Pakiet MAK jako narzędzie do produkowania danych dla TEX

Jerzy Swianiewicz
Biblioteka Narodowa, Warszawa
jswian@bn.org.pl

Streszczenie

Pakiet MAK jest zbiorem programów obsługi baz danych określonego typu. Celem tego artykułu jest przedstawienie zasad przetworzenia danych z bazy na dane dla edytora ze szczególnym uwypukleniem zastosowania TEX-a.

Pakiet MAK jest zbiorem programów obsługi baz danych określonego typu. Ze względu na to, że został on opracowany w Bibliotece Narodowej rozpowszechnił się w wielu bibliotekach publicznych w całym kraju. Dla wielu jednak bibliotek okazał się on za trudny lub za mało przyjazny tak, że tylko w części jest naprawdę wykorzystywany.

W ramach pakietu MAK istnieje zestaw programów pozwalający na dowolne, sterowane, przekształcanie informacji pobieranych z bazy danych na sekwencyjny zbiór tekstowy, w którym informacje mogą być dowolnie ułożone i obudowane innymi tekstami. Ten mechanizm stwarza możliwość przygotowania tekstów będących danymi wejściowymi dla różnego rodzaju edytorów bądź programów składu drukarskiego. Takim edytorem może być TEX, LATEX, WORD i wiele innych.

Tak się złożyło, że właśnie dla TEX-a opracowanych zostało wiele tzw. *formatów*, sterujących przetwarzaniem danych z bazy na różnego rodzaju zestawienia, tabele, bibliografie, katalogi itp. Wiele wydawnictw tego typu powstaje, z zastosowaniem TEX-a w ostatnim etapie, prawie automatycznie.

Celem tego artykułu jest przedstawienie zasad działania wspomnianego mechanizmu przetworzenia danych z bazy na dane dla edytora ze szczególnym uwypukleniem zastosowania TEX-a. Autor liczy na zainteresowanie środowiska TEX-owego przedstawioną metodą przygotowywania publikacji. Są dwa powody dlaczego to zainteresowanie jest bardzo pożądane.

Najpierw, może się przyczynić do poprawienia jakości produktów w ten sposób uzyskiwanych. Autor zdaje sobie sprawę, że stosowane przez niego mechanizmy TEX-owe mogłyby być bardziej sprawne.

Następnie, bardzo pożądane byłoby ujawnienie się specjalistów od TEX-a do których mogłyby się zwracać biblioteki o pomoc w przygotowaniu różnego rodzaju publikacji.

Jest niezaprzeczalnym faktem, że pakiet MAK, mimo że jest produktem dość staroświeckim – pracuje w DOSie – bardzo się rozpowszechnił w różnych większych i mniejszych bibliotekach publicznych w kraju i można oczekiwać, będzie się dalej rozpowszechniał z powodu różnych refundacji jakie organizuje Ministerstwo Kultury.

Bazy MAK

Pakiet MAK jest zbiorem programów obsługi baz danych typu MAK, w skrócie: *baz* MAK. Baza MAK to zbiór rekordów zwanych *dokumentami*. Informacje (teksty) w dokumencie podzielone są na *pola*, które z kolei składają się z *podpól*. Tylko podpola mogą zawierać informacje tekstowe. Pole ma tylko nazwę (*nazwa pola*) i grupuje pewną liczbę podpól również posiadających nazwy (*nazwa podpola*).

Pola i podpola są powtarzalne. W dokumencie, każde pole może być powtórzone dowolną ilość razy lub nie występować w ogóle. Podobnie w każdym polu jego podpola mogą być powtarzane dowolną ilość razy lub w ogóle nie występować.

Podpole *występuje* w polu jeżeli zawiera jakąkolwiek treść (tzn. tekst dowolnej długości). Pole *występuje* w dokumencie jeżeli występuje w nim przynajmniej jedno podpole.

Zakładając bazę należy określić strukturę jej dokumentów to znaczy wskazać jakie pola mają prawo wystąpić w dokumencie (podając ich nazwy), oraz – dla każdego pola – jakie podpola mogą w nim wystąpić (podając ich nazwy).

W rzeczywistości, pełną nazwą podpola jest para: (*nazwa-pola, nazwa-podpola*) co nadal nie identyfikuje wystąpienia podpola w dokumencie, ale powinno określać jego przeznaczenie.

Pełną identyfikację podpola w konkretnym dokumencie uzyskamy dodając do nazwy pola numer jego wystąpienia w tym dokumencie a do nazwy podpola – numer wystąpienia tego podpola w danym polu.

W dokumencie, pola mogą występować w dowolnej kolejności. Również, kolejne wystąpienia określonego pola nie muszą być zgrupowane obok siebie, mogą być przedzielone wystąpieniami innych pól. To samo dotyczy podpól i ich wystąpień w polu: mogą być rozmieszczone dowolnie.

W bazie można definiować *indeksy* dla wartości wskazanych pól/podpól. Indeksy ułatwiają i – przede wszystkim – przyśpieszają wyszukiwanie dokumentów zawierających daną wartość.

Program MAKD

Funkcją programu MAKD jest przetwarzanie danych pobieranych z bazy MAK na sekwencyjny zbiór tekstowy. Tym zbiorem będzie plik o nazwie takiej jak nazwa bazy i rozszerzeniu `.tex`. Sposób konwersji danych pobieranych z dokumentów bazy opisuje tzw. *format-D*, który jest programem wykonywanym przez program MAKD przy opracowywaniu kolejnych dokumentów bazy.

Format-D jest plikiem tekstowym zawierającym instrukcje realizowane przez program MAKD. Instrukcje te wyróżniają się w tekście tym, że zaczynają się od znaku `<` i ogólnie mają postać:

<*skrót*>*parametry*

Liczbowe parametry instrukcji również podawane są w nawiasach kątowych: `<`, `>`.

Program MAKD dostaje na wejściu format-D i listę dokumentów z bazy MAK a jego działanie polega na kopiowaniu do tekstowego zbioru wyjściowego tekstu formatu-D. Podczas tego kopiowania są wychwytywane i realizowane instrukcje MAKD. Po dojściu do tzw. części centralnej formatu-D, (której początek zaznaczony jest instrukcją `<PD>`), program MAKD pobiera pierwszy dokument z listy i realizuje część centralną formatu-D w powiązaniu z tym dokumentem (zawarte w części centralnej instrukcje będą się odnosiły do tego dokumentu). Następnie część centralna będzie wykonywana w powiązaniu z kolejnymi dokumentami z listy, itd. aż do wyczerpania dostarczonej listy dokumentów. Na koniec, zostaje skopiowana część końcowa formatu-D zwana EPILOGIEM.

Dla wygody dalszego opisu wprowadźmy termin *interpretowanie tekstu* rozumiejąc go jako czynność kopiowania tekstu formatu-D na wyjście wraz z realizacją zawartych w nim instrukcji.

