

ESCRIBA

— Um Formatador de Texto Lições de uma Experiência

J. A. Legatheaux Martins
UNL

RESUMO: Neste relatório é descrito um conjunto de programas para processamento de texto. Depois de um enquadramento geral do problema, são apresentadas as soluções adoptadas e seu respectivo balanço. Bastante atenção é dada à portabilidade do sistema quer em relação ao código Pascal, linguagem em que os programas estão escritos.

ABSTRACT: This paper describes a set of programs for text processing. After a general discussion on text processing the options we took and their implications are presented. Portability concerning computer peripherals and Pascal code, the language we have chosen for implementation, has been our main purpose.

1. Composição automática de documentos

Para os profissionais cuja actividade exige a constante edição e preparação de documentos a disponibilidade de meios eficazes para a sua produção é muito importante.

Em geral o recurso a métodos manuais permite obter uma grande qualidade final dos textos à custa de tempos de edição muitas vezes inoportáveis.

Factores de diversa ordem permitem antever uma situação em que a edição manual de documentos se tornará um luxo.

Entre estes, a subida do preço da mão de obra, a descida dos preços do material informático e o aparecimento de impressoras de computador com boa qualidade gráfica, são talvez os mais importantes.

Se nos restringirmos ao quadro das instituições de vocação para o ensino, investigação ou outra actividade técnico/científica, o seu pessoal tem necessidade crescente de apoio em processamento de textos.

Com a generalização dos micro computadores e as impressoras de agulhas de baixo custo e qualidade média, estes profissionais têm à sua disposi-

ção os meios materiais mínimos para automatizar a produção e edição de documentos.

Este relatório descreve um conjunto de programas que permite utilizar qualquer computador dispondo de compilador da linguagem Pascal e de uma impressora, como sistema de preparação de textos de qualidade média, adequado a suportar a actividade do pessoal com o perfil atrás referido na edição de documentação técnica.

2. Características de um sistema de composição e problemas que colocam

a) Objectos manipulados pelo sistema

Uma característica importante destes sistemas são os objectos que podem manipular.

Por um lado temos a considerar as entidades elementares: caracteres, palavras, índices em expressões matemáticas, componentes de figuras, etc. Por outro há que considerar as possibilidades de associação e estruturação de objectos elementares: linhas, parágrafos, páginas, expressões matemáticas, figuras, procedimentos, programas, etc.

Tradicionalmente os editores de texto apenas permitiam manipular sequências de caracteres estruturados em linhas. A tentativa de rentabilizar o trabalho dos utilizadores de sistemas interactivos leva à necessidade de se dispor de editores capazes de manipularem programas e suas componentes, palavras, linhas, figuras, expressões matemáticas, ...

Esta categoria de sistemas exige técnicas e periféricos especiais. O seu desenvolvimento é em parte tema de investigação para o qual a experiência na Computação Gráfica, Compilação, Sistemas interactivos e Linguagens de comandos é muito importante.

b) Características dos dispositivos de entrada

Nada impede que um sistema deste tipo receba dados de um leitor de cartões. É evidente que esse não é o meio ideal.

Se o sistema apenas processa caracteres um terminal alfanumérico é o dispositivo mínimo que se exige. Se se pretender ir mais longe é bastante recomendável que se utilize um terminal com possibilidades gráficas.

Os protocolos de diálogo entre este tipo de terminais e o computador por um lado, e entre o terminal e o seu utilizador por outro, não estão standardizados e por esse motivo estes sistemas colocam grandes problemas de portabilidade.

c) Características dos dispositivos de saída

Se se pretende que o sistema apenas manipule linhas de texto uma impressora é suficiente. Mesmo assim há que tomar em consideração as suas possibilidades de sublinhar caracteres e de colocar acentos para que se obtenham resultados de melhor qualidade.

Se se pretende manipular outros objectos mais complexos, como expressões matemáticas e figuras, são necessários periféricos mais sofisticados.

Mesmo a grande maioria das impressoras com algumas possibilidades interessantes (por exemplo através de «escape sequences») começam a colocar problemas de portabilidade do software que as utiliza.

d) Características de ergonomia

Os sistemas de processamento de texto são confrontados com categorias muito heterógenas de utilizadores de formações e hábitos muito diversificados.

