyć PDF \TeX -owego polecenia `\pdfoutline` (opisanego na str. 19). Standardowo, znaki w treści zakładek muszą być zgodne z kodowaniem PDFDocEncoding, nie zawierającym polskich znaków diakrytycznych. Stosując opcję `unicode=true` przy wywoływaniu pakietu można uzyskać polskie



Rysunek 16: Sposób wyświetlania zakładek uzyskanych przy pomocy opcji `unicode=true` w programie Acrobat w systemie HPUX (po lewej) oraz w środowisku MS Windows (po prawej).

znaki w zakładkach, przy czym należy zwrócić uwagę, iż będą one poprawnie wyświetlane jedynie przez program Acrobat w środowisku MS Windows (rys. 16)².

6.1.5. Formularze

menu	opcja menu i opis
File	Open (otwiera okno <i>Otwórz</i>), Close (zamyka dokument), Scan (otwiera okno <i>Adobe Acrobat Scan</i>), Save (otwiera okno <i>Zapisz</i>), SaveAs (otwiera okno <i>Zapisz jako</i>), Optimizer:SaveAsOpt (otwiera okno pozwalające na optymalizację dokumentów – <i>Select Folder To Process</i>), Print (<i>Drukuj</i>), PageSetup (pozwala zmienić <i>Ustawienia wydruku</i>), Quit (<i>Wyjdź</i>)
File, Import	ImportImage (otwiera okno pozwalające zaimportować rysunek – <i>Importuj image</i>), ImportNotes (pozwala <i>zaimportować notatki</i>), AcroForm:ImportFDF (pozwala <i>zaimportować formularze</i>)
File, Export	ExportNotes (otwiera okno do <i>eksportu notatek</i>), AcroForm:ExportFDF (otwiera okno <i>eksportu formularzy</i>)
File, DocumentInfo	GeneralInfo (otwiera okno zawierające ogólne informacje o dokumencie – <i>General Info</i>), OpenInfo (otwiera okno <i>Open Info</i>), FontsInfo (otwiera okno z <i>informacjami o fontach</i>), SecurityInfo (otwiera okno z <i>informacjami o zabezpieczeniach</i>), Weblink:Base (otwiera okno z <i>bazowym adresem URL</i>), AutoIndex:DocInfo (otwiera okno umożliwiające podłączenie pliku z <i>indeksami</i>)
File, Preferences	GeneralPrefs (otwiera okno <i>General Preferences</i>), NotePrefs (otwiera okno <i>Annotation Preferences</i>), FullScreenPrefs (otwiera okno <i>Full Screen Preferences</i>), Weblink:Prefs (otwiera okno <i>Weblink Preferences</i>), AcroSearch:Preferences ((dla Windows) lub AcroSearch:Prefs (dla Mac) otwiera okno <i>Acrobat Search Preferences</i>), Cpt:Capture (otwiera okno <i>Acrobat Paper Capture Preferences</i>)
Edit	Undo (cofa ostatnią zmianę), Cut (wycina zaznaczony fragment), Copy (kopiuje zaznaczony fragment), Paste (wstawia zawartość schowka), Clear (usuwa zawartość ...), SelectAll (pozwala zaznaczyć wszystko), Ole:CopyFile (kopiuje aktualny plik do schowka), TouchUp:TextAttributes (zmienia atrybuty tekstu), TouchUp:FitTextToSelection (dopasowuje tekst do zaznaczonego obszaru), TouchUp:ShowLineMarkers (wyświetla znaczniki linii tekstu), TouchUp:ShowCaptureSuspects (wyświetla proponowane słowo przy OCR-owaniu tekstu), TouchUp:FindSuspect (odpowiada menu Tools, Paper Capture <i>Find First Suspect</i>), Properties (otwiera okno właściwości dla różnych elementów: zakładek, łącz itp. – <i>Properties</i>)
Edit, Search	AcroSrch:Query (otwiera okno <i>Adobe Acrobat Search</i>), AcroSrch:Indexes (otwiera okno <i>Index Selection</i>), AcroSrch:Results (wyświetla <i>rezultaty wyszukiwania</i>), AcroSrch:Assist (otwiera okno <i>Word Assistant</i>), AcroSrch:PrevDoc (pozwala na przejście do <i>poprzedniego dokumentu</i>), AcroSrch:PrevHit (przechodzi do miejsca <i>poprzedniego wystąpienia</i>), AcroSrch:NextHit (przechodzi do miejsca <i>następnego wystąpienia</i>), AcroSrch:NextDoc (przechodzi do <i>następnego dokumentu</i>)

