

메인프레임 下에서의 워드 프로세싱 시스템의 設計에 관한 研究

黃 大 勳
(電 算 室)

..... < 차 례 >

- I. 序 論
- II. WORDPRO를 위한 制御시스템 與件
 - 1. 國·漢文 文字 入出力을 위한 work-station
- III. WORDPRO의 構成
 - 1. 파일 管理
 - 2. Editor
 - 3. Formatter
- IV. 結 論
- 參考文獻

I . 序 論

컴퓨터 시스템의 發展 및 그 利用의 增大로 因한 事務自動化的 추세는 文書 및 書式과 같은 document의 處理를 위한 Word Processing System의 發展을 맞이하게 되었다. 特히 小型計算機를 利用한 Word Processor의 發展은 近來에 이르러 急速히 이루어져 많은 小型컴퓨터 製造業體들이 다투어 各機種 特有的 하드웨어體系 및 소프트웨어體系를 가지는 Word Processor를 開發하여 市販하기에 이르렀다.

그러나 이러한 시스템을 開發하는데 있어서 우리의 주어진 環境을 감안할 때 Word Processor 는 한글處理가 必然的으로 可能하여야 하며 때에 따라서는 漢字 및 英文處理까지도 同時에 可能한 시스템의 開發이 要求되고 있는 實情이다. 또 Word Processor 의 對象이 되는 document 그 自體도 單純히 어떤 書式이나 報告書等 個人的 文書處理 中心의 시스템 段階에서 우리 言語로 表記된 많은 量의 資料를 多數의 利用者가 同時에 處理하게 할 수 있는 汎用시스템(Multi-tasking on-line control system)에 대한 必要가 많은 것이 現實이다.

이러한 現實的인 需要에 反하여 現在 開發되어 있거나 或은 開發中인 Word Processor 는 주로 小型計算機를 통한 個人業務處理 中心으로 設計되었기 때문에 멀티 타스킹(multi-tasking)이나 大量的 補助記憶裝置等을 利用한 大單位 document 나 漢字處理 或은 多數利用者에 依한 document 의 同時處理等은 不可能한 實情이다.

本 研究는 이와같은 word processor 의 現實性을 감안하여 一般的인 大型의 汎用컴퓨터의 處理機能을 利用하여 한글·漢字·英文으로 構成된 많은 量의 document 를 處理할 수 있도록 하는 word processing system(以下 WORDPRO)으로 本 研究院이 保有하고 있는 汎用컴퓨터(IBM 4341/L02)에서 運用될 수 있도록 設計·開發한 시스템에 對하여 說明하고자 한다.

Ⅱ. WORDPRO 를 위한 制御시스템 興件

本 研究에서 다루고 있는 WORDPRO시스템의 基本設計 概念이 一般的인 汎用컴퓨터가 갖추고 있는 온라인制御機能과 通信機能(Online task and Communication control)을 그대로 利用하되 우리말로 構成된 文章의 入出力과 편집 및 資料管理 機能을 遂行할 수 있는 하나의 應用프로그램(Application Program)으로서 構想·設計되고 있기 때문에 이 시스템이 遂行(operating)되기 위한 Host system의 基礎 制御機能은 하나의 前提條件으로 理解되어야 할 必要가 있다.

一般的인 汎用컴퓨터의 基本 制御機能에는, 해당 시스템의 資源을 가장 效率的으로 使用하기 위해 채택된 多重프로그래밍(multiprogramming) 機能과 주어

진 入出力裝置와 應用프로그램間的 資料傳達를 위한 入出力制御機能이 가장 基本되는 機能으로 區分될 수 있으나 最近의 거의 모든 시스템에 있어서는 온라인 리얼 타임(online real time)處理機能을 위한 制御機能이 거의 必須적으로 常備되어 있다. 온라인 리얼 타임 制御機能은 그 機能에 對한 應用業務의 性格上 必須적으로 多數의 利用者가 同一한 處理機能을 서로 할애하며 使用할 수 있도록 하는 multi-thread task 制御機能이 있어야 되며 아울러 多數의 터미날 或은 work-station에 對한 入出力機能을 Queing하여야 할 制御機能 또한 必須적인 것이다.