Por este motivo as características ergonómicas destes sistemas adquirem grande importância na sua apreciação; nomeadamente, a sua robustez, fiabilidade, facilidade de guiarem o utilizador na sua própria aprendizagem, facilidade de utilização, etc.

São preferíveis os sistemas que permitem ao utilizador visualizar directa e imediatamente o resultado final do seu trabalho. Também são preferíveis os sistemas dispostos de menus e teclados de funções ao invés de sistemas com intrincadas linguagens de comandos baseados em mnemónicas.

As características de ergonomia do sistema estão em grande relação quer com os dispositivos de entrada, quer com as técnicas de Engenharia de Software utilizadas no seu desenho.

e) Independência dos periféricos

Em geral é desejável que o sistema não esteja demasiado dependente do computador e periferia utilizados (tal não é a opinião da maioria dos fabricantes). Tal objectivo é em geral contraditório com o grau de sofisticação do sistema. Mais uma vez é necessário fazer apelo à experiência da Computação Gráfica e da Engenharia de Software para que o sistema resultante seja portátil e adaptável.

3. Objectivos a atingir

Colocados perante este conjunto de problemas optamos por definir um conjunto de objectivos que permitissem descortinar quais as soluções mais adequadas.

- a) O utilizador típico do sistema é o pessoal técnico/científico e respectivo pessoal de apoio.

A documentação a editar será em geral documentação técnica.

- b) O sistema deverá poder ser implementado em qualquer computador disponível (micros de 8 bits incluídos) desde que este disponha de editor de textos, compilador da linguagem em que está escrito e impressora (mesmo de baixa qualidade).
- c) O sistema deverá ser fácil de aprender para que seja preferível utilizá-lo a aprender a trabalhar com toda uma multiplicidade de sistemas com a mesma função e disponíveis no mercado.
- d) Dentro das suas limitações o sistema deverá ser o mais ergonomicamente adequado que possível privilegiando o utilizador que pretende obter textos sem efeitos especiais.

4. Opções tomadas

Em função dos objectivos e problemas focados foram tomadas as opções que passaremos a enunciar:

- a) O sistema tem a forma de um processador de texto tendo como entrada um ficheiro contendo o texto a formatar intercalado com comandos que dirigem a formatação. Para criar o texto a formatar o utilizador usa o editor de textos disponível no seu sistema. Desta forma garante-se a independência em relação aos dispositivos de entrada.

Se esta opção tem a desvantagem de impedir a visualização imediata do resultado final, tem pelo menos a vantagem de dispensar a aprendizagem de mais outro editor para o utilizador normal do computador em que o sistema está instalado.

Optou-se também por definir o sistema como um processador de textos cujos efeitos mais sofisticados são o sublinhar e colocação de acentos.

- b) No que toca ao dispositivo de saída optou-se por fixar que este é uma impressora convencional. Para se conseguirem certos efeitos pressupõe-se que a mesma pode executar a operação de puxar a cabeça atrás em resposta à recepção do carácter «backspace».

Veremos adiante que a forma concreta como os acentos e sublinhados são colocados é na prática irrelevante para o formatador, pois outro programa permite virtualizar a impressora utilizada.

- c) Como linguagem de implementação seria necessário seleccionar uma linguagem amplamente disponível e cujas características permitam a manipulação de texto sem o recurso a extensões não portáveis.

A primeira característica aponta para o Basic, Fortran ou Pascal. A segunda leva à eleição do Pascal como linguagem de implementação. A seguir se descreverão as técnicas que foram utilizadas para que o sistema seja portátil.

5. Decisões no que toca à ergonomia

A grande maioria dos formatadores disponíveis baseiam-se no conceito de comando, isto é, palavra reservada que sempre que é encontrada no texto condiciona o processamento subsequente.

A aproximação normal consiste em considerar que sempre que um comando é encontrado, um parâmetro de condicionamento da formatação é redefinido ficando com o novo valor até uma nova redefinição.

