

워드프로세서 —처리언어 및 발전방향—

李 壽 洙*

■ 차 례 ■

- | | |
|-------------------------|-------------------------|
| 1. 워드프로세싱언어란 ? | 2,4 수학적 문서처리용, TEX, EQN |
| 2. 문제점과 발전의 방향 | 2,5 도표의 처리 |
| 2.1 요구의 다양성과 Macro기능 | 2,6 총괄처리와 회화적처리 |
| 2.2 상식과 데이터 베이스 | 3. 끝맺음 |
| 2.3 논리적 정보 및 물리적 정보의 분리 | 참 고 문 헌 |

1 워드프로세싱언어란 ?

작성코쳐하는 문서의 형태로 제어하기 위하여 문 서등을 표시하는 문자열간에 삽입되는 제어문자를 총 칭하여 “워드프로세싱언어”라 하며, 이를 이용함 으으로써 문서형태를 자유로히 변경할 수가 있을 뿐만 아니라 자동식자, 채판, 인쇄라고 하는 온라인 문서 작성과정을 자동화 시킬 수가 있다.

이러한 워드프로세싱기술은 사회적 요구와 관련된 기술분야의 발전에 힘입어 급속히 발달, 보급되고 있으며 특히 영문의 경우에는 인쇄, 제본, 판매 이외의 모든 과정을 저자측에서 수행하여 만든 서적의 예도 결코 적지가 않다.

그러나 종래의 방법에 의하여 원고가 인쇄물로 완성되기까지는 지금까지 일반저작가가 그다지 인식치 않아도 좋은 사람들이 워드프로세서를 이용하기 위하여서는 사실, 여러가지 정보를 부가하지 않으면 안된다. 예를 들면 활자의 크기를 몇 포인트로 할까? 활자의 종류는 어떤 것으로 할까, 표제는 몇 행으로 할까., 그림은 어떻게 배치시킬까. 등등의 지정을 행하지 않으면 안된다. 이러한 것들을 종래의 방식에서 편집자가 적극적으로 지정하든가, 또는 제 판 작업원

의 상식에 의거한 경험적인 판단기능에 의존하는지 하여 이루어져 왔다. 식자과정을 자동화 시키기 위하여서는 그러한 종류의 정보를 어떠한 형태이던지 온라인원고에 포함시켜야 한다.

즉 원고의 본문중에서 “여기서부터가 절의 시작” 또는 “이 부근에 도표가 삽입코쳐 한다.” 등과 명령(코멘드)을 원고본문과 명확히 구분할 수 있는 형태로, 본문중에 적당한 장소에 삽입시켜야 한다. 따라서 워드프로세싱이란 명령을 해석하여 페이지의 형식, 크기, 활자의 변경등 자유로히 구사하면서 저작가 또는 편집자가 희망하는 데로 문서를 작성하는 것을 의미하며 주로 워드프로세서라고 칭하는 프로그램의 집합에 의거 처리된다. 여기서 워드프로세싱언어라 함은 원고 본문중에 포함된 “명령”을 뜻한다.

워드프로세싱언어의 대표적인 유형의 한가지는 J. Saltzer 가 처음 개발한 CTSS 용 문서작성시스템 RUNOFF²⁾에 기원 한 것으로, 이러한 형태의 시스템은 RUNOFF 라는 명칭을 부분적으로 포함시키고 있다. 그 예로써는 벨 연구소의 Troff, 동경 대학의 Proff 등을 들 수가 있는데 RUNOFF 형태의 입출력문을 예를 들어 설명하여 보면 그림. 1의 (a), (b)과 같다.

즉 행(라인)의 선두에 마침표(.)가 있는 행이 명령 행으로, 그 외의 행은 본문의 내용을 표시하고 있다. “.pl 14”는 1페이지의 라인수(행수)를 14행

*正會員 : 光云大學 電算機工學科 副教授

- pl 14
- he " Daddy Long Legs Page #
- fo " Jean Webster
- rm 40
- nf
- in 20

LOCK WILLOW
SEPT 19 TH

- fi
- in 0
- ul

Dear Daddy,

- ti 5

Something has happened and I need your advice.
I need it from you, and from nobody else in the world.
Wouldn't it be possible for me to see you ?
It's so much easier to talk than to write; and I'm afraid your secretary might open the letter.