Budowa formatu-D. Ogólna struktura formatu-D jest więc następująca:

```
DEFINICJE
$$$
PROLOG<K>komentarz
CZĘŚĆ CENTRALNA FORMATU-D
<EP>EPILOG
```

Objaśnienia:

DEFINICJE – mogą być puste i w tej prezentacji założymy, że są puste. Nie będziemy ich omawiać,

$$$ – separator oddzielający DEFINICJE od PROLOGU,

PROLOG – tekst interpretowany raz przed przystąpieniem do przetwarzania grupy dokumentów bazy,

`<K>` – instrukcja MAKD pełniąca rolę separatora. Tutaj, zaznacza koniec PROLOGu,

komentarz – dowolny tekst, nie zawierający znaku '`<`'. Komentarz zaczyna się bezpośrednio po instrukcji `<K>`, a kończy go dowolna instrukcja MAKD,

CZĘŚĆ CENTRALNA FORMATU-D – omówiona niżej część formatu-D, która jest interpretowana, kolejno, dla każdego opracowywanegu dokumentu bazy,

`<EP>` – instrukja MAKD zaznaczająca koniec CZĘŚCI CENTRALNEJ i początek EPILOGU

EPILOG – tekst interpretowany raz po zakończeniu przetwarzania grupy dokumentów bazy.

CZĘŚĆ CENTRALNA formatu-D. CZĘŚĆ CENTRALNA formatu posiada budowę następującą:

<PD>*t-pd*<K>*komentarz*
OPRACOWANIE-POLA np_1
OPRACOWANIE-POLA np_2
... ...
OPRACOWANIE-POLA np_n
<KD>*t-kd*<K>*komentarz*

Objaśnienia:

t-pd – tekst interpretowany przed rozpoczęciem przetwarzania każdego dokumentu bazy. Instrukcja `<PD>` wskazuje początek tego tekstu, a `<K>` – koniec,

OPRACOWANIE-POLA np_i – omówiona dalej struktura w formacie-D, która jest interpretowana dla pola o nazwie np_i. Jest ona interpretowana tyle razy ile jest wystąpień pola np_i w dokumencie: kolejno, dla każdego wystąpienia,

t-kd – tekst interpretowany po zakończeniu przetwarzania każdego dokumentu bazy. Instrukcja `<KD>` wskazuje początek tego tekstu, a `<K>` – koniec.

Pola np_1, np_2, ... np_n jest to wybór pól, które twórca danego formatu-D postanowił uwzględnić przy opracowywaniu tekstu wyjściowego. Opracowanie tego samego pola może wystąpić w CZĘŚCI CENTRALNEJ wielokrotnie i za każdym razem być inaczej interpretowane (w ciągu np_i elementy mogą się powtarzać).

Przy zastosowaniu przyjętej powyżej terminologii, dla konkretnego dokumentu, realizacja CZĘŚCI CENTRALNEJ polega na:

(1) zinterpretowaniu tekstu *t-pd*,
(2) realizacji, kolejno, wszystkich: „OPRACOWANIE-POLA np_i" (będzie to omówione w punkcie następnym),
(3) zinterpretowaniu tekstu *t-kd*.

OPRACOWANIE-POLA np_i. OPRACOWANIE-POLA np_i jest oznaczeniem następującej struktury w formacie:

<Np>np_i<K>*komentarz*
<PjI>*t-pji*<K>*komentarz*
<PjK>*t-pjk*<K>*komentarz*
<PwpI>*t-pwpi*<K>*komentarz*
<PwpK>*t-pwpk*<K>*komentarz*
<PwsI>*t-pwsi*<K>*komentarz*
<PwsK>*t-pwsk*<K>*komentarz*
<PwoI>*t-pwoi*<K>*komentarz*
<PwoK>*t-pwok*<K>*komentarz*
<Pbr>*t-pbr*<K>*komentarz*
OPRACOWANIE-PODPOLA npp_1
OPRACOWANIE-PODPOLA npp_2
... ...
OPRACOWANIE-PODPOLA npp_n

Objaśnienia:

<Np> – instrukcja rozpoczynająca strukturę OPRACOWANIE POLA i określająca nazwę pola. Nazwą pola jest tekst np_i zawarty między tą instrukcją i <K>.

<PjI> – tekst *t-pji* ma być interpretowany tylko wtedy gdy opracowywane pole jest jednokrotne. Ponadto, interpretacja tekstu *t-pji* ma nastąpić przed opracowaniem podpól należących do tego pola.

<PjK> – tekst *t-pjk* ma być interpretowany tylko wtedy gdy opracowywane pole jest jednokrotne. Ponadto, interpretacja tekstu *t-pjk* ma nastąpić po opracowaniu podpól należących do tego pola.

<PwpI> – tekst *t-pwpi* ma być interpretowany tylko wtedy gdy opracowywane pole jest wielokrotne i jest to jego *pierwsze* wystąpienie. Ponadto, interpretacja tekstu *t-pwpi* ma nastąpić przed opracowaniem podpól należących do tego pola.

<PwpK> – tekst *t-pwpk* ma być interpretowany tylko wtedy gdy opracowywane pole jest wielokrotne i jest to jego *pierwsze* wystąpienie. Ponadto, interpretacja tekstu *t-pwpk* ma nastąpić po opracowaniu podpól należących do tego pola.

<PwsI> – tekst *t-pwsi* ma być interpretowany tylko wtedy gdy opracowywane pole jest wielokrotne i jest to jego *środkowe* wystąpienie. Ponadto, interpretacja tekstu *t-pwsi* ma nastąpić przed opracowaniem podpól należących do tego pola.

<PwsK> – tekst *t-pwsk* ma być interpretowany tylko wtedy gdy opracowywane pole jest wielokrotne i jest to jego *środkowe* wystąpienie. Ponadto, interpretacja tekstu *t-pwsk* ma nastąpić po opracowaniu podpól należących do tego pola.

<PwoI> – tekst *t-pwoi* ma być interpretowany tylko wtedy gdy opracowywane pole jest wielokrotne i jest to jego *ostatnie* wystąpienie. Ponadto, interpretacja tekstu *t-pwoi* ma nastąpić przed opracowaniem podpól należących do tego pola.

<PwoK> – tekst *t-pwok* ma być interpretowany tylko wtedy gdy opracowywane pole jest wielokrotne i jest to jego *ostatnie* wystąpienie. Ponadto, interpretacja tekstu *t-pwok* ma nastąpić po opracowaniu podpól należących do tego pola.

<Pbr> – tekst *t-pbr* ma być interpretowany tylko wtedy gdy opracowywane pole nie występuje w opracowywanym dokumencie.

OPRACOWANIE-PODPOLA npp_i – omówiona niżej struktura w formacie-D, która jest interpretowana dla podpola o nazwie npp_i. Jest ona interpretowana tyle razy ile jest wystąpień podpola npp_i w opracowywanym polu: kolejno, dla każdego wystąpienia.

Uwaga – jeżeli którykolwiek z tekstów: *t-pji*, *t-pjk*, *t-pwpi*, *t-pwpk*, *t-pwsi*, *t-pwsk*, *t-pwoi*, *t-pwok* lub *t-pbr* jest pusty to cały wiersz w strukturze OPRACOWANIE-POLA np_i może być pominięty.