Esta aproximação tem, quanto a nós, diversos efeitos que passaremos a enunciar:

- a) Confusão entre directivas de efeito localizado e directivas de efeito global. Isto é, os parâmetros que guiam a formatação são localmente alterados para efeitos localizados mas têm de ser restaurados para os seus anteriores valores, sob pena de um efeito que se pretende local afectar todo o subsequente processamento.
- b) O texto a processar não tem qualquer estrutura com excepção da que lhe é conferida pelas palavras e linhas.
- c) A redefinição de qualquer parâmetro pré-definido exige a inserção no texto a processar dos comandos adequados. Se se pretende mais tarde modificar o estilo gráfico do texto impresso é necessário modificar o texto a processar.

No sentido de obviar a alguns destes inconvenientes decidimos tomar as seguintes opções:

- a) Os parâmetros de formatação são globais e redefiníveis independentemente do texto a processar. Escriba é automaticamente inicializado por um ficheiro onde o seu utilizador fixa o estilo gráfico do resultado. É possível definir independentemente do texto a formatar características como:
 - Espaçamentos entre os diversos tipos de parágrafos.
 - Espaçamentos entre linhas de diversos tipos.
 - Marginações das páginas pares e ímpares.
 - Estilo e posição da numeração.
 - Paginação e aproveitamento da página em geral.
 - Se se admitem sublinhados e outros efeitos especiais e forma concreta que assumem.
 - Os caracteres com significado pré-definido.
- b) O texto a processar está subdividido em blocos (parágrafo em geral). Um bloco de uma forma geral é constituído por uma sequência contígua de linhas não vazias. Linhas em branco separam implicitamente os blocos. O bloco sem qualquer tipo de tratamento especial é um parágrafo a justificar à esquerda e à direita.

O alcance de qualquer comando restringe-se ao bloco que prefixa ou que o contém. Não existe o conceito de parâmetro corrente redefinível sempre que se deseje. No fim de cada bloco todos os parâmetros adquirem o valor que lhes foi definido em tempo de inicialização.
- c) O bloco sem prefixo é um parágrafo normal. A edição de um romance quase que não precisa

de comandos. Os blocos para os quais o utilizador pretende formas gráficas especiais são prefixados por um comando que indica que o próximo parágrafo é especial.

Existem dez prefixos (comandos) e um total de cerca de doze tipos diferentes de parágrafo. Alguns dos parágrafos de utilização mais frequente são: parágrafos a centrar, títulos a centrar, títulos a sublinhar, parágrafos com títulos em evidência à esquerda, parágrafo com marginação especial, etc.

- d) A presença de certos caracteres especiais actua como indicador no parágrafo que os contém, por exemplo forçando espaço em branco suplementar, marcando o início ou o fim de uma frase a sublinhar ou ainda indicando que o carácter que se segue deve ser impresso sobre o carácter que o antecede. (efeito conseguido com «backspace» na dactilografia convencional).

É evidente que este tipo de opções favorece o utilizador que pretende um estilo gráfico uniforme e penaliza o utilizador que pretende um texto com muitos efeitos especiais (muito recortado). O leitor interessado deve consultar [1].

6. Virtualização da impressora

A portabilidade de Escriba acenta em quatro pilares:

- a) O facto de ser um processador de texto.
- b) O seu ficheiro de inicialização.
- c) A forma como foi escrito em Pascal.
- d) A virtualização da impressora.

No que toca a a) já o discutimos atrás. No que toca a b) diremos apenas que certos parâmetros redefiníveis podem inibir efeitos especiais difíceis de conseguir ou mesmo impossíveis em certas impressoras. No que toca a c) será objecto de um próximo ponto.

No que toda a d), a solução adoptada pode ser descrita em traços largos da seguinte forma:

Pressupõe-se que o dispositivo de impressão é equivalente a uma máquina de escrever de qualidade média, isto é, permite a impressão de linhas com um conjunto constante de caracteres, sem caracteres especiais do alfabeto português, capaz de puxar a cabeça atrás por recepção de «backspace» e com espaçamento constante.

Se o conjunto de possibilidades é bastante reduzido tem em contra-partida a vantagem de se encontrar disponível em qualquer impressora de agulhas de baixo custo com excepção do efeito de puxar a cabeça atrás.

Este problema pode ser resolvido de duas formas pelo utilizador:

- a) Inibindo na parametrização todos os efeitos que implicam a emissão de «backspace» sem que isso signifique que deixa de poder sublinhar os títulos.