² dzięki uprzejmości Wojtka Myszk

menu	opcja menu i opis
Document	AcroForm:Actions (otwiera okno dostępne z menu <i>Set Page Action...</i>), CropPages (pozwala na ustawienie /CropBox <i>dla poszczególnych stron</i>), RotatePages (pozwala na <i>obrót stron</i>), InsertPages (pozwala <i>wstawić strony</i>), ExtractPages (pozwala <i>wyeksportować strony</i>), ReplacePages (pozwala <i>zastąpić strony</i>), DeletePages (pozwala <i>usunąć strony</i>), NewBookmark (tworzy <i>nową zakładkę</i>), SetBookmarkDest (pozwala ustalić obszar docelowy <i>dla zakładki</i>), CreateAllThumbs (tworzy <i>ministry</i>), DeleteAllThumbs (<i>usuwa ministry</i>)
Tools, Paper Capture	Cpt:CapturePages (otwiera okno <i>Acrobat paper Capture</i>)
Tools, Forms, Fields	AcroForm:Duplicate (pozwala na <i>skopiowanie pola</i>), AcroForm:TabOrder (pozwala na ustawienie kolejności przechodzenia między <i>polami</i>)
Tools	Hand (wybiera narzędzie z paska narzędziowego – <i>hand</i>), ZoomIn (wybiera narzędzie – <i>lupę powiększającą</i>), ZoomOut (wybiera narzędzie – <i>lupę pomniejszającą</i>), SelectText (wybiera narzędzie do <i>zaznaczania tekstu</i>), SelectGraphics (wybiera narzędzie do <i>zaznaczania grafiki</i>), Note (pozwala wstawić <i>notatkę</i>), Link (pozwala wstawić <i>łącze</i>), Thread (wybiera narzędzie wstawiania <i>wątku</i>), AcroForm:Tool (wybiera narzędzie tworzenia <i>formularzy</i>), Acro_Movie:MoviePlayer (wybiera narzędzie do wstawiania <i>filmu</i>), TouchUp:TextTool (wybiera narzędzie <i>tekstu</i>), Find (otwiera okno dialogowe <i>Find – Znajdź</i>), FindAgain (odpowiada poleceniu <i>Znajdź ponownie</i>), CreateNotesFile (odpowiada poleceniu <i>Summarize Annotations</i>)
View	ActualSize (wyświetla dokument w <i>normalnej skali</i>), FitVisible (dopasowuje wyświetlanie dokumentu do szerokości <i>obszaru widzialnego</i>), FitWidth (dopasowuje wyświetlanie dokumentu do <i>szerokości strony</i>), FitPage (dopasowuje wyświetlanie dokumentu do <i>całej strony</i>), ZoomTo (wywołuje okno powiększenia – <i>Zoom To</i>), FullScreen (wyświetla dokument na całym ekranie – <i>prezentacja</i>), FirstPage (przechodzi do <i>pierwszej strony</i>), PrevPage (przechodzi do <i>poprzedniej strony</i>), NextPage (przechodzi do <i>następnej strony</i>), LastPage (przechodzi do <i>ostatniej strony</i>), GoToPage (wywołuje okno umożliwiające wybranie <i>numeru strony</i>), GoBack (przechodzi do <i>poprzedniego widoku</i>), GoForward (przechodzi do <i>następnego widoku</i>), SinglePage (powoduje wyświetlenie w trybie <i>pojedynczych stron</i>), OneColumn (powoduje wyświetlanie w trybie <i>jednej kolumny</i>), TwoColumns (powoduje wyświetlanie w trybie <i>dwóch kolumn</i>), ArticleThreads (otwiera okno <i>z artykułami</i>), PageOnly (wyświetla tylko stronę, <i>bez zakładek</i>), ShowBookmarks (otwiera obszar <i>z zakładkami</i>), ShowThumbs (wyświetla <i>ministry</i>)
Window	ShowHideToolBar (włącza i wyłącza <i>pasek narzędzi</i>), ShowHideMenuBar (włącza i wyłącza <i>menu</i>), ShowHideClipboard (włącza i wyłącza <i>podgląd schowka</i>), Cascade (wyświetla wszystkie otwarte dokumenty <i>kaskadowo</i>), TileHorizontal (wyświetla dokumenty układając je <i>poziomo</i>), TileVertical (wyświetla dokumenty układając je <i>pionowo</i>), CloseAll (powoduje <i>zamknięcie wszystkich dokumentów</i>)
Help	HelpUserGuide (otwiera <i>podręcznik użytkownika</i>), HelpTutorial (otwiera podręcznik „ <i>Acrobat 4.0 Tour</i> ”), HelpExchange (otwiera <i>podręcznik użytkownika</i>), HelpScan (otwiera <i>podręcznik użytkownika</i>), HelpCapture (otwiera <i>podręcznik użytkownika</i>), HelpPDFWriter (otwiera <i>podręcznik użytkownika</i>), HelpDistiller (otwiera <i>podręcznik użytkownika</i>), HelpSearch (otwiera <i>podręcznik użytkownika</i>), HelpCatalog (otwiera <i>podręcznik użytkownika</i>),
Help(Windows)	About (otwiera okno z informacjami <i>o programie</i>)

Tabela 11: Polecenia dostępne z menu programu Acrobat przy pomocy polecenia \Acrobatmenu.

6.2. pdfcrypt

Kolejny pakiet autorstwa Heiko Oberdieka umożliwiający ustawienie zabezpieczeń dokumentu PDF. Pakiet współpracuje z PDF_TE_X-em w wersjach pdfTeX-1.00a, pdfTeX-0.14h-pdfcrypt-20010310 lub V_TE_X-em 7.35.

6.2.1. Konfiguracja i opcje pakietu

Podobnie jak w pakiecie hyperref wszystkie opcje można ustawić jako pary „klucz=wartość”, przy czym mogą być one ustawione jako argumenty polecenia \usepackage albo przy użyciu polecenia \pdfcryptsetup, np.:

```
\pdfcryptsetup{
  owner=hasło_twórcy,
```

opcja	opis
owner	hasło twórcy – ciąg znaków, który będzie służył twórcy dokumentu PDF do otworzenia go w przeglądarce
user	hasło użytkownika – ciąg znaków, który będzie służył użytkownikom do otwarcia dokumentu PDF
print	wartość true lub false, zezwalająca lub nie na drukowanie dokumentu
copy	true lub false, zezwalająca lub nie na zaznaczanie tekstu lub grafiki i przenoszenie ich poprzez schowek do innych aplikacji
edit	true lub false, zezwalająca lub nie na wprowadzanie zmian do dokumentu
annotate	true lub false, zezwalająca lub nie na dodawanie komentarzy i zakładek (w programie Adobe Acrobat)
all, none	
debug	

Tabela 12: Opcje pakietu pdfcrypt

```

user=hasło_użytkownika,
print=true, copy=false
}

```

co spowoduje, że do otwarcia pliku PDF w przeglądarce wymagane będzie podanie jednego z haseł: albo hasła twórcy pliku, albo hasła użytkownika. Ponadto możliwe będzie drukowanie pliku, nie możliwe zaś zaznaczanie tekstu bądź grafiki. Wszystkie opcje pakietu przedstawione są w tabeli 12

Domyślnie szyfrowanie jest włączane przez polecenie `\begin{document}` nie mniej jednak może być włączone wcześniej poprzez dodanie polecenia `set`, np.:

```

\usepackage[owner=ala,user=ola,print=false]{pdfcrypt}
...
\begin{document}

```

lub

```

\usepackage{pdfcrypt}
\pdfcryptsetup{owner=ala,user=ola}
\pdfcryptsetup{print=false,set}
...
\begin{document}

```

Znaki `{`, `}` oraz `\` muszą być poprzedzone znakiem `\`, ponadto ciąg zawierający jeden z tych znaków nie może być podany jako opcja pakietu a jedynie w poleceniu `\pdfcryptsetup`, np.:

```

\pdfcryptsetup{user={\{\haslo\}},print=false}

```

dzięki czemu ustawione hasło to `{\haslo}`.