Realizując powyższą strukturę, program MAKD wybierze, z aktualnie przetwarzanego dokumentu bazy, wszystkie pola o nazwie np_i i będzie je opracowywał kolejno. Sposób opracowywania będzie zależał od tego do jakiej kategorii to kolejne pole się zalicza: pojedyncze, pierwsze z wielu, środkowe z wielu, ostatnie z wielu czy „brakujące" (kategoria „brakujące" będzie wybrana gdy pole o nazwie np_i nie występuje w przetwarzanym dokumencie).

A więc, w przypadku pola pojedynczego zostaną wykonane kolejno następujące czynności:

(1) interpretacja tekstu *t-pji*,
(2) realizacja, kolejno, wszystkich struktur:
OPRACOWANIE-PODPOLA npp_i
(3) interpretacja tekstu *t-pjk*

Dla „pierwszego z wielu", „środkowego z wielu" i „końcowego z wielu" czynności będą analogiczne, tylko inne teksty będą wybrane do interpretacji.

W przypadku „brakującego" pola jedyną czynnością będzie interpretacja tekstu *t-pbr*.

Program MAKD, dla formatów-D, zdefiniował dwie fikcyjne nazwy pól: **@brak@** i **@kazde@**

Pole o nazwie **@brak@**, użyte w formacie-D, zachowuje się tak, jak gdyby było określone w każdej bazie, ale w żadnym dokumencie nie występuje (należy do kategorii „brakujące")

Nazwa **@kazde@**, użyta w formacie-D jako nazwa pola, zachowuje się tak, jak gdyby była równa każdej nazwie pola w dokumencie. Dokument jest wtedy traktowany tak, jak gdyby zawierał jedno wielokrotne pole. Podczas opracowywania takiego pola można, za pomocą odpowiedniej instrukcji, dowiedzieć się jaka jest prawdziwa nazwa każdego wystąpienia tego pola.

Wybiegając trochę naprzód powiemy że, stworzona została również fikcyjna nazwa podpola **@kazde@**. Nazwa **@kazde@**, użyta w formacie jako nazwa podpola, zachowuje się tak, jak gdyby była równa każdej nazwie podpola w polu. Pole jest wtedy traktowane tak, jak gdyby zawierało jedno wielokrotne podpole. Podczas opracowywania takiego podpola można, za pomocą odpowiedniej instrukcji, dowiedzieć się jaka jest prawdziwa nazwa każdego wystąpienia tego podpola.

OPRACOWANIE-PODPOLA npp_i**.** OPRACOWANIE-PODPOLA npp_i jest oznaczeniem następującej struktury w formacie:

`<Npp>`npp_i`<K>`*komentarz*
`<PPj>`*t-ppj*`<K>`*komentarz*
`<PPwp>`*t-ppwp*`<K>`*komentarz*
`<PPws>`*t-ppws*`<K>`*komentarz*
`<PPwo>`*t-ppwo*`<K>`*komentarz*
`<PPbr>`*t-ppbr*`<K>`*komentarz*

Objaśnienia:

`<Npp>` – instrukcja rozpoczynająca strukturę OPRACOWANIE PODPOLA i określająca nazwę podpola. Nazwą podpola jest tekst npp_i zawarty między tą instrukcją i `<K>`.

`<PPj>` – tekst *t-ppj* ma być interpretowany tylko wtedy gdy opracowywane podpole jest jednokrotne.

`<PPwp>` – tekst *t-ppwp* ma być interpretowany tylko wtedy gdy opracowywane podpole jest wielokrotne i jest to jego *pierwsze* wystąpienie.

`<PPws>` – tekst *t-ppws* ma być interpretowany tylko wtedy gdy opracowywane podpole jest wielokrotne i jest to jego *środkowe* wystąpienie.

`<PPwo>` – tekst *t-ppwo* ma być interpretowany tylko wtedy gdy opracowywane podpole jest wielokrotne i jest to jego *ostatnie* wystąpienie.

`<PPbr>` – tekst *t-ppbr* ma być interpretowany tylko wtedy gdy opracowywane podpole nie występuje w opracowywanym polu.

Uwaga – jeżeli którykolwiek z tekstów: *t-ppj*, *t-ppwp*, *t-ppws*, *t-ppwo* lub *t-ppbr* jest pusty to cały wiersz w strukturze OPRACOWANIE-PODPODPOLA npp_i może być pominięty.

Realizując strukturę OPRACOWANIE-PODPOLA npp_i, program MAKD wybierze, z aktualnie przetwarzanego pola, wszystkie podpola o nazwie npp_i i będzie je opracowywał kolejno. Przyczem, sposób opracowywania będzie zależał od tego do jakiej kategorii to kolejne podpole się zalicza: pojedyncze, pierwsze z wielu, środkowe z wielu, ostatnie z wielu czy „brakujące" (kategoria „brakujące" będzie wybrana gdy podpole o nazwie npp_i nie występuje w przetwarzanym polu).

A więc, w przypadku podpola pojedynczego zostanie wykonana interpretacja tekstu *t-ppj*,
w przypadku pierwszego wystąpienia podpola wielokrotnego – interpretacja tekstu *t-ppwp*,
w przypadku środkowego wystąpienia podpola wielokrotnego – interpretacja tekstu *t-ppws*,
w przypadku ostatniego wystąpienia podpola wielokrotnego – interpretacja tekstu *t-ppwo*,
i w przypadku nie wystąpienia podpola – interpretacja tekstu *t-ppbr*

Instrukcje programu MAKD. W omawianej powyżej strukturze formatu występowały instrukcje MAKD takie jak: `<PD>`, `<K>`, `<KD>`, `<Np>`, `<PjI>`, `<PjK>`, `<PwpI>`, `<PwpK>`, `<PwsI>`, `<PwsK>`, `<PwoI>`, `<PwoK>`, `<Pbr>`, `<Npp>`, `<PPj>`, `<PPwp>`, `<PPws>`, `<PPwo>`, `<PPbr>` i `<EP>`.

Instrukcje te, w większości, można traktować jako instrukcje warunkowe lub deklaracyjne. Np.:

`<PD>` – jeżeli przystępujesz do przetwarzania kolejnego dokumentu, to interpretuj tekst zawarty między tą istrukcją a instrukcją `<K>`,

`<PwpI>` – jeżeli przystępujesz do przetwarzania pierwszego z wielu wystąpień pola wskazanego przez instrukcję `<Np>`, to interpretuj tekst zawarty między intrukcją `<PwpI>` a `<K>`.

Inne instrukcje mogą być umieszczone w PROLOGU, EPILOGU lub *tekstach* CZĘŚCI CENTRALNEJ FORMATU. Wśród tych instrukcji podstawowymi są dwie instrukcje pobierania danych z bazy. Są to:

`<Dpp>` – przepisz zawartość bieżącego podpola na wyjście,

`<Bpp>` – pobierz zawartość bieżącego podpola do bufora **B**.

Te instrukcje mogą występować w *tekstach* struktury OPRACOWANIE-PODPOLA i odnoszą się do bieżącego podpola, którego nazwa określona jest przez instrukcję `<Npp>`.