- ti 20
- ul

Judy

- sp 1
- ti 5

p.s. I'm very unhappy.

(a)

Daddy Long Legs Page 1

LOCK WILLOW
SEPT 19 TH

Sear Daddy,
Something has happened and I need yout advice, I need it from you, and

Jean Webster

Daddy Long Legs Page 2

fuom nobody else in the world. Wouldn't it be possible for me to see you ? It's so much easier to talk than to write; and I'm afrsid your secretary might open the letter.

Judy

Jean Webster

Daddy Long Legs Page 3

P.S. I'm very unhappy.

Jean Webster

(b)

그림 1. RUNOFF 형의 입력문 (a), 출력문(b)

으로 지정한 명령이며, ". he ………" 는 페이지의 표제 (page heading)의 관한 지정이며, ". fo ………" 는 페이지 하단을 지정하고 있다. ". he ………" 행의 #는 실제 페이지 번호로 치환된다. ". rm 40" 은 1행의 길이를 40 자로 지정하며, ". nf" 는 입력행과 출력행과의 대응관계를 1대 1로 지정하고 있으며, 또한 ". in 20" 은 Indentation (행의 좌단으로부터 그 행속에서 시작되는 문자 간으로 공백의 크기를 뜻함)을 20 문자분으로 하도록 지정하고 있다.

". fi" 는 ". fo" 의 끝을 표시하며, ". in 0" 는 Indentation 을 페이지의 좌단과 일치케하고 있고, ". ul" 은 다음 행에 하선 (underline)을 치도록 명령하는 것이다. ". ti 5" 는 다음 행의 선두를 임시

로 5 문자분의 공백을 두고 시작케하는 명령으로, 이는 절의 시작을 표시하기 위한 관습을 표기하기 위하여 사용되고 있다. ". ti ………" 의 전후의 행이 출력행에서는 다른 행에서 표현키 위해 명령이며, ". sp1" 은 그 곳을 1행을 뛰어라는 것을 의미한다. 출력(b)에서는 실선으로 페이지를 구분하였으며, 점선으로 페이지중에 들어가는 본문의 부분을 표시하고 있다.

이러한 RUNOFF 형식의 워드프로세싱언어(명령)는, 언어라고 하기는 약간 소박한 점이 있으나, 문서 처리상의 본질적인 특성을 많이 포함코 있다. 한편 종래, 주류의 방식과 다른 새로운 워드프로세싱 기술이 1980년 초, CMU(Carnegie Mellon Univ.)

의 SCRIBE³⁾가 발표된 이래 Knuth의 TEX⁴⁾, 일본 교토 대학의 EPPUNI¹⁾ 등이 줄을 있고 있다. 이 중에서 EPPUNI(에뿌이)는 필자가 1979년에서 1981년에 걸쳐 연구 개발한 것이다.

워드프로세서의 종래의 문제점과 발전방향을 설명하기 전에 시스템의 설계에 있어서 중요한 4가지 측면과, 이러한 측면을 실현시키기 위한 몇 가지의 고려 사항에 대하여 잠시 언급하면 다음과 같다. 먼저 4가지 측면으로써는 ①입력측면 ②출력측면 ③기계측면 ④인간측면이 있는데, ①은 워드프로세싱언어를 구사하여 어떻게 하면 보다 용이하게 문서를 작성할 수 있을까?, ②는 어떻게 하면 일반 인쇄물의 품질에 필적하는 고품질의 출력 문서를 작성할 수 있을까?, ③은 어떻게 하면 보다 적은 소프트웨어, 하드웨어적 공유 자산을 이용하여 효과적인 시스템을 작성할까하는 측면이며, 마지막으로 ④는 출력된 결과를 기초로 한 입력본문의 수정이라는 인간의 자연스러운 수정 과정을 어떻게 하면 효과적으로 수행할까하는 측면으로 볼 수가 있다.