Należy zwrócić uwagę na fakt, że niektóre pakiety, np. `thumbpdf` wysyłają na wyjście pewne obiekty przy pomocy polecenia `\immediate\pdfobj{...}` – w takim przypadku polecenie `\pdfcrypt` zostanie zignorowane bez ostrzeżenia lub komunikatu o błędzie. Dlatego też pakiet `pdfcrypt` powinien być wywołany wcześniej oraz powinna być użyta opcja `set`, np.:

```

\usepackage{pdfcrypt}
\pdfcryptsetup{..., set}
...
\usepackage{thumbpdf}

```

6.3. pdfpages

Pakiet `pdfpages` umożliwia włącznie do tworzonego pliku PDF stron z innego, istniejącego już, pliku PDF. Co prawda podobny efekt można uzyskać wykorzystując polecenie `\includegraphics`

opcja	opis
<code>pages</code>	należy podać strony, które mają być włączone, przy czym argumentem mogą być strony bądź zakresy stron oddzielone przecinkami, np. <code>pages=2,4-10,13</code> . Aby wstawić pustą stronę należy użyć zapisu <code>{}</code> .
<code>nup</code>	umożliwia wstawienie kilku stron na jednej stronie papieru. Dopuszczalne wartości to 1, 2 oraz 4; domyślną wartością jest 1.
<code>landscape</code>	nadanie tej opcji wartości <code>true</code> spowoduje, iż wstawiane strony będą w układzie poziomym; domyślną wartością jest <code>landscape=false</code> .
<code>turn</code>	nadanie tej opcji wartości <code>false</code> spowoduje, iż strony nie będą wyświetlane w układzie poziomym.
<code>openright</code>	przyjmuje wartości <code>true</code> lub <code>false</code> . Powinno być używane łącznie z opcją <code>nup=2</code> lub <code>nup=4</code> ; przy wartości <code>true</code> powoduje, iż pierwsza strona umieszczana jest po prawej stronie.
<code>frame</code>	nadanie tej opcji wartości <code>true</code> powoduje wyświetlenie ramki dookoła każdej wstawianej strony.
<code>deltax</code>	dodaje dodatkowy poziomy odstęp pomiędzy stronami ułożonymi na jednej kartce papieru; domyślnie jest <code>deltax=0</code> .
<code>deltay</code>	dodaje dodatkowy pionowy odstęp pomiędzy stronami ułożonymi na jednej kartce papieru; domyślnie jest <code>deltay=0</code> .
<code>offset</code>	przesuwa włączane strony w kierunkach poziomym i pionowym; domyślne wartości to <code>offset=0</code> .
<code>pagecommand</code>	pozwalia na zdefiniowanie polecenia \LaTeX -a, które jest wykonywane dla każdej tworzonej kartki papieru; domyślnie jest to: <code>pagecommand={\thispagestyle{empty}}</code> .

Tabela 13: Opcje polecenia `\includepdf`

z pakietu `graphics`, nie mniej jednak wydaje się, że wygodniejsze jest korzystanie z pakietu `pdfpages`.

Użycie pakietu jest standardowe, w preambule dokumentu wystarczy dodać:

```
\usepackage[opcje]{pdfpages}
```

przy czym jako opcje można podać `final`, dzięki czemu strony będą włączane (jest to opcja domyślna), lub `draft` – strony nie będą włączane, drukowane jest jedynie ramka dookoła dodawanej strony i nazwa pliku.

Poszczególne strony oraz ich układ włączane są przy pomocy polecenia:

```
\includepdf[opcja=wartość]{plik.pdf}
```

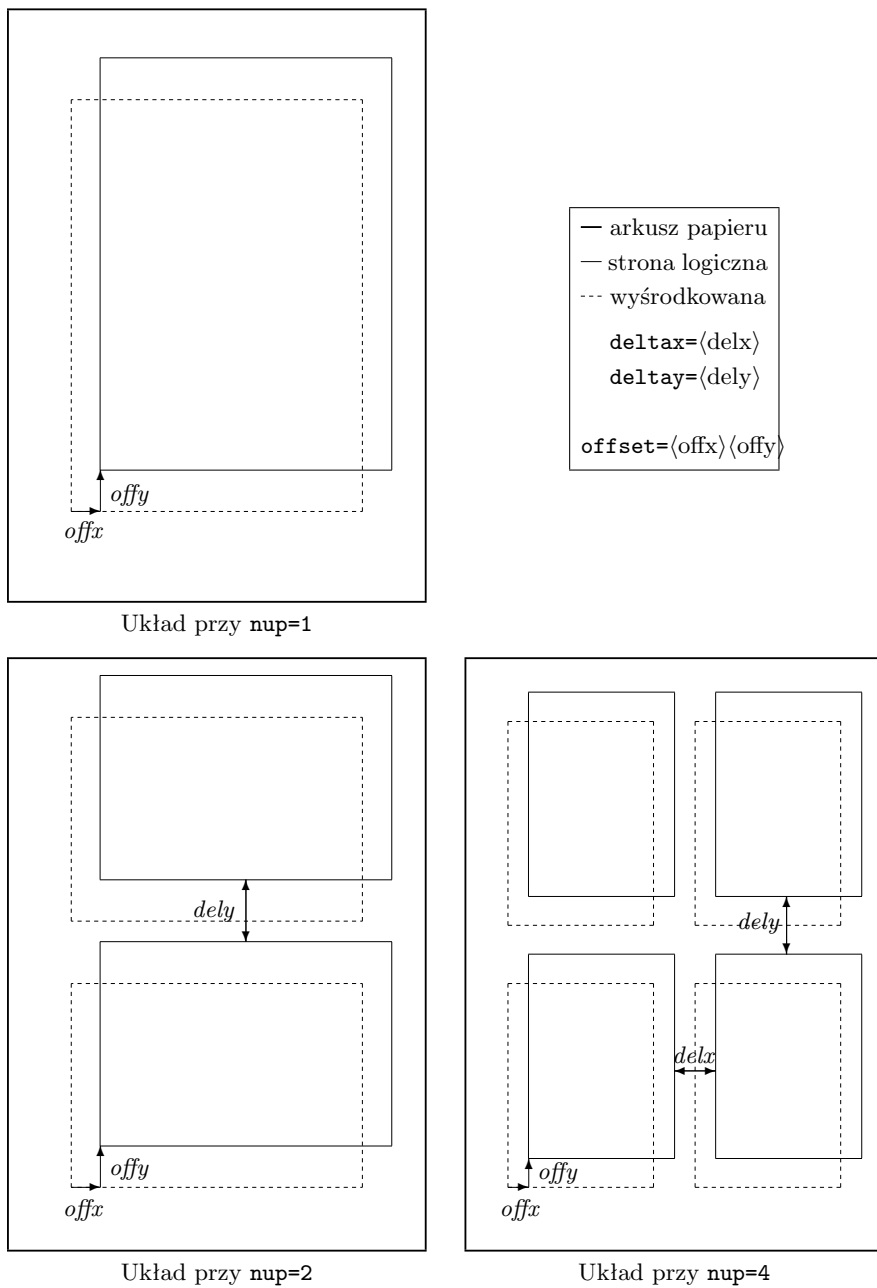
zaś dostępne opcje opisane są w tabeli 13.