Istnieje wiele instrukcji pozwalających operować na buforze **B**, w szczególności badać i zmieniać jego zawartość bądź wyprowadzać na wyjście.

Oprócz bufora **B**, w środowisku programu MAKD, istnieje 10 buforów tekstowych, $\mathbf{T}_0$, $\mathbf{T}_1$, ..., $\mathbf{T}_9$, na które – za pomocą odpowiednich instrukcji – można przełączać strumień wyjściowy. Specjalne instrukcje pozwalają dokonywać pewnych operacji na tych buforach oraz wyprowadzać ich zawartość na wyjście podstawowe.

W środowisku programu MAKD zdefiniowane są również *rejestry.* Są to:

- 34 rejestry numeryczne 16-bitowe, r_0, r_1, ..., r_{33}. Niektóre z nich są automatycznie inicjowane przez takie wartości jak: *numer systemowy bieżącego dokumentu, numer powtórzenia bieżącego pola, numer powtórzenia bieżącego podpola, długość tekstu w buforze* **B** i inne;
- 20 rejestrów numerycznych 32-bitowych, rd_0, ..., rd_{19}. Niektóre z nich są również automatycznie inicjowane;
- 12 rejestrów tekstowych 32-bajtowych, rt_0, ..., rt_{11}. Dwa z nich są automatycznie inicjowane przez nazwę bieżącego pola i podpola.

Przykłady instrukcji sterujących. Opiszemy teraz te instrukcje sterujące, które później zostaną wykorzystane w przykładach formatów-D.

`<crlf>` – jest zapisem pary znaków powodujących przejście do nowej linii. Znaki łamania linii umieszczone w tekście formatu są pomijane w trakcie *interpretowania* tekstów, natomiast ta instrukcja spowoduje wysłanie na wyjście właściwych znaków;

`<Ozer><r>` – zeruj rejestr r;

`<Onast><r>` – zwiększ zawartość rejestru r o 1;

`<Osre><r><p>` – wstaw p do rejestru r;

`<Oddr><rd>` – wypisz zawartość rejestru długiego rd w zapisie dziesiętnym;

`<Ofun><n><p>` – wykonaj funkcję tr_n z parametrem p; np.:

`<Ofun><1><p>` – wypisz p pierwszych wyrazów z bufora **B** (funkcja tr_1);

`<Ofun><3><0>` – wypisz zawartość bufora **B** pomijając początkowe znaki '0' (funkcja tr_1);

`<Oskp><r><p>` – jeżeli zawartość rejestru $r = p$ – zaniechaj interpretowania bieżącego pola, podporządkowanych mu podpól oraz tekstu kończącego pole;

`<Orow><r><p>`*tekst1*`<el>`*tekst2*`<sep>` – jeżeli zawartość rejestru $r = p$ – interpretuj *tekst1*, w przeciwnym przypadku, – *tekst2*. W przypadku gdy *tekst2* jest pusty, używa się wersji skróconej: `<Orow><r><p>`*tekst*`<sep>`;

`<Owie><r><p>`*tekst1*`<el>`*tekst2*`<sep>` – jeżeli zawartość rejestru r spełnia warunek $r = p$, interpretuj *tekst1*, w przeciwnym przypadku, – *tekst2*. W przypadku gdy *tekst2* jest pusty, używa się wersji skróconej: `<Owie><r><p>`*tekst*`<sep>`;

`<Ozirow><rt><p>`tekst*tekst1*`<el>`*tekst2*`<sep>` – jeżeli tekst zawarty w rejestrze tekstowym rt jest równy, złożonemu z p znaków, tekstowi występującemu bezpośrednio po instrukcji, interpretuj *tekst1*, w przeciwnym przypadku, – *tekst2*. W przypadku gdy *tekst2* jest pusty, używa się wersji skróconej: `<Ozirow><rt><p>`tekst*tekst*`<sep>`. Korzystając z tego, że rejestry rt_4 i rt_5 zawierają, odpowiednio, nazwę bieżącego pola i podpola, instrukcja ta jest używana do rozpoznania, które pole(podpole) jest aktualnie przetwarzane;

`<Otbz><n>` – zeruj bufor $\mathbf{T}n$;

`<Otst><n>` – przełącz wyjście na bufor $\mathbf{T}n$;

`<Otko><0>` – przerwij wypisywanie do $\mathbf{T}n$ (przełącz wyjście na podstawowe);

`<Odt><n>` – wypisz zawartość bufora $\mathbf{T}n$;

Przykłady formatów

Uproszczona KARTA KATALOGOWA (format PR01). W dodatku A, pokazany jest w całości *format-D* bardzo uproszczonej bibliotecznej karty katalogowej. (Formaty dla pełnych kart katalogowych składają się z około 1000 linii.) Odnosi się on do struktury bazy bibliotecznej w tzw. formacie MARC BN, gdzie sygnatura książki jest zapisana w podpolu 023/s (taki zapis odczytujemy: pole 023, podpole s), autor bądź autorzy opisani są w polach 100 (podpole 1 nazwisko, podpole 2 imię), tytuł książki – pole 200 (podpole a – tytuł właściwy, podpole e – dodatek do tytułu), tzw. adres wydawniczy – pole 210 (podpole a – miejsce wydania, podpole c – wydawnictwo, podpole d – rok wydania). Pozostałe informacje o książce i osobach za nią odpowiedzialnych zostaną pominięte.

Tradycyjny układ graficzny karty każe sygnaturę(y) umieścić w prawym górnym rogu karty. Następnie hasło, którym jest autor lub (gdy go nie ma) pierwsze słowa tytułu. Hasło drukuje się od lewego marginesu pismem pogrubionym. W następnych akapitach: tytuł i adres wydawniczy. (*W tym miejscu zachęcamy czytelnika do przyjrzenia się, umieszczonemu w dodatku A, formatowi-D PR01.*)

W wyniku zastosowania tego *formatu-D* do dwóch dokumentów z odpowiedniej bazy program MAKD wyprodukował poniższy zbiór TEX-owy.

```
%********** PROLOG
\hoffset=-15mm\voffset=-10mm
\hsize=110mm\vsize=75mm
\tolerance=9900
%******* POCZĄTEK DOKUMENTU: 8
\rightline{\bf 1.355.506}
\vskip2mm
\noindent{\bf
\uppercase{Braszczyński} Janusz }\par
\vskip2mm
Krystalizacja odlewów\par
\vskip1mm
Warszawa : Wydawnictwa Naukowo-Techniczne\unskip, 1991\par
%*******    KONIEC DOKUMENTU: 8
\vfill\eject
%******* POCZĄTEK DOKUMENTU: 15
\rightline{\bf 1.355.533}
\vskip2mm
\noindent{\bf
\uppercase{Wielkie tematy  ...}}\par
\vskip2mm
Wielkie tematy myśli politycznej i politologicznej\par
\vskip1mm
Warszawa : COMSNP\unskip, 1989\par
%*******    KONIEC DOKUMENTU: 15
\vfill\eject
%*******    EPILOG
\end
```

A tak wyglądają wyprodukowane kartki[1]:

1.355.506

BRASZCZYŃSKI Janusz

Krystalizacja odlewów

Warszawa : Wydawnictwa Naukowo-Techniczne, 1991

1.355.533

WIELKIE TEMATY ...