- b) Processando o ficheiro produzido por Escriba e contendo «backspace» por um programa especial, Lpsback [2]. Este programa é no seu essencial semelhante a um «device driver» facilmente adaptável e que simula na impressora disponível o efeito dos «backspaces».

Se à primeira vista e após cerca de quatro meses de utilização as opções tomadas no que toca aos aspectos ergonómicos e aos dispositivos de entrada se têm até ao momento revelado adequadas, é essencialmente no que toca à virtualização da impressora que nos parece que melhores resultados poderiam ser obtidos. Com a generalização das impressoras de cabeça de agulhas com efeitos especiais, certas facilidades interessantes destas máquinas deveriam ter sido virtualizadas para que Escriba não as despreze sistematicamente.

Utilizando técnicas bem conhecidas da Computação Gráfica e ampliando as possibilidades de Lpsback seria possível dar mais sofisticação à Escriba sem que isso sacrificasse a sua portabilidade. Seria pelo menos interessante utilizar as possibilidades de mudança de tipo e espaçamento horizontal que se começam a generalizar em impressoras de baixo custo.

7. Portabilidade do código Pascal

No que a seguir se apresenta, standard refere-se à linguagem definida em [4], já que o standard ISO do Pascal foi aprovado em 1981 e a grande maioria dos compiladores disponíveis não lhe obedece.

O objectivo ideal a atingir no que toca à portabilidade do código deveria ser: o programa é instalável sobre qualquer sistema de operação sem nenhuma modificação. Este objectivo tem que ser em geral substituído por um mais modesto: o programa é instalável sobre qualquer sistema de operação munido de compilador da linguagem em que está escrito e após pequenas modificações, localizadas, documentadas, já previstas e realizáveis por qualquer programador que ignore o funcionamento do programa em pormenor.

No que se refere a programas escritos em Pascal os problemas de transporte tem diversas origens. Podemos dividi-las em vários grupos:

- a) Utilização de extensões não standard. É evidente que a utilização deste tipo de extensões deve ser proibida quando se pretende escrever programas transponíveis.
- b) Questões deixadas propositadamente indefinidas no standard. Exemplo: forma de estabelecer a associação entre um parâmetro do programa e um ficheiro do sistema de operação.
- c) Particularidades impostas pelo conjunto de caracteres do computador usado e forma de tratamento dos caracteres em geral (e dos de controlo em particular) pelas funções e procedimentos standard: eof, eoln, ord, chr, put, get, page, etc.
- d) Limitações impostas por um dado sistema e relacionadas com a forma como os diferentes tipos de dados são representados interna-

mente. Exemplos: cardinalidade dos set's, valor de MAXINT, etc.

- e) Restrições ao Pascal impostas por um compilador particular. Exemplo: uma variável utilizada no índice de uma interação de um for statement pode em Pascal ser qualquer enquanto que em certos compiladores se exige que tenha sido declarada no «scope» corrente.
- f) Desvios em relação ao standard. Exemplos não faltam.

Técnicas a usar a fim de evitar a escrita de código Pascal dificilmente transportável são no que se refere a alguns dos grupos anteriores as seguintes:

- b) Alguns dos problemas e respectivas técnicas a aplicar são as seguintes:
 - A grande maioria dos compiladores não distingue mais do que 8 caracteres num identificador. Evite ultrapassar este limite.
 - O código para associar uma variável de tipo ficheiro a um nome de ficheiro no sistema de operação deve ser isolado em procedimentos documentados e assinalados como não transportáveis. Esta observação também é válida para o «fecho» de ficheiros.
 - Quando o programa dialoga com o utilizador pelo terminal é necessário ler e escrever caracteres do e para o mesmo. Diversas variantes da forma como os caracteres são «buferezados» e mostrados ao utilizador pelos diferentes sistemas de operação, assim como os problemas relacionados com os chamados «ficheiros interactivos» implicam que essas zonas de diálogo sejam não transportáveis e portanto se devam adoptar soluções idênticas às referidas atrás.
 - A forma de abortar abruptamente a execução de um programa pode ser: executando um GOTO para fim do mesmo, chamado um procedimento pré-definido não standard (em geral HALT). A forma escolhida deve ser posta em evidência.
 - A forma de introduzir no programa as opções de compilação não é standardizada. Isole e documente as suas opções.
- c) As dependências do tipo dos caracteres, sua representação, código interno de representação e forma como os caracteres de controlo são tratados exige a utilização de técnicas adequadas como por exemplo:
 - Definir como funções isoladas e documentadas como não transportáveis os predicados: uma caracter é uma letra, um caracter deve ser interpretado como um banco no ficheiro de entrada, um caracter é um caracter que alguns sistemas não reconhecem, como por exemplo '(' e ')'.
 - Definir como procedimentos isolados e documentados como não transportáveis os procedimentos de transformação da letra grande em pequena e vice-versa.
 - Dar pelo menos em comentário uma redefinição do procedimento page para impressoras