Układ kilku stron logicznych na jednym arkuszu papieru przedstawia rysunek 17.

7. Prezentacje w PDF-ie

Być może format PDF nie został stworzony do tworzenia prezentacji, nie mniej jednak, przy małym nakładzie pracy, można wykorzystać go do utworzenia pełnoekranowej prezentacji jakością nie ustępującej prezentacjom wykonanym przy użyciu Power Point-a. Prezentacje mogą być wykonane w oparciu o dowolny pakiet wspomagający ich tworzenie: począwszy od `foils`, poprzez `seminar`, `prosper` a skończywszy na `pdfslide` czy `pdfscreen`. Prezentację taką można wzbogacić o efektowne przejścia między poszczególnymi stronami, a także (w ograniczonym zakresie) o efekty dźwiękowe i filmy wideo. Istnieje również postprocesor dla plików PDF o wdzięcznej nazwie `PPoint4`, dzięki któremu można uzyskać zaskakujące efekty wizualne.

Format PDF przewiduje m.in. możliwość tworzenia tła strony, definiuje różnorodne atrybuty strony oraz przejścia pomiędzy stronami w czasie ich wyświetlania.



Rysunek 17: Układ kilku stron logicznych na arkuszu papieru przy różnych opcjach pakietu `pdfpages`

7.1. Tło

Format PDF pozwala na dodanie tła do każdej ze stron w pliku, przy czym tło może być jednolite, może być zdefiniowane jako funkcja przejść pomiędzy danymi kolorami, za tło może wreszcie posłużyć rysunek.

PDF_{TEX} nie umożliwia wykorzystania „przezroczystości” w plikach graficznych typu PNG, tzn. obrazek zostanie wstawiony, ale obszar „przezroczysty” zostanie zastąpiony kolorem białym. Można się pokusić o zabawę z zarysem selekcji lub maskami ale wydaje się to zagadnieniem zbyt skomplikowanym. Można też obszar „przezroczysty” zamienić na kolor tła strony – dzięki czemu uzyskamy podobny efekt.

Przykład ustawienia jednolitego tła w kolorze szarym `[gray]{0.9}` dla danej strony przedstawiony jest poniżej.

```
\EveryShipout{%
  \setbox255=\vbox{%
    \vbox to 0pt{%
      \vss
      \moveright -1in \hbox{%
        \let\protect\@typeset@protect
        \color[gray]{0.9}%
        \vrule width \paperwidth height \paperheight
      }%
      \kern 1in
      \kern -\paperheight
    }%
    \nointerlineskip
  \box255
}%
}
```

Można skorzystać też z polecenia `\pdfliteral`:

```
\pdfliteral direct{%
  q % gsave
  .8 .85 1 rg .8 .85 1 RG %% <-- kolor
  n % newpath
  0 0 597 845 re % .. .. wysokosc/szerokosc strony
  f % fill
  Q% grestore
}%
```

7.2. Atrybuty strony

Polecenia PDF_{TEX}-a `\pdfpagesattr={}` oraz `\pdfpageattr={}` pozwala na definicję parametrów odpowiednio dla wszystkich lub wybranych stron dokumentu. Możliwe atrybuty podane są w tabeli 14.

Dla przykładu, polecenie zmieniające wygląd strony może wyglądać następująco:

```
\pdfpageattr={
  /Trans <<
    /Type /Trans
    /S /Dissolve
    /D 1
  >>
  /Dur 1.5
}
```

opcja	opis
/MediaBox [llx lly urx ury]	obszar określający naturalny rozmiar strony, przy czym dwie pierwsze współrzędne (podane w punktach) oznaczają dolny lewy róg, dwie następne – górny prawy róg, np.: dla formatu A4 będzie to [0 0 596 842]
/CropBox [llx lly urx ury]	obszar określający wymiary strony jaka będzie wyświetlana lub drukowana
/BleedBox [llx lly urx ury]	opcja ta określa obszar, do jakiego ma być przycięta strona przez aplikację przygotowania do druku (prepress)
/TrimBox [llx lly urx ury]	określa obszar jaki pozostanie po wydrukowaniu i przycięciu
/ArtBox [llx lly urx ury]	obszar ten używany jest podczas wstawiania pliku PDF do innych aplikacji
/Rotate liczba całkowita	kąt o jaki będzie obrócona strona, musi być wielokrotnością 90 stopni (domyślnie 0 stopni), przy czym obrót dokonywany jest zgodnie z kierunkiem wskazówek zegara
/Dur liczba rzeczywista	np. /Dur 1.5 – określa czas (w sekundach) przez jaki dana strona będzie wyświetlana w prezentacji pełnoekranowej (z menu View, Full screen), w innym trybie opcja ta nie działa
/Hid	skrót od Hidden – strona będzie schowana (nie będzie wyświetlana) podczas prezentacji
/Trans	opcja ta pozwala zdefiniować rodzaj przejścia pomiędzy stronami w prezentacji

Tabela 14: Parametry wpływające na wygląd strony

rodzaj	parametry	opis
/Split	/Dm, /M	Najprostsze przejście, w którym wykorzystano efekt kurtyny, tzn. dwie linie przesuwają się po ekranie otwierając bądź zamykając stronę
/Blinds	/Dm	Podobne do poprzedniego, z tym, że strony zmieniają się na zasadzie żaluzji
/Box	/M	Prostokąt rozchodzący się z/do środka strony
/Wipe	/Di	
/Dissolve	brak	efekt podobny do „krystalizacji”
/Glitter	/Di	Podobnie jak w /Dissolve, przy czym efekt przesuwa się od jednego rogu do przeciwległego
/R	brak	Replace – podstawowy efekt; jedna strona jest zastępowana następną bez żadnego widocznego przejścia.

Tabela 15: Efektowne przejścia pomiędzy stronami

7.2.1. Przejścia między stronami w prezentacji

Format PDF definiuje kilka zachowań, jakie można uzyskać w czasie przechodzenia pomiędzy poszczególnymi stronami w trakcie prezentacji pełnoekranowej. Poszczególne rodzaje przejść wraz z opcjami zostały opisane w tabeli 15.