Wielkie tematy myśli politycznej i politologicznej

Warszawa : COMSNP, 1989

[1] Ze względów edycyjnych, obrazy kartek zostały zmniejszone w stosunku do zadeklarowanych w TEX

Uproszczona KARTA ODSYŁACZOWA (format PR02). W opisie książki w systemie MARC BN, w polu 700 (podpola: 1 – nazwisko, 2 – imię, v – rodzaj odpowiedzialności) opisuje się osoby, które przyczyniły się do powstania książki ale inne niż autor, np. tłumacz, redaktor, autor przedmowy, ilustrator. Dla tych osób tworzy się karty odsyłaczowe. Jeżeli opis książki zawiera kilka pól 700 to z jednego opisu powstaje kilka kart odsyłaczowych. Mają one część wspólną: skrócony opis katalogowy książki, do którego się odsyłamy.

Format ten jest ilustracją zastosowania bufora $\mathbf{T}_0$. Mianowicie, w buforze $\mathbf{T}_0$ kompletuje się opis książki, a następnie, dla każdego wystąpienia pola 700, generuje się odsyłacz.

W wyniku zastosowania powyższego *formatu* do dwóch dokumentów z odpowiedniej bazy program MAKD wyprodukował poniższy zbiór TEX-owy.

```
%********** PROLOG
\hoffset=-15mm\voffset=-10mm
\hsize=110mm\vsize=75mm
\tolerance=9900
%******* POCZĄTEK DOKUMENTU: 20
\noindent{\bf\uppercase{Ciołkosz} Adam
\hskip10mm[tł]}\par
\vskip2mm
\line{\hfil zob.\hfil\hfil}
\noindent{\bf
\uppercase{Weissberg} Alexander }\par
\vskip2mm
Wielka czystka\par
\vskip1mm
Warszawa : "Czytelnik"\unskip, 1990\par
\vfill\eject
\noindent{\bf\uppercase{Herling-Grudziński}
Gustaw \hskip10mm[tł]}\par
\vskip2mm
\line{\hfil zob.\hfil\hfil}
\noindent{\bf
\uppercase{Weissberg} Alexander }\par
\vskip2mm
Wielka czystka\par
\vskip1mm
Warszawa : "Czytelnik"\unskip, 1990\par
\vfill\eject
%*******    KONIEC DOKUMENTU: 20
%******* POCZĄTEK DOKUMENTU: 29
\noindent{\bf\uppercase{Szwarcman-Czarnota}
Bella \hskip10mm[tł]}\par
\vskip2mm
\line{\hfil zob.\hfil\hfil}
\noindent{\bf
\uppercase{Van Hamme} Jean }\par
\vskip2mm
Largo Winch i Grupa W\par
\vskip1mm
Warszawa : "Orbita"\unskip, 1991\par
\vfill\eject
%*******    KONIEC DOKUMENTU: 29
%*******    EPILOG
\end
```

Z przetworzonych 2 opisów książek powstały następujące 3 kartki odsyłaczowe:

CIOŁKOSZ Adam **[tł]**

zob.

WEISSBERG Alexander

Wielka czystka

Warszawa : "Czytelnik", 1990

HERLING-GRUDZIŃSKI Gustaw **[tł]**

zob.

WEISSBERG Alexander

Wielka czystka

Warszawa : "Czytelnik", 1990

SZWARCMAN-CZARNOTA Bella **[tł]**

zob.

VAN HAMME Jean

Largo Winch i Grupa W

Warszawa : "Orbita", 1991

TEX w formatach

Pełne karty katalogowe. Pełne opisy książek bywają dość rozbudowane i nie mieszczą się na jednej kartce. W takim przypadku powinien powstać ciąg kartek złożony z kartki *wiodącej* i jednej lub kilku kartek *kontynuacyjnych*. Wymaga się ponadto aby kartki stanowiące jeden opis książki miały taki sam tekst nagłówkowy (zwykle składający się z sygnatur i hasła) oraz żeby były ponumerowane numerkami umieszczonymi w lewym górnym rogu kartki. Na kartce wiodącej taki numerek ma się pojawić tylko wtedy, gdy jest przynajmniej jedna karta kontynuacyjna.

Dla sprostania powyższym wymaganiom zostały zaprojektowane makro-definicje:

`\dglowka{`*tekst nagłówkowy*`}`, `\dkloc{`*tekst*`}`, `\dkoniec`

Teraz zadaniem formatu MAKD jest produkowanie opisów kart katalogowych w postaci następujących ciągów makr:

`\dglowka{`*tekst nagłówkowy*`}`, `\dkloc{`*tekst*`}`, ..., ..., `\dkloc{`*tekst*`}`, `\dkoniec`

Makro `\dglowka` przechowuje tekst nagłówkowy w pudełku (`\box`), które zostanie wykorzystane przez makro `\dkloc`, gdy zajdzie potrzeba utworzenia karty kontynuacyjnej. Makro `\dkloc` magazynuje kolejne *teksty* w pudełku aż do momentu kiedy dodanie kolejnego tekstu spowoduje przekroczenie ustalonej wysokości pudełka zbiorczego. Wtedy dotychczas zebrany w pudełku zbiorczym tekst zostanie wyeksportowany (`\eject`), a krytyczny *tekst* wstawiony do opróżnionego pudełka zbiorczego.

Polecenie `\dkoniec` powoduje natychmiastowe opróżnienie (wyeksportowanie) pudełka zbiorczego i zakończenie cyklu.

Makro `\dkloc` rozróżnia przypadek kartki wiodącej i kontynuacyjnej. Przy tej ostatniej, przed opróżnieniem pudełka zbiorczego opróżnia się pudełko `\glowka` magazynujące tekst nagłówkowy.

A oto omawiane makra „w naturze":

```
\newbox\glowka
\newbox\wkart \newbox\tkart
\newbox\bkloc
\newcount\numer \numer=0
\newcount\nkre \nkre=0
\newdimen\robdim \newdimen\vkart
\vkart=51mm  % wartosc graniczna wysokosci
             % tekstu na kartce
\def\dkloc#1{\setbox\bkloc=\vbox{#1}
                  \robdim=\ht\wkart
             \advance\robdim by \dp\wkart
             \advance\robdim by \ht\bkloc
             \advance\robdim by \dp\bkloc
             \ifnum\numer>0
              \advance\robdim by \ht\glowka
              \advance\robdim by \dp\glowka\fi
   \ifdim\robdim>\vkart
     \ifnum\numer=0
      \dnum\smallskip\unvbox\wkart\vfill\dkre
                                   \eject
      \setbox\wkart=\box\bkloc
     \else
      \dnum\smallskip\copy\glowka\smallskip%
             \unvbox\wkart\vfill\dkre\eject
     \setbox\wkart=\box\bkloc\fi
   \else
    \setbox\tkart=\vbox{\unvbox\wkart%
                  \smallskip\unvbox\bkloc}
    \setbox\wkart=\box\tkart\fi}
%
\def\dnum{\advance\numer by 1
        {\par\noindent\it\the\numer}}
%
\def\dkre{\relax}
%
\def\dkoniec{\ifnum\numer=0\relax\unvbox\wkart%
                         \vfill\dkre\eject
             \else\dnum\smallskip\unvbox\glowka%
                                   \smallskip
                  \unvbox\wkart\vfill\dkre\eject\fi
             \numer=0}
%
\def\dglowka#1{\setbox\wkart=%
\vbox{\hrule height 0pt depth 0pt width \hsize}
\setbox\glowka=\vbox{#1\hrule}}
```