이러한 측면을 실현시키기 위하여 현재까지 부분적 또는 전체적으로 도입된 기능적, 개념적인 방법을 나열하면 다음과 같다.

- ① 그래프, 표와 같은 비문자 정보의 자동작성 및 자동배치
- ② Location-free type의 명령 체계
- ③ Macro 기능
- ④ 데이터 베이스 기능의 도입
- ⑤ 블록 개념의 도입
- ⑥ 무지령형의 워드프로세싱언어

2 문제점과 발전의 방향

2.1 요구의 다양성과 Macro 기능

이 분야의 한가지 특성은 이용자의 요구가 극도로 다양하다는 것이다. 각주의 처리 방법만 생각해도 무서울 만큼의 다양성이 있다. 각주는 페이지의 하단에 위치하는 것이 보통이나, 장이나 절의 끝부분에 넣는 방법도 있다. 전자의 경우에 있어서도 각주의 크기가 해당 페이지에 들어 갈 수 없을때, 다음 페이지에 삽입시키는 방법³⁾과 해당 페이지의 본문을 다음 페이지로 넘기고 각주의 다른 각주 영역을 우선적으로 자동배치하는 방법¹⁾ 등이 있다. 또한 각주의 표시 방법도 다양하여 별표, 화살표, 번호 등으로 표시하고 있으며, 각주가 도표와 동일한 페이지에 삽입되어야 하는 경우, 그 처리는 한층 더 복

잡하게 된다.

그러나 EPPUNI¹⁾에서는 도표의 영역과 각주 영역의 처리에 있어서 처리의 우선 순위와 각주 영역 크기의 가변성을 이용하여 이를 효과적으로 처리하고 있다.

이와 같이 요구의 다양성을 충족시키기 위하여서는 전절에서 표시한 정도의 워드프로세싱언어으로써는 아주 부족하고, 사용키 불편한 점이 많아, 새로운 기능을 강구할 필요가 있다. 그 중에 전형적인 종래의 기능으로써는 Macro 기능을 들 수가 있다.

Macro의 기능은 Macro 정의부와 Macro call 부분으로 대별되며, Macro 정의부에서는 몇 개의 명령 및 문자열의 조합으로, 보다 용이하게 사용할 수 있는 처리 중심의 고수준 언어를 정의하는 능력이며 Macro call은 고수준 언어의 명령에 해당된다. 종래의 방식중, 예를 들면 벨 연구소의 UNIX용 Troff⁵⁾는 Macro의 개념의 대대적으로 채택한 대표적인 시스템으로, Troff의 이용자는 Macro 정의를 사용하여 문서를 작성하는 것으로 처리 원칙으로 하고 있다. 그러나 Macro 형식만으로써 문서 작성에 따른 문제점은 기성의 Macro 기능만으로써는 충족하기 어려운 경우에는 새로운 Macro의 정의가 필요하게 되며, 이러한 Macro 정의가 용이하지 않는 경우에는 문서의 작성이 아주 어렵게 되는 경향이 있다. 따라서 Macro 정의 기능은 워드프로세싱언어의 보조적 수단으로 사용되는 것이 적절하게 생각되며, 한번 정의된 Macro가 타의 문서 작성에 있어서 유용하게 사용될 수 있는 즉 Macro 정의문의 축척 기능이 바람직하다.

2.2 상식과 데이터 베이스

문서의 작성에 있어서, 그 사회에서 당연한 것으로서 가정되고 있는 파라메타가 있다. 예를 들면 미국에서 또는 일본에서의 용지의 폭이 규격화되는 경우도 있을 수가 있고, 학술 잡지의 참고 문헌의 인용 방법도 각각 틀릴 수가 있다.

이러한 문제들, 문서를 작성할 때의 인간 중심으로 생각하여 "전기공학회에 제출하는 논문 형식으로 작성하라", 또는 "편지 형식으로 작성하라"는 간단한 명령으로 처리하는 것이 바람직하다.