Dodatkowo, każdemu z tych przejść można ustawić jeden z parametrów wyszczególnionych w tabeli 16.

Ponadto, można wpływać na czas trwania przejścia, wykorzystując w tym celu parametr /D. Przykładowe przejście podane jest poniżej:

```
\pdfpageattr={
  /Trans <<
    /Type /Trans
    /S /Box
    /M /0
    /D 1
  >>
}
```

W dalszej części opisano kilka pakietów utworzonych specjalnie do tworzenia pełnoekranowych prezentacji.

opcja	opis
/Di	kierunek ruchu, wyrażony w stopniach, liczony w kierunku przeciwnym do kierunku obrotu wskazówek zegara. Wartość powinna być wielokrotnością 90 stopni.
/Dm	określa kierunek dla przejść poziomych i pionowych; możliwe wartości to /H – poziomo lub /V – pionowo
/M	określa kierunek „do środka” (/I) lub „na zewnątrz” (/O)

Tabela 16: Parametry przejść

7.3. pdfscreen

Pakiet ten, opracowany przez C. V. Radhakrishnana, pozwala na tworzenie dokumentów, które mogą być kompilowane do postaci wygodnej przy czytaniu z ekranu monitora oraz, po zmianie kilku opcji, do drukowania. Ekran monitora ma inną proporcję obrazu niż standardowa kartka papieru – m.in. z tego też względu dokumenty przygotowywane do czytania „na ekranie” powinny różnić się od dokumentów przygotowywanych do druku. Takie dokumenty mogą być też wzbogacone o elementy nawigacyjne przeznaczone do poruszania się po dokumencie.

Przykład dokumentu przygotowanego przy pomocy pakietu **pdfscreen** przedstawia rys. 4; oryginalny dokument, wraz z odpowiednikiem przygotowanym do druku, znajduje się pod adresem <http://www.immt.pwr.wroc.pl/~sapi/faq/>.

Pakietu używa się dopisując do preambuły tworzonego dokumentu

```
\usepackage[screen,...opcje...,polish]{pdfscreen}
```

przy czym nie trzeba już ładować pakietu **hyperref**, gdyż **pdfscreen** go wczytuje. Jeżeli istnieje jednak potrzeba zmiany lub dopisania jakichś opcji dla pakietu **hyperref**, należy go umieścić przed pakietem **pdfscreen**.

7.3.1. Opcje pakietu oraz plik konfiguracyjny

Dostępne są opcje wymienione w tabeli 17. Jedną z opcji jest `[polish]`, dzięki której generowane są napisy na przyciskach. Należy jednak zwrócić uwagę, że treść przycisku `\panelfullscreenname` nie została przetłumaczona, a domyślnie jest **Full Screen**. Można w prosty sposób usunąć tę niedogodność albo zmieniając treść deklaracji wewnątrz pliku ze stylem, albo przeddefiniowując polecenie np.:

```
\renewcommand{\panelfullscreenname}{Prezentacja}
```

Ponadto pakiet wprowadza kilka nowych poleceń, które mogą być wykorzystane do nadania dokumentowi indywidualnych cech. Poleceniem

```
\emblema{plik_graficzny}
```

można zdefiniować nazwę pliku graficznego, który będzie umieszczony na panelu. Aby uczynić ten plik aktywnym linkiem do strony internetowej wystarczy użyć polecenia

```
\urlid{http://adres.internetowy}
```

Polecenie

```
\screensize{wysokość}{szerokość}
```

powoduje ustalenie wymiarów tworzonego dokumentu PDF. Domyślnie nie są ustawione żadne wymiary i dlatego też powinno być to ustalone przez użytkownika, przy czym nie ma żadnych ograniczeń co do wymiarów. Należy pamiętać, że domyślna szerokość panelu ustalona jest na 15% szerokości strony. Szerokość dodatkowych marginesów można ustawić poleceniem

```
\marginwidth{lewy}{prawy}{górny}{dolny}
```

Domyślnie wymiary wszystkich marginesów wynoszą zero.

Przykładowy plik przedstawiony jest poniżej, zaś na rys. 18 wygląd tego dokumentu.

opcja	opis
screen	tworzy wersję do czytania z ekranu
print	tworzy wersję do wydrukowania
panelleft lub leftpanel	umieszcza panel nawigacyjny po lewej stronie
panelright lub rightpanel	umieszcza panel nawigacyjny po prawej stronie
nopanel	nie umieszcza panelu w dokumencie
paneltoc	powoduje umieszczenie spisu treści na panelu nawigacyjnym, przy czym do spisu treści brane są tylko tytuły rozdziałów pierwszego poziomu (dla klasy artykuł – tytuły rozdz. section); stosując tę opcję nie należy umieszczać polecenia <code>\tableofcontents</code> w dokumencie
sectionbreak	powoduje złamanie strony przed nagłówkiem rozdziału
code	udostępnia środowisko <code>decl</code> i polecenia umożliwiające wstawianie kodu źródłowego \LaTeX -a, podobnie jak ma to miejsce w przypadku środowiska <code>verbatim</code>
dvips, dvipsone, dvipdf, dvipdfm, dviwindo, pdftex, ps2pdf, vtex	jako opcję można podać jaki driver powinien być użyty przy kompilacji dokumentu; domyślnie jest to <code>pdftex</code>
blue, bluelace, chocolate, gray, orange, palegreen	opcja pozwalająca wybrać jeden ze schematów kolorów, w jakich będą tworzone panel i przyciski, domyślnie jest to <code>blue</code>
polish	opcja spowoduje utworzenie polskich napisów na przyciskach; dostępnych jest kilka nazwie języków europejskich
nocfg	spowoduje pominięcie wczytywania pliku konfiguracyjnego przy ładowaniu pakietu