sem «form-feed» se a paginação é importante para os seus resultados.

- Definir como constantes a ordem dos caracteres de controlo relevantes para o seu processamento.
 - Evitar usar os caracteres '(' e ')' nos comentários e comentários dentro de comentários.
- d) No que se refere às questões deste grupo algumas das técnicas são:
- Evitar usar set's de cardinalidade superior a 256 e seus construtores na forma valor .. valor.
 - Evitar usar case statements em que a dimensão do intervalo de variação da variável é superior a 256.
 - Evitar usar string's contendo caracteres especiais, de controlo ou de dimensão superior à largura do terminal.
 - Definir como constantes e pôr em evidência as constantes que são relevantes em algoritmos numéricos como: o maior real, o menor real, o intervalo que separa dois reais «muito próximos», etc.

8. Aspectos de concepção

ESCRIBA foi desenhado pelo método da programação pela sintaxe [3]. Assim ele reduz-se no essencial a um autómato finito com funções semânticas tendo por conjunto terminal o conjunto das linhas (classificadas em função de eventuais prefixos).

Tal como em [5] o método revelou-se adequado e praticamente todo o tempo dispendido em testes foi para afinar e modificar as acções semânticas.

Neste método de programação, a adequada escolha do conjunto terminal é vital. A solução adoptada foi analisada em alternativa com a escolha para conjunto terminal de um conjunto baseado em palavras.

Se a escolha por nós feita tem a vantagem de diminuir o número de estados do autómato, o que é importante quando não se dispõe de gerador automático do algoritmo, tem em contrapartida a desvan-

tagem de não fazer coincidir as unidades sintácticas elementares com as unidades semânticas mais frequentemente processadas. (a unidade semântica mais frequentemente tratada é a palavra).

Este aspecto é muito importante quando se utiliza aquele método e de momento ainda não temos ideias claras sobre se a opção tomada foi a mais adequada.

Lpsback é baseado num curioso algoritmo cuja verificação é apresentada em [2].

9. Agradecimentos

ESCRIBA e todos os outros programas aqui referidos foram desenvolvidos no âmbito do Projecto 3PES sobre Metodologia da Programação da Linha de Acção I do Centro de Informática da Universidade Nova de Lisboa suportado pelo INIC.

10. Referências

- [1] J. A. Legatheaux Martins. ESCRIBA — Um formatador de texto — Manual do utilizador. Relatório interno do Centro de Informática da Universidade Nova de Lisboa. CIUNL — 1/82. Lisboa. Fev. 1982.
- [2] —. LPSBACK — Um exercício de construção e verificação de um algoritmo por sucessivas transformações. Relatório interno do Centro de Informática da Universidade Nova de Lisboa. CIUNL — 6/82. Lisboa. Março de 1982.
- [3] —, Luís Monteiro. Linguagens Formais e autómatos. Texto de um curso. Universidade Nova de Lisboa. Departamento de Informática. Lisboa. 1981.
- [4] K. Jensen, N. Wirth. Pascal — user manual and report. Springer Verlag. Berlin. 1975.
- [5] J. A. Legatheaux Martins. NICE — um exercício de aplicação da teoria dos autómatos à programação. Relatório interno do Centro de Informática da Universidade Nova de Lisboa. CIUNL — 5/82. Lisboa, Março de 1982.

(Recebido em 20.7.1982)