바꾸어 말하면 특정 문서의 작성에 따른 각종 파라메터를 데이터 베이스화시켜 놓고, 시스템이 자동적으로 파라메타의 선정과 이 파라메타에 의거한 문서 형태를 작성하는 것이 필요하며, 시스템의 설계는 아주 복잡하게 되나, 이용자는 아주 간단하게

문서를 작성할 수 있는 특성이 있다.

이러한 사고 방식을 대폭적으로 도입한 것으로서 Reid의 SCRIBE³⁾, Lee S. Y.의 EPPUNI¹⁾를 들 수 있다.

예를 들면 @ DEVICE (출력기기명)을 쓰며 출력기기가 Print 형태, 또는 plot 형태에 따라 각종 파라메터가 자동적으로 선정되며. 본문의 일부를 @I(KWANG WOON UNIV)와 같이 괄호 속에 표현하면, 괄호 속의 문자열은 전부 이탤릭체를 표시되고, 만약 Print 형태의 출력기기인 경우에는 이 명령 문자체가 무시된다. 또한 입력문의 최초에 @ Make (Thesis)가 있으면 학위논문에 알맞는 서식의 문서가 작성된다.

SCRIBE는 이용자의 평판이 좋은 시스템이며, 작성 가능한 기능에서는 아주 간단히 고품질의 출력이 보장되고 있으나, 단지 결점으로써는 SCRIBE의 기능 중에 포함되지 않는 것은 전혀 작성이 불가능하다는 비판도 있다. EPPUNI에서는 출력기기의 하드웨어적 성질 등과 같이 일반 이용자에게 잘 알려져 있지 않는 사항에 대하여서는 SCRIBE와 같이 처리하고 있으나 기타는 Macro 기능을 이용하여 문서제어할 수 있도록 하고 있다.

2.3 논리적 정보 및 물리적 정보의 분리

EPPUNI 및 SCRIBE는 물리적 정보와 논리적 제어 정보의 분리를 시도한 시스템의 좋은 예이다. 여기서 물리적, 논리적 정보(명령)이라함은 다음과 같다. 즉 "페이지의 길이는 14행, 1행간의 최대 문자수는 40문자"의 명령은 문서의 내용과는 관계없이 문서의 체제에만 하는 사항이다. 이러한 정보를 물리적 정보라고 하며 다른 예로써는 출력기기의 지정 및 문서 전체의 기준 활자의 지정 등등 수가 있다.

이것에 대하여 순수하게 문서의 내용에 관계하는 것을 논리적 정보라고 하며, 그 예로써는 Indentation의 지정명령이나, 조건부 또는 무조건 개행 명령, 부분적 활자 변경의 명령이 여기에 해당된다.

이러한 논리적·물리적 정보의 구분에 관한 개념은 포트란으로 작성된 프로그램에 있어서 선언문의 집합과 그 밑에 있는 실행문의 집합으로 볼때, 선언문의 집합이 물리적 정보에 해당된다고 볼 수가 있다.

이러한 물리적 정보를 SCRIBE와 EPPUNI에서는 테이블타베이스 또는 계산기 내부 표현으로 대체함으로써 이용자는 전혀 물리적인 사항에는 신경을

쓰지 않아도 좋게 설계되어 있다.

2.4 수학적문서처리용, TEX, EON

워드프로세싱언어의 설계자에 있어서 가장 흥미로운 대상중에는 수학적문서의 작성이란 문제가 있다. 실제, 수학적문서에는 특수기호, Subscript/superscript 등이 많고, 내용을 깊게 이해하고 있는 사람이외는 수학적표현이 조금만 이상하여도 이해하기 어려운 상황이 발생하기 쉬우므로, 수학적출판물의 제작은 아주 힘들고 비싸게 취급되고 있다. 이미 설명한 RUNOFF로써는 수식의 표현이 거의 불가능하다고 보아도 과언이아니하며 EPPUNI, SCRIBE에서도 간단한 수식의 작성에만 한다고 있는 실정이다.

이 방면에 대하여 유용한 시스템으로써 Kernighan 등의 EQN⁶⁾와 Knuth의 TEX⁴⁾를 들 수 있는데, 이하 TEX에 대하여 중점적으로 설명하고자 한다.