Formaty produkujace tabelki. Wiele formatów produkuje *tabelki*, a więc ciąg rubryk poziomych podzielonych na kolumny, w które wkładane są treści pobierane z podpól bazy. Zwykle z jednego dokumentu bazy powstaje jedna lub kilka rubryk. Rubryki te na ogół później muszą być uporządkowane według jakiegoś klucza. Dlatego, jest istotnym aby poszczególne rubryki były samodzielnymi tabelkami, które można posortować by później złożyć z nich jedną zbiorczą tabelkę.

Zastosowane makra są analogiczne do makr zastosowanych do produkcji pełnych kart katalogowych. Są to makra:

`\eglowka`, `\ekloc{`*rubryka*`}`, `\dkoniec`

Zadaniem formatu MAKD jest produkowanie tabelki w postaci następującego ciągu makr:

`\eglowka`, `\ekloc{`*rubryka*`}`, ...,
`\ekloc{`*rubryka*`}`, `\ekoniec`

Tutaj *rubryka* reprezentuje opis jednej rubryki poziomej tabelki zbiorczej. Z punktu widzenia TEX-a *rubryka* jest samodzielną tabelką utworzoną z makr `\kol{`*szer*`}{`*pudełko*`}`, gdzie: *szer* jest wymiarem (`\dimen`) określającym szerokość kolumny w milimetrach, a *pudełko* – zawartość kolumny (zwykle akapit).

Dokładniej, struktura *rubryki* jest następująca:

```
\halign{&\vrule#&\strut\ #\ \cr
&\kol{szer}{pudełko}  &&
&\kol{szer}{pudełko}  &&
                ...
&\kol{szer}{pudełko}  & \cr
\noalign{\hrule}}
```

Makro `\eglowka` definiuje *pudełko* zawierające rubrykę nagłówkową umieszczaną jako pierwsza na każdej stronie tabelki. Rubryka nagłówkowa zbudowana jest z makr `\kolh` i `\kolhh`, które są szczególnymi przypadkami ogólniejszego makra `\kol`.

A oto omawiane makra „w naturze":

```
\newbox\glowka
\newbox\wkart \newbox\tkart
\newbox\bkloc
\newcount\numer \numer=0
\newdimen\robdim \newdimen\vkart \vkart=180mm
%
\def\eglowka{\setbox\wkart=\vbox{\hrule height 0pt
      depth 0pt width \hsize}
\setbox\glowka=\vbox{%
\halign{&\vrule##&\strut\ ##\ \cr
\noalign{\hrule height1pt}
&\kolhh{19}{Nr inwent.}{Sygnatura} &&
\kolhh{13}{Data}{wpisu} &&
\kolhh{70}{Autor}{Tytuł} &&
\kolhh{19}{Miejsce wyd.}{Rok wydania} &&
\kolh{25}{Wydawca} &&
\kolh{18}{Nr akcesji} &&
\kolhh{15}{Sposób}{nabycia} &&
\kolhh{17}{Cena lub}{wartość} &&
\kolhh{18}{Nr}{ubytku} &&
\kolh{18}{Uwagi} &
\cr
```

```
\noalign{\hrule height1pt}}
}}
%
\def\ekoniec{%
\halign{&\vrule##&\strut\ ##\ \cr
\noalign{\hrule height1pt}}
\unvbox\glowka\unvbox\wkart\vfill\eject}
%
\def\ekloc#1{\setbox\bkloc=\vbox{#1}
             \robdim=\ht\wkart
             \advance\robdim by \ht\bkloc
             \advance\robdim by \ht\glowka
   \ifdim\robdim>\vkart
     {\offinterlineskip%
     \copy\glowka\unvbox\wkart}\vfill\eject
     \setbox\wkart=\box\bkloc
   \else
    \setbox\tkart=\vbox{\unvbox\wkart\unvbox\bkloc}
    \setbox\wkart=\box\tkart\fi}
%
\def\kol#1#2{\vtop{\hsize=#1mm\vskip.1mm%
\noindent\baselineskip=10pt%
\rightskip0pt plus 1fil{#2}\vskip1mm}}
%
\def\kolh#1#2{\vtop to 10mm{\vfil\hbox
            to #1mm{\hfil #2\hfil}\vfil}}
\def\kolhh#1#2#3{\vtop to 10mm{\vfil\hbox
            to #1mm{\hfil #2\hfil}%
\vfil\hbox to #1mm{\hfil #3\hfil}\vfil}}
```

Formaty typu „Ankieta". Jeżeli zadaniem formatu jest rozmieszczenie tekstów pochodzących z dokumentu bazy w nieregularnie rozlokowanych miejscach na stronie (jak to bywa w przypadku wypełniania ankiet bądź formularzy) to do opracowania takiego formatu przydatna będzie następująca makro-definicja:

```
\def\wxyb#1#2#3{\vbox to 0pt{\hbox to 0pt{%
\hbox{\strut\hskip#1mm\vtop{%
\hbox{\ }\vskip#2mm{#3}}}\hss}\vss}}
```

W tej makrodefinicji: `#1` określa współrzędną poziomą, zaś `#2` współrzędną pionową punktu odniesienia tekstu określonego przez `#3`. Parametr `#3` to tekst (zwykle w postaci *pudełka*).

Powyższą makrodefinicję wygodnie jest skojarzyć z makrodefinicją definiującą *pudełko*, które będzie użyte jako 3-ci parametr makra `\wxyb`. Może to być makrodefinicja podobna do następującej:

```
\newbox\hcxxx
\def\opxxx#1{\setbox\hcxxx=\vbox{\hsize=45mm%
\noindent\rightskip0pt plus 1 fill{\eight #1}\par}}
```