Tabela 17: Opcje pakietu `pdfscreen`

```

\documentclass[12pt]{mwart}
\usepackage[T1]{polski}
\usepackage[latin2]{inputenc}
\usepackage{lucidaso}
\usepackage[screen,panelright,polish]{pdfscreen}
\screensize{6in}{9in}
\panelwidth=2in

\emblem{../images/mailbox.png}
\urlid{www.immt.pwr.wroc.pl/~sapi}

\begin{document}
\section{Rozdział 1}
tutaj zawartość rozdziału
\section{Rozdział 2}
tutaj zawartość rozdziału
...
\end{document}

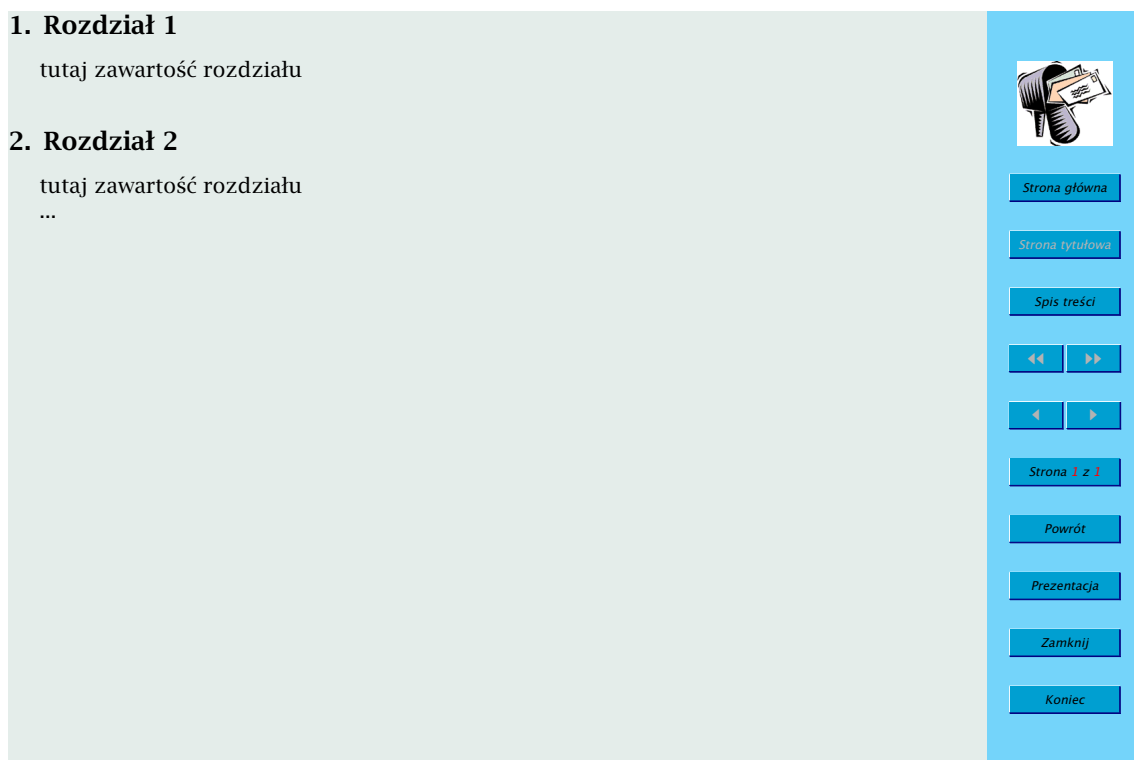
```

Pakiet `pdfscreen` wczytuje również zawartość pliku konfiguracyjnego `pdfscreen.cfg` o ile taki plik istnieje. W pliku tym można zdefiniować własne kolory dla poszczególnych elementów prezentacji, nazwy przycisków, URL, logo itp. Zalecane jest korzystanie z pliku konfiguracyjnego – dzięki temu można wprowadzić globalne zmiany wpływające na wygląd wszystkich dokumentów tworzonych z pomocą pakietu `pdfscreen` (nadać im indywidualny styl), a także preambuła tworzonego dokumentu staje się bardziej czytelna.

7.3.2. Panel nawigacyjny

Zawartość panelu oraz jego wygląd każdy użytkownik może zdefiniować według własnych potrzeb. Do tego celu przewidziano kilka poleceń:

```
\addButton{długość}{tekst przycisku}
```



Rysunek 18: Przykład dokumentu uzyskanego przy pomocy pakietu pdfscreen

Następna strona

Rysunek 19: Przycisk utworzony przy pomocy \addButton

spowoduje utworzenie przycisku o długości określonej jako **długość**, który będzie zawierał tekst **tekst przycisku**. Tak zdefiniowany przycisk (rys. 19) można „podpiąć” pod menu Acrobat-a przy pomocy polecenia \Acrobatmenu, np.:

```
\Acrobatmenu{NextPage}{\addButton{24mm}{Następna strona}}
```

dzięki czemu możliwa staje się nawigacja wewnątrz dokumentu. W podobny sposób można utworzyć przycisk zawierający mapę bitową:

```
\imageButton{szerokość}{wysokość}{plik graficzny}
```

który następnie można połączyć z hipertekstowym odnośnikiem:

```
\href{http://www.gust.org.pl}{\imageButton{szer}{wys}{plik}}
```

Szerokość panelu nawigacyjnego może być zmieniona poleceniem \panelwidth=szerokość. Domyślnie jest to 15% szerokości okna lub 1 cal w przypadku, gdy 15% szerokości wynosi mniej niż 1 cal. Także zawartość panelu może być zmieniona i dostosowana do potrzeb użytkownika, wystarczy przededefiniować \panel:

```
\def\panel{\colorbox{panelbackground}
{..... definicja ... } }
```

7.3.3. Pozostałe właściwości pakietu

Tło wyświetlanego obszaru może być zmienione poleceniem \backgroundcolor{kolor}, gdzie kolor musi być wcześniej zdefiniowanym kolorem. Ponadto, jako tło można wykorzystać plik graficzny wstawiając go poleceniem \overlay{plik graficzny}, przy czym plik ten jest przeskalowywany do rozmiarów strony.

Rysunek 20: Menu utworzone przy pomocy polecenia `\bottombuttons`

Również tło panelu może być zmienione: albo poleceniem `\paneloverlay{plik graficzny}`, wykorzystując plik graficzny jako tło (plik taki jest przeskalowywany do wymiarów panelu), albo poprzez zdefiniowanie koloru `panelbackground` aby uzyskać inny niż domyślny kolor. Podobnie można zdefiniować kolory pozostałych elementów, np. `buttonshadow`, `logobackground`, `buttonbackground`. Polecenia: `\paneloverlayempty` oraz `\overlayempty` wyłączają podkładanie plików graficznych jako tło i przywracają zdefiniowany kolor tła.