EPPUNI, SCRIBE는 처리중심으로 볼때 고수준 언어에 된다면 TEX는 수학적의 제작성, 모든 관습과 기능을 철저히 분석하여 작성코져하는 모든 수학적 표현을 제어할 수있는 기능이 세분된 저수준언어로 비유할 수 있다.

TEX (정확하게는 그리스문자인 τ, ϵ, χ 의 대문자임)의 개발에서는 미국 수학회인 AMS(American Mathematical Society)의 후원을 받았으며, Knuth는 TEX와 관련하여 각종서체, 기호를 작성하기 위한 시스템인 METAFONT⁷⁾도 아울러 개발하였다.

TEX도 Macro 정의를 대대적으로 이용한 시스템으로 Macro 정의를 포함하여 수식제어에 있어서 복잡미묘한 현상이 발생할 가능성도 있으나, 이용방법의 철저한 지도로 말미암아, 비교적 깊은 지식이없이, 숙달을 통하여 유용하게 사용할 수 있다고 볼 수 있다.

타의 워드프로세싱기법과 비교할 때, TEX의 특기할 특성이라함은 상자(box), 풀(glue)의 개념을 도입하여 수학적문서의 작성에 때때로 생기는 복잡한 처리를 간결하게 표현하고자 하고 있다. 예를 들면 문자는 하나의 상자에 해당되며 상자를 풀로 붙여 연결시킨 것도 새로운 상자가 된다. 물론 풀은 필요에 따라서 어느 정도 신축가능하다.

이러한 개념을 이용하면 행의 좌측정렬(Left justify), 우측정렬(Right justify), 중앙배치(Centering)가 용이하게 된다. 또한 문자폭이 활자의 종류, 문자의 종류마다 각기 상이한 가변폭활자

(Font) 인 경우, 종래에는 일반적으로 행의 우측 정렬을 행할 경우, 행단위로 단어간의 간격을 신축시켜 왔으나, TEX의 경우에서 복수개의 행속에서 단어의 이동 및 문자간의 간격조정이 가능한 것이 특징이다. 즉 상자와 풀의 개념은 페이지상의 2차원 배치에 관계한 상황을 명확하게 표현하는데 강력한 개념이다.

2.5 도표의 처리

워드프로세싱언어가 가질 수 있는 능력중에서, 특히 중요한 것이 도표의 처리라고 할 수 있다.

도표의 처리라 함은 도표의 작성문제와 도표의 배치 문제로 구분된다. 그러나 워드프로세서에서는 주로 도표의 배치 문제를 직접적으로 관련시키고 있으나 도표의 작성방법에 대하여서는 타 분야로 나누어서 생각하는 경향이 많다.

도표의 배치 문제에 있어서 일반적으로 도표가 들어갈 부분을 공백으로 남겨두는 시스템이 많으며 실제 도표를 문서속에 삽입시키는 방식중에는 Man-machine interactive 방식^{8), 9)}과 자동배치 방식¹¹⁾으로 나누어 볼 수가 있다.

상기의 각각에 있어서 도형배치에 관한 융통성 및 배치위치의 적합성에서는 전자가 좋으나, 문서의 효율적인 작성이라는 점에서는 후자가 월등하다. 이후 도형의 자동배치 및 도형의 자동작성은 워드프로세싱 분야가 확대되어져 나갈 가장 중요한 분야중에 하나로 생각되고 있다.