本 研究에서 다루고 있는 WORDPRO는 이와 같은 Host 시스템의 制御機能으로 IBM의 DOS/VSE operating system이 利用되었으며 온라인 리얼 타임 制御를 위한 시스템 프로그램(system program)으로는 VTAM/CICS가 利用되었다.

1. 國·漢文 文字 入出力을 위한 work-station

現在 國內에 보급·설치된 大部分의 汎用컴퓨터는 一般的으로 英文과 數字 그리고 一部 特殊記號에 對해서만 그 內部的 記憶方法(internal storage mode)을 갖도록 設計되어 있어 한글이나 漢字에 對해서는 別途의 內部的 記憶形態를 設計(IBM機種의 경우 2 바이트 코드體系)하여 任意로 記憶시키고 있다.

그러나 이와 같이 2 바이트(byte)로 表示된 國·漢文 文字의 形態는 個別 記憶形態에 相應하는 文字의 形象(memorized character font)을 處理하여 알맞은 出力裝置에 表示되어야 하는 바, 本 WORDPRO의 開發·試險은 NEC 6300 work-station(Intelligent terminal)을 利用하였는데 이는 SDLC 通信方法을 채택하여 IBM 3270에 대응하는 NE3270KH emulator을 使用하고 있다. 또 이 work-station은 現在 1,598 字의 한글과 약 5,076 字의 漢字, 그리고 日本글자를 포함한 一部 特殊文字에 對한 文字形象과 2 바이트 內部記憶形式을 가지고 있다.

Ⅲ. WORDPRO의 構成

Word processing system의 一般的인 機能으로는 原始텍스트(text)의 入力과 入力蓄積된 텍스트의 편집을 위한 editor 機能, 그리고 편집된 텍스트를 利用者가 目的하는 方式으로 바꾸어 주는 formatter 機能으로 區分·設計될 수 있는데 本 研究에서 다루는 시스템은 많은 量의 資料와 여러 利用者의 텍스트를 處理·管理하여야 하는 點을 감안하여 위의 두 機能 以外에 別途로 파일管理機能(file management function)을 따로 두어 크게 3가지로 區分하여 說明하고자 한다.

1. 파일 管理

Editing 및 formatting의 對象이 되는 텍스트를 效率적으로 管理하기 위하여 本 研究는 library-file, text-file, 그리고 work-file로 區分하여 3개의 資料파일(data file)을 편성토록 하였으며, 이는 모두 KSDS(Key Sequenced Data Set)를 利用한 VSAM(Virtual Storage Access Method)파일로 構成하였다.

利用者가 任意의 텍스트를 最初로 入力하고자 하면 library-file에 텍스트이름을 登錄한 後, 텍스트의 내용이 들어갈 text-file上的 位置를 指定함으로써 利用者가 入力하는 내용이 레코드(record)單位로 text-file上에 蓄積된다.

任意의 기존 入力된 텍스트를 利用者가 편집·교정하고자 하면, 우선 library-file로부터 찾고자 하는 텍스트의 固有確認 項目을 찾아내고 이 固有項目에 있는 內容中 해당 텍스트의 처음 레코드의 text-file上的 始作番地를 가지고 편집·교정의 對象이 되는 텍스트를 text-file에서 불러내도록 한다. 읽혀진 텍스트는 편집·교정을 遂行하기 위해 다시 work-file에 옮겨지며 텍스트에 對한 實質적인 편집·교정機能은 이 곳에서 遂行된다. 以後에 work-file에서의 편집·교정機能이 完了된 텍스트는 다시 text-file에 옮겨지게 되며 同時에 새로히 變更된 처음의 레코드連結因子(record pointer)등을 update한 後 library-file上的 變動事項을 update함으로써 해당 텍스트의 편집·교정을 마치도록 하였다.