W pakiecie zostało zdefiniowane również polecenie `\changeoverlay` oraz dołączone zostały pliki PDF, które mogą posłużyć za tło: `overlay.pdf ... overlay10.pdf`, dzięki którym można zmieniać tło przy zmianie rozdziału. Oryginalne polecenie `\changeoverlay` zostało zdefiniowane w sposób następujący:

```
\newcounter{overlay}
\def\changeoverlay{\def\section{\if@sectionbreak\clearpage\fi%
  \stepcounter{panel@section}%
  \change\scr@section}}%
\def\change{\stepcounter{overlay}%
  \ifnum\c@overlay>10\setcounter{overlay}{1}\fi
  \overlay{overlay\theoverlay}
}
```

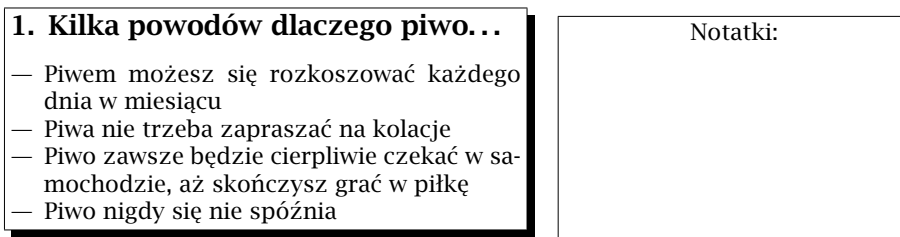
Można oczywiście podmienić pliki `overlayxx.pdf` lub zmienić działanie polecenia dostosowując je do swoich potrzeb.

Możliwe jest również utworzenie menu nawigacyjnego umieszczonego na dole lub na górze strony (rys. 20) – służą do tego polecenia `\bottombuttons` oraz `\topbuttons`. Tworzenie takich menu odwołują odpowiednio polecenia: `\nobottombuttons` i `\notopbuttons`. Należy zauważyć, że pomimo ustawienia opcji `polish` przy wczytywaniu pakietu `pdfscreen`, napisy w menu dolnym i górnym tworzone są w języku angielskim. Aby to zmienić należy w pliku konfiguracyjnym (lub w preambule dokumentu) zdefiniować polecenie `\@@buttons`, np. według poniższego:

```
\makeatletter
\def\@@buttons{\ifx\@shorttitle\@empty\relax\else%
  \quad{\@vattrib\@shorttitle}\fi\hfill%
  \Acrobatmenu{FirstPage} {\@battrib Pierwsza }
  \Acrobatmenu{PrevPage}  {\@battrib Poprzednia }
  \Acrobatmenu{NextPage}  {\@battrib Następna }
  \Acrobatmenu{LastPage}  {\@battrib Ostatnia }
  \Acrobatmenu{GoBack}    {\@battrib Wstecz }
  \Acrobatmenu{FullScreen}{\@battrib Prezentacja }
  \Acrobatmenu{Close}     {\@battrib Zamknij }
  \Acrobatmenu{Quit}      {\@battrib Wyjdź }
\quad}\vspace*{1.5pt}}
\makeatother
```

Pakiet przewiduje także środowisko `slide`, które tworzy pudełko o wymiarach obszaru tekstowego, pudełko, którego zawartość jest centrowana w pionie.

```
\begin{slide}
...
zawartość
...
\end{slide}
```



Rysunek 21: Przykład slajdu w wersji drukowanej wraz z miejscem na tworzenie notatek.

Należy zwrócić również uwagę na fakt, że wielkość wszystkich fontów została również zmieniona tak, aby łatwiejszym było czytanie tekstu z ekranu monitora. Jeżeli zaistniałaby potrzeba zmiany fontów do ich naturalnej wielkości, należy dodać słowo **real** do poleceń zmieniających wielkość, np. `\realnormalsize` aby uzyskać `\normal`, `\reallarge` dla `\large` itd.

Prezentacje przygotowane przy pomocy pakietu `pdfscreen` nadają się również do dalszej obróbki, np. przy pomocy programu `PPower4`, dzięki czemu możliwe jest uzyskanie stopniowych zmian zawartości strony.

Jeżeli użyta zostanie opcja `print`, zawartość środowiska `slide` drukowana jest w ramce, obok niej zaś drukowana jest druga ramka, zatytułowana **Notes**. Opcja ta może być przydatna w przypadku przygotowywania materiałów dla słuchaczy prelekcji – mogą oni w tych ramkach zapisywać notatki lub pytania dotyczące wykładu. Tytuł takich notatek musi być ustalony poleceniem:

```
\notesname{Notatki:}
```

Jeżeli tytuł ten nie zostanie zdefiniowany, ramka taka będzie bardzo wąska. Przykład przedstawia rys. 21

Przejścia między stronami można zrealizować przy pomocy polecenia `\pagedissolve{opcje}`, przy czym poszczególne opcje opisane są w tabelach 15 oraz 16. Polecenie to może wyglądać następująco:

```
/S /Split /D 2 /Dm /H /M /O
```

/noindent co stworzy efekt kurtyny trwający 2 sekundy /D 2, poruszający się poziomo /Dm /H ze środka w stronę brzegu /M /O.

7.4. pdfslide

Pakiet `pdfslide` jest rozwinięciem pakietu `pdfscreen`, pozwalającym na stosunkowo łatwe przygotowanie slajdów oraz pełnoekranowej prezentacji. Poszczególne slajdy mogą mieć własne tło, przy pomocy pakietu `pause.sty` i dodatkowego przetworzenia przy pomocy programu `PPoint` możliwe jest zrealizowanie wyświetlania poszczególnych punktów „po kolei”, również animowane zmiany poszczególnych stron nie stanowią problemu.

7.5. ifmslide

8. Dołączanie dźwięku oraz filmów wideo

Jak już wspomniano przy okazji poleceń PDF_{TEX}-a, do pliku PDF można dołączyć różnego rodzaju notatki m.in.:

- tekstową,
- dźwiękową,
- film wideo

Najprostszym rodzajem notatki jest notatka tekstowa, która w PDF_{TEX}-u może wyglądać następująco:

```
\pdfannot{
  /Subtype /Text
  /Rect [30 30 60 60]
  /Open true
  /T (to jest tytuł)
  /Contents (a to jest tresc)
  /C [1 0 1]
  /Border [0 0 1]
}
```

przy czym zarówno w treści jak i w tytule mogą występować znaki zgodne z PDFDocEncoding lub znaki UNICODE. W przypadku UNICODE ciąg znaków w treści notatki musi zaczynać się od <FEFF...>, np. dla ciągu ABC wygląda to następująco:

```
/Contents (<FEFF004100420043>)
```

Link do pliku dźwiękowego można zrealizować w następujący sposób:

```
ala ma asa
\pdfstartlink
  height 10pt depth 2pt
  attr {
    /C [1 0 0]
    /Border [0 0 2]
  }
  user{
    /Subtype /Link
    /A <<
      /S /launch
      /F (dingdong.wav) % <--- plik dźwiękowy
    >>
  }
{treść linku}
\pdfendlink
```

przy czym należy zauważyć, że rozwiązanie to przenośne jest w zasadzie razem z komputerem. Po pierwsze, plik dźwiękowy nie jest wbudowywany w dokument i musi znajdować się na dysku (w bieżącym katalogu), po drugie zaś, Acrobat domaga się plugina do obsługi tegoż (Quicktime), linuksowy xpdf nie obsługuje tego rodzaju , podobnie (k)gv.