Poniższe dane dla TEX-a zostały wyprodukowane przez format-D rozważanego typu:

```
\hsize=75mm
\vsize=125mm
%
\def\wxyb#1#2#3{\vbox to 0pt{\hbox to 0pt{%
\hbox{\strut\hskip#1mm\vtop{%
\hbox{\ }\vskip#2mm{#3}}}\hss}\vss}}
%
\overfullrule=0pt\footline{\hfil}
%
\font\eight=plss8
\font\kurs=plti7 scaled\magstep0
\font\gkur=plbxti10 scaled\magstep0
%
\newbox\hcaut\newbox\hctyt\newbox\hcdzi
\newbox\hcdat\newbox\hccen\newbox\hclin
\newbox\hclins\newbox\hglin\newbox\vglink
%
\def\opdat#1{\setbox\hcdat=\hbox
to 45mm{{\eight #1}\hfil}}
\def\opcen#1{\setbox\hccen=\hbox
to 45mm{{\eight #1}\hfil}}
\def\opdzi#1{\setbox\hcdzi=\hbox
to 22mm{{\eight #1}\hfil}}
\def\opaut#1{\setbox\hcaut=\hbox
to 45mm{{\gkur #1}\hfil}}
\def\optyt#1{\setbox\hctyt=\vbox{\hsize=45mm%
\noindent\rightskip0pt plus 1 fill{\eight #1}\par}}
\def\oplin{\setbox\hclin=%
\hbox to 70mm{\vrule depth2mm
height-1.8mm width69mm\hfil}}
\def\oplins{\setbox\hclins=%
\hbox to 70mm{\vrule depth2mm
height-1.8mm width22mm\hfil}}
\def\oplig{\setbox\hglin=%
\hbox to 70mm{\vrule depth2mm
height-1.6mm width69mm\hfil}}
\def\vklig{\setbox\vglink=%
\hbox to .5mm{\vrule depth19mm
height0pt width.4mm\hfil}}
%====================================
\offinterlineskip
\oplin\wxyb{1}{11}{\box\hclin}
\oplins\wxyb{1}{18}{\box\hclins}
\oplin\wxyb{1}{25}{\box\hclin}
\oplig\wxyb{1}{30}{\box\hglin}
\vklig\wxyb{23}{13}{\box\vglink}
\opaut{Boczkowski P.\unskip, [2 in.]}%
\wxyb{25}{9}{\box\hcaut}
\optyt{Zadania eksperymentalnych badań nad
społecznym systemem osiedla mieszkaniowego}%
\wxyb{25}{16}{\box\hctyt}
\opdzi{ + 215d}\wxyb{25}{33}{\box\hcdzi}
\opdat{{\kurs Data wł. do biblioteki:} K/27/98}%
\wxyb{25}{29}{\box\hcdat}
\opcen{{\kurs Uwagi}: 30,00 {\kurs zł.}}%
\wxyb{1}{33}{\box\hccen}
\vfill\eject
\oplin\wxyb{1}{11}{\box\hclin}
\oplins\wxyb{1}{18}{\box\hclins}
\oplin\wxyb{1}{25}{\box\hclin}
\oplig\wxyb{1}{30}{\box\hglin}
\vklig\wxyb{23}{13}{\box\vglink}
\optyt{Z różańcem w dłoniach}%
\wxyb{25}{16}{\box\hctyt}
\opdat{{\kurs Data wł. do biblioteki:} K/35/91}%
\wxyb{25}{29}{\box\hcdat}
\opcen{{\kurs Uwagi}: 25,00 {\kurs zł.}}%
\wxyb{1}{33}{\box\hccen}
\vfill\eject
%===================================
\end
```

Z powyższego tekstu TEX wyprodukuje następujące dwie tzw. karty książki (pokazujemy poniżej same nagłówki tych kart):

Boczkowski P., [2 in.]

Zadania eksperymentalnych badań nad społecznym systemem osiedla mieszkaniowego

Data wł. do biblioteki: K/27/98

Uwagi: 30,00 *zł.* + 215d

Z różańcem w dłoniach

Data wł. do biblioteki: K/35/91

Uwagi: 25,00 *zł.*

Programy wspomagające

Program MAKD, sterowany *formatem-D* i formatujący dane pobierane z bazy *B*, wytwarza na wyjściu plik `B.tex`. Z reguły, dane zawarte w tym zbiorze – nim zostaną przekazane do przetworzenia procesorowi TEX – podlegają dalszemu przetwarzaniu przez pewne programy wspomagające.

W tym dodatkowym przetwarzaniu, prawie zawsze bierze udział program ZAMIANY, który – sterowany odpowiednio przygotowaną *listą zamian* – dokonuje szeroko pojętego przekodowania danych. W szczególności, dokona się tu przekodowanie na zapis TEX-a obcych liter akcentowanych, liter greckich, znaków specjalnych mających w MAK-u własny sposób kodowania.

W wielu zastosowaniach, konieczne jest przesortowanie materiału pobranego z bazy według określonych kryteriów. Tę funkję wykonuje program SKSORT, który omówimy niżej. Np. etap sortowania występuje w procesie konstruowania KSIĘGI INWENTARZOWEJ, w której rubryki poziome muszą być uporządkowane według numerów inwentarza.

Dalej, wiele operacji na tekście może być dokonanych dopiero po wykonaniu sortowania. Na przykład, różne operacje arytmetyczne (obliczanie sum częściowych czy innych rodzajów obliczeń statystycznych odpowiednio zgrupowanych danych) mogą być teraz wykonywane za pomocą programu STATY. Innym przykładem będzie grupowanie numerów pozycji w różnego rodzaju indeksach (program REDINDD).

Warto tutaj zwrócić uwagę na fakt, że działanie programów wspomagających musi być przewidziane podczas opracowywania *formatu-D*. Dane, które pojawią się na wyjściu programu MAKD muszą być tak skonstruowane aby działanie programów wspomagających prowadziło do pożądanych efektów. W szczególności, dane te powinny zawierać instrukcje sterujące programami, które będą je przetwarzać.

Uwagi końcowe

Przedstawiona tu prezentacja mechanizmów programowych służących do przetwarzania danych z baz MAK do postaci TEX-a jest oczywiście niekompletna i nie może służyć jako dokumentacja. Pełna specyfikacja omawianych tutaj programów zawarta jest w [4] i [5].

Przedstawione przykłady są bardzo uproszczone. W życiu, jak wiadomo, wszystko jest znacznie bardziej skomplikowane. Tym niemniej, powstało wiele pakietów, stanowiących uzupełnienie do pakietu MAK, umożliwiających zautomatyzowane przetwarzanie odpowiednio skonstruowanej bazy MAK na materiał do książki zapisany w TEX-u. Oczywiście, z takiego materiału książka sama nie wyskoczy. Musi się nim zająć jakiś wydawca. Tu właśnie zaczynają się „schody”, bo większość wydawców, jeżeli nawet o TEX-u słyszeli, to że jest to coś trudnego, nieprzyjaznego i niesympatycznego.

Potrzeba zatem, ujawnienia się wydawców, którzy chętnie posługują się TEX-em, bądź TEXpertów, którzy mieliby ochotę pośredniczyć między bibliotekami przygotowującymi publikacje przy pomocy pakietu MAK a wydawcami, którzy TEX-a nie znają. Tych bibliotek może być dużo, bo pakiet MAK (też podobno „nieprzyjazny”) bardzo się rozpowszechnił w bibliotekach w całej Polsce. A – podobno – wszystkie powiatowe biblioteki publiczne przystąpiły do opracowywania bibliografii regionalnej dla swojego regionu.