2.6 총괄처리와 회화적처리

일반적으로, 워드프로세싱언어는 사전에 삽입되어 있는 명령에 의거하여 일거에 문서작성을 수행하는 총괄적(batch) 처리방식을 택하고 있는 경향이 많으나, 꼭 이러한 방식이 좋다고는 할 수 없다. 즉 문서작성의 과정상에서 볼때 인간은 출력문서의 결과를 보면서 출력문서가 만족스럽지 않을때는 즉시 수정하면서 점차적으로 출력문서를 작성해가는 방법이 자연스럽게 여겨지게 된다. 종래에는 출력문서의 고속작용에 따라 출력장치(display 등)의 빈곤과 컴퓨터의 기억용량의 부족등으로 총괄처리 형식이 많았으나 하드웨어적, 소프트웨어적 주변 환경의 발전에 힘입어 display 등을 이용하여 완성품의 모양을 보면서 출력문서를 작성해가는 회화적인 처리 방식이 많아지는 경향도 있다. 회화적 처리방식은 도표나 수식이 많이 포함되거나 출력문서의 서식의 변화가 다양한 환경에서 특히 유효하게 이용될 수가 있다.

사용자 중심으로 볼때 총괄처리방식은 워드프로세싱처리에 관한 전문적인 지식이 많이 필요치 않아 일반적인 사용자용이 많으나, 회화적인 방식은 전문적인 편집자용이 많을 것으로 생각된다.

③ 끝맺음

지금까지의 내용은 워드프로세서의 전반에 관한 것이라기 보다는 차라리 몇몇 전형적 시스템의 개발에 따른 아이디어를 중심으로 설명한 것에 불과하며 각 종서식의 문서의 작성을 위한 워드프로세싱언어의 존재를 사전에 가정하고 있었다. 역설하면 입력본문의 구조로부터 워드프로세싱처리를 위한 방식 즉 명령형의 워드프로세싱언어를 생각해보면 어떨까?

예를들면 입력본문중의 1행의 공백행은 출력문서에서 절의 구분으로써 사용될 수도 있을 것이다.

이곳에서는 주로 영문처리를 워드프로세싱어를 중심으로 설명하였으나 한자를 포함하고 있는 한글워드프로세싱과 비교하여 볼때 워드프로세싱의 본질은 전혀 같다고 볼 수가 있다. 단지 틀리다면 입력기능인 에디팅기능의 차이가 있을것이다.

끝으로 필자가 개발한 EPPUNI 워드프로세서는 현재 일본의 교토대학, 쓰쿠바대학에서 일반사용자에게 서비스를 제공하고 있음을 밝히고, 교토대학 정보공학과의 사카이도시유끼(坂井利之) 교수께 감사의 뜻을 표한다.

참 고 문 헌

- 1) S. Y. Lee and T. SaKai : " An English text processing system with automatic layout - decision of figures and tables ", Trans of IECE, Vol. J 65-D, No. 5, pp 558~565, May 1982 (in Japanese), and S. Y. LEE : " An automatic generation system of bilevel document image by dynamic composition method ", Thesis of Doctor at Kyoto Univ. pp 59~96, Jan. 1983 (in English)
- 2) Saltzer, J. : " Manuscript typing and editing - TYPESET, RUNOFF, P.13, Section AH. 9. 01 in MIT computing center : the compatible time-sharing system", MIT press, Cambridge, Mass. 1965.

- 3) Reid B. K. and Wadlker, J.H. : " SC RIBE introductory User manual , and SCRIBE format designers guide , " pp. 332, Carnegie.-Mellon Univ. , Jul . 1979 .
- 4) Knuth, D. E : " TAU EPSILON CHI - A system for technical text , " pp. 201, In D. E. Knuth. , " TEX and METAFONT" , Digital Press and American Mathematical Society, Bedford, Mass. and Providence, R. I. 1979.
- 5) Ossanna, Jr. : " NROFF/TROFF, p. 33, Computing Science Technical Report No. 54, Bell Lab. , Oct.1976.
- 6) Kernighan, B. W. and Cherry, L.L. : " A system for typesetting mathematics , Comm. ACM, Vol .18 , No. 3, pp 151 - 157, Mar. 1975.
- 7) Knuth, D. E : " METAFONT - A system for Alphabet Design, pp. 105, referred to (4).
- 8) 安部外 2 人 : " 英文図表清書システム FROFF について情報処理, Vol . 22, No .3, pp. 198~ 205, May, 1981.
- 9) 寺西ら 1 人 : " 図表を含む英文原稿用エディタ, 信学論, IE 79 - 71, pp 29~36, Nov. 1979.