9. Źródła

Poszczególne programy oraz pakiety dostępne są w Sieci:

Acrobat to komercyjny program stworzony przez Adobe. Więcej informacji dostępnych jest na stronach producenta: <http://www.adobe.com>.

dvipdfm program umożliwiający tworzenie plików PDF bezpośrednio z DVI, dostępny zarówno w wersji dla Linuksa jak i dla Windows. Więcej informacji na stronie <http://gaspra.kettering.edu/dvipdfm/>.

epstopdf

Ghostscript jest to interpreter postscriptu, umożliwiający drukowanie plików PS na różnego rodzaju urządzeniach. Dostępny jest m.in. z adresu <http://www.cs.wisc.edu/~ghost/index.html>.

hyperref pakiet wspomagający tworzenie dokumentów interaktywnych przy pomocy PDF_{TEX}-a, dostępny na serwerach CTAN w katalogu [/macros/latex/.../hyperref](#), najbardziej aktualna wersja umieszczana jest na serwerze <http://www.tug.org/applications/hyperref>.

mwcls polskie klasy dokumentów, opracowane przez Marcina Wolińskiego, dostępne są pod adresem <http://www.mimuw.edu.pl/~wolini/mwcls.html>.

pdfcrypt pakiet służący do ustawiania zabezpieczeń w plikach PDF, dostępny na serwerze CTAN w katalogu [/macros/latex/contrib/supported/oberdiek](#).

pdfTeX implementacja \TeX -a pozwalająca na tworzenie plików PDF. Źródła dostępne są na serwerze [ftp://ftp.cstug.cz/pub/tex/local/cstug/thanh/pdftex/](#).

pdfscreen pakiet służący do tworzenia dokumentów przystosowanych do czytania z ekranu monitora, dostępny na serwerze CTAN w katalogu [/macros/latex/contrib/supported/pdfscreen](#).

pdfslide

prosper pakiet wspomagający tworzenie prezentacji, dostępny z .

PStill program na licencji shareware umożliwiający destylację plików postscriptowych, dostępny z serwera <http://www.this.net/~frank/pstill.html>.

PSTricks jest to zaawansowany pakiet graficzny, wspomagający postscript, dostępny pod adresem <http://www.tug.org/applications/PSTricks>.

seminar klasyczny pakiet służący do tworzenia prezentacji. Najnowsza wersja wraz z bogatą dokumentacją dostępna jest pod adresem .

Bibliografia

- [1] Mark A. Wicks. Dvipdfm user's manual. Podręcznik użytkownika dołączany do programu, dostępny również pod adresem <http://gaspra.kettering.edu/dvipdfm/>, 1999.
- [2] Adobe Systems Incorporated. Acrobat Distiller Parameters. Technical Note 5151. Dokument dostępny w postaci elektronicznej pod adresem <http://www.adobe.com>, 1999.
- [3] Zbigniew Okoń. Tworzenie dokumentów pdf za pomocą ghostscripta. Dokument dostępny w wersji elektronicznej pod adresem <http://www.pckurier.pl/webmaster/2000/marzec/okon/pdf.html>, 2000.
- [4] Hàn Thé Thàhn, Sebastian Rahtz, Hans Hagen. PDF \TeX user manual. Podręcznik PDF \TeX -a, dostępny w wersji elektronicznej pod adresem <http://www.tug.org/applications/pdftex/>, 2000.
- [5] Adobe Systems Incorporated. *PDF Reference Version 1.3*. Addison-Wesley, wydanie 2, 2000. Dokument dostępny w postaci elektronicznej pod adresem: <http://partners.adobe.com/asn/developer/technotes.html>.
- [6] Tomasz Przechlewski, Stanisław Wawrykiewicz. Instalowanie fontów Type 1/TrueType. Opis dostępny pod adresem <http://www.gust.org.pl/fonty/fonthint.html>, 2001.
- [7] Damir Rakityansky. Using truetype fonts with tex (latex) and pdftex (pdflatex). Dokument dostępny w wersji elektronicznej pod adresem <http://www.radamir.com/tex/ttf-tex.htm>.
- [8] Wojciech Myszka. Włączanie grafik do tekstów w \LaTeX 2 ϵ . publikacja dostępna w wersji elektronicznej pod adresem <http://www.immt.pwr.wroc.pl/~myszka/grafika/grafika.pdf>, 2000.
- [9] Michel Goossens, Sebastian Rahtz, Eitan M. Gurari, Ross Moore, Robert S. Sutor. *The \LaTeX Web Companion*. Tools and Techniques for Computer Typesetting. Addison-Wesley Longman, Inc., Reading, Massachusetts, wydanie 1, 1999.
- [10] Heiko Oberdiek. Pdf information and navigation elements with hyperref, pdftex, and thumbpdf. *Biuletyn GUST*, 13:31–39, 1999.

O dokumencie

Najnowszą wersję tego dokumentu można znaleźć pod adresem <http://www.sapijaszko.net/pedeefy.pdf>. Bieżący numer wersji to 1.4. Dokument ten dostępny jest na licencji LPPL w wersji 1.2 lub wyższej tak jak to opisano w \LaTeX Project Public License, <http://www.ctan.org/tex-archive/macros/latex/base/lppl.txt>.

Revision Log for pedeefy.tex

Revision 1.4 2002/03/26 21:42:11 grzegorz
 Dodano opis pakietów **pdfcrypt** i **pdfpages**, zaktualizowano również opis pakietu **hyperref** .

Revision 1.3 2001/05/31 17:45:40 sapi
 Dodano parę słów o destylacji plików postscriptowych, ponadto dokonano niewielu zmian kosmetycznych.

Revision 1.2 2001/03/31 21:27:44 sapi
 Poprawiono opis pakietu **hyperref** oraz dodano kilka słów na temat fontów.

Revision 1.1 2001/01/15 20:41:24 sapi
 Initial revision