Bibliografia

1. Bartoszewicz-Fabiańska, Bożena [et al.]: Adaptacja formatu MARC BN dla potrzeb bibliografii regionalnej. Wydaw. SBP, Warszawa 1999.
2. Knuth, Donald E.: The TEXbook. Addison-Wesley, Reading, MA 1994.
3. Maj, Jerzy: Elementarz MAK-a dla Bibliotekarzy. Wydawnictwo SBP, Warszawa 1999.
4. Swianiewicz, Jerzy: Pakiet MAK – edycja i drukowanie. BN, Warszawa 2001.
5. Wierzbowski, Jan: Pakiet MAK – obsługa bazy. BN, Warszawa 1999.

Jerzy Swianiewicz
Biblioteka Narodowa, Warszawa
`jswian@bn.org.pl`

Dodatek A. Formaty-D w postaci źródłowej

Format-D PR01. Format podany został w trzech odcinkach ponieważ odcinek środkowy jest wspólny z formatem-D PR02 pokazanym w następnym paragrafie. Komentarze na końcu.

PR01. Odcinek 1.

```
$$$
%********** PROLOG<crlf>
\hoffset=-15mm\voffset=-10mm<crlf>
\hsize=110mm\vsize=75mm<crlf>
\tolerance=9900<crlf><K>
<PD><Ozer><4>%******* POCZĄTEK DOKUMENTU: <Oddr><0><crlf><K>
==================
<Np>023<K>           * POLE 023 *
<PjK><Orow><4><0>\rightline{\bf ...}<crlf><sep>\vskip2mm<crlf><K>
<PwoK><Orow><4><0>\rightline{\bf ...}<crlf><sep>\vskip2mm<crlf><K>
<Pbr>\rightline{\bf ...}<crlf>\vskip2mm<crlf><K>
+++++
<Npp>s<K>            * podpole s pola 023 *
<PPj>\rightline{\bf <Dpp>}<crlf><Onast><4><K>
<PPwp>\rightline{\bf <Dpp>}<crlf><Onast><4><K>
<PPws>\rightline{\bf <Dpp>}<crlf><Onast><4><K>
<PPwo>\rightline{\bf <Dpp>}<crlf><Onast><4><K>
```

PR01. Odcinek 2.

```
==================
<Np>@brak@<K>         * POLE BRAKUJĄCE *
<Pbr><Ozer><4>\noindent{\bf<crlf><K>
==================
<Np>100<K>            * POLE 100 *
<PwsI><Owie><4><0>\unskip, <sep><K>
<PwoI><Owie><4><0>\unskip, <sep><K>
+++++
<Npp>1<K>             * podpole 1 pola 100 *
<PPj>\uppercase{<Dpp>} <Osre><4><1><K>
<PPwp>\uppercase{<Dpp><Osre><4><1><K>
<PPws><Dpp><K>
<PPwo><Dpp>} <K>
+++++
<Npp>2<K>             * podpole 2 pola 100 *
<PPj><Dpp> <Osre><4><1><K>
<PPwp><Dpp> <Osre><4><1><K>
<PPws><Dpp> <K>
<PPwo><Dpp> <K>
==================
<Np>200<K>            * POLE 200 *
<PjI><Oskp><4><1><K>
<PwpI><Oskp><4><1><K>
<PwsI><Oskp><4><1><K>
<PwoI><Oskp><4><1><K>
+++++
<Npp>a<K>             * podpole a pola 200 *
<PPj>\uppercase{<Bpp><Ofun><1><2><Owie><20><0> ...<sep>}<Onast><4><K>
<PPwp>\uppercase{<Bpp><Ofun><1><2><Owie><20><0> ...<sep>}<Onast><4><K>
==================
<Np>@brak@<K>         * POLE BRAKUJĄCE *
<Pbr><Owie><4><0>}\par<crlf>
<el>\noindent{\bf Brak hasła}\par<crlf><sep>
<Ozer><4>\vskip2mm<crlf><K>
==================
<Np>200<K>            * POLE 200 *
<PjK><Owie><4><0>\par<crlf><sep><K>
```

```
55. <PwsI><Owie><4><0>\unskip; <sep><K>
56. <PwoI><Owie><4><0>\unskip; <sep><K>
57. <PwoK><Owie><4><0>\par<crlf><sep><K>
58. +++++
59. <Npp>a<K>            * podpole a pola 200 *
60. <PPj><Dpp><Onast><4><K>
61. <PPwp><Dpp><Onast><4><K>
62. <PPws>\unskip. <Dpp><K>
63. <PPwo>\unskip. <Dpp><K>
64. +++++
65. <Npp>e<K>            * podpole e pola 200 *
66. <PPj> : <Dpp><Onast><4><K>
67. <PPwp> : <Dpp><Onast><4><K>
68. <PPws> : <Dpp><K>
69. <PPwo> : <Dpp><K>
70. ==================
71. <Np>@brak@<K>        * POLE BRAKUJĄCE *
72. <Pbr><Ozer><4>\vskip1mm<crlf><K>
73. ==================
74. <Np>210<K>           * POLE 210 *
75. +++++
76. <Npp>@kazde@<K>      * dowolne podpole pola 210 *
77. <PPj><Dpp>\par<crlf><Onast><4><K>
78. <PPwp><Dpp><Onast><4><K>
79. <PPws><Ozirow><5><1>c : <Dpp><sep><Ozirow><5><1>d\unskip, <Dpp><sep><K>
80. <PPwo><Ozirow><5><1>c : <Dpp><sep><Ozirow><5><1>d\unskip, <Dpp><sep>
81. \par<crlf><K>
```

PR01. Odcinek 3.

```
82. ==================
83. <Np>@brak@<K>        * POLE BRAKUJĄCE *
84. <Pbr><Ozer><4><K>
85. ==================
86. <KD>%*******    KONIEC DOKUMENTU: <Oddr><0><crlf>\vfill\eject<crlf><K>
87. <EP>%*******    EPILOG<crlf>\end<crlf>
```

Komentarze, objaśnienia do formatu-D PR01.

Postać źródłowa: `OPTXPR01.SFO`
Postać wykonywalna: `OPTXPR01.TTT`

Wiersze: 7, 12, 18, 21, 25, 31, 37, ... oraz teksty na prawo od `<K>` – stanowią *komentarz*

- 2–5: PROLOG
- 6–86: CZĘŚĆ CENTRALNA
- 87: EPILOG

========

- 8: Tekst interpretowany na początku każdego dokumentu

———

- 8–17: Opracowanie pola 023
- 13–17: Opracowanie podpola 023/s (sygnatura)

———

- 22–36: Opracowanie pola 100 (autor)
- 26–30: Opracowanie podpola 100/1 (nazwisko)
- 32–36: Opracowanie podpola 100/2 (imię)
- 38–46: Opracowanie pola 200 (tytuł)
- 44–46: Opracowanie podpola 200/a

———

- 53–69: Opracowanie pola 200 (ponowne)
- 59–63: Opracowanie podpola 200/a (tytuł właściwy)
- 65–69: Opracowanie podpola 100/2 (dodatek do tytułu)

———

- 74–81: Opracowanie pola 210 (adres wydawniczy)
- 76–81: Opracowanie wszystkich podpól pola 210

———

- 86: Tekst interpretowany na końcu każdego dokumentu

Wiersze 20, 49–51, 72, 84 – zawierają teksty interpretowane podczas opracowywania każdego dokumentu.