Formatter 는 text-file 로부터 1 레코드씩을 계속 읽어, 出力버퍼(output-buffer)에 利用者가 設計한 書式대로 出力形態를 形成하여 내보내는데 出力버퍼가 完成되면 이 버퍼를 出力파일(output file)이나 출력 리스트(list)로 出力함과 同時に 새로운 出力버퍼를 形成하는 과정을 反復한다.

여기에서 library-file 은 각 텍스트의 텍스트固有項目(text-identifier), 각 텍스트에 대한 text-file 內의 처음 레코드 始作番地(record start address), 레코드갯수, 暗號 및 作成者 이름 등 각 텍스트에 관련되는 情報를 가지며 그 레코드 構成은 <表 1>과 같다.

Text-file 은 여러개의 原始텍스트가 前方連結因子(forward pointer) 및 後方連結因子(backward pointer)를 가지고 무작위(random)하게 保管되어 있는 파일로서 각 텍스트의 始作레코드 및 終了레코드는 前方連結因子 및 後方連結因子가 各各 null 로 set 된다. 이때 前方連結因子는 다음 레코드를 指定하며 後方連結因子는 앞의 레코드를 指定하는데, 이와 같이 兩方連結리스트(doubly linked list)構造를 가지는 text-file 의 레코드 構成은 <表 2>와 같다.

아래에서와 같이 1 레코드에는 實質적으로 72 바이트(한글 · 한자 36 字)의 텍스트 內容이 있게 되는데 이 72 바이트의 텍스트가 편집 · 교정의 單位가 되는 text line size 가 된다.

Work-file 은 實質적인 편집 · 교정機能이 遂行되는 補助記憶裝置(auxiliary storage device)上的 領域으로서 만약 text-file 自體를 work-file 로 한다면 여러번에 걸쳐서 텍스트를 편집 · 교정하였을 경우, 그때마다 text-file 의 레코드 連結因子가 바뀌어져 複雜하여 지고 따라서 레코드呼出性能(record access performance)이 떨어진다는 點에서 work-file 을 따로 두었다. 이렇게 함으로

<表 1> Library-file 의 레코드 構成

text-id	start address	record number	pass-word	depart-ment	author	create-date	subject	comment
char(7)	pic'(7)9'	pic'(6)9'	char(4)	char(14)	char(8)	char(6)	char(60)	char(300)

<表 2> Text-file 의 레코드 構成

record-id	text	forward pointer	backward pointer
pic'(7)9'	char(72)	pic'(7)9'	pic'(7)9'

써 편집·교정時 복잡하게 連結된 work-file 上의 레코드連結因子들이 편집·교정이 끝나고 text-file 로 옮겨질 때 레코드 連結因子들이 順次的으로 整理되도록 하였고, 또 편집·교정時 우발적으로 發生할 수 있는 事故로부터 텍스트를 보호할 수 있게 하였다. 따라서 work-file 의 레코드 構成은 text-file 의 것과 같다.

2. Editor

(1) Text-buffer 를 통한 편집·교정의 論理構造

利用者가 편집·교정의 對象이 되는 텍스트를 指定함으로써 work-file 로 옮겨진 텍스트는 편집·교정處理를 위하여 다시 프로그램內의 work area(working memory)로 옮겨지는데 이 work area 를 텍스트 버퍼(text buffer)라고 表現하겠다. 텍스트 버퍼는 단말기의 모니터(moniter)에 표시(display)되는 스크린 페이지(screen page)와 같은 概念으로 segment 로 나누어 處理한다. 따라서 텍스트 버퍼나 스크린 페이지는 편집·교정의 基本單位가 되는 같은 數의 텍스트 라인(text-line)을 가지며, 하나의 텍스트 라인은 work-file 이나 text-file 의 경우와 같이 72 바이트로 形成되기 때문에 2 바이트의 한글·漢字의 경우 36字 까지 하나의 텍스트 라인으로 처리될 수 있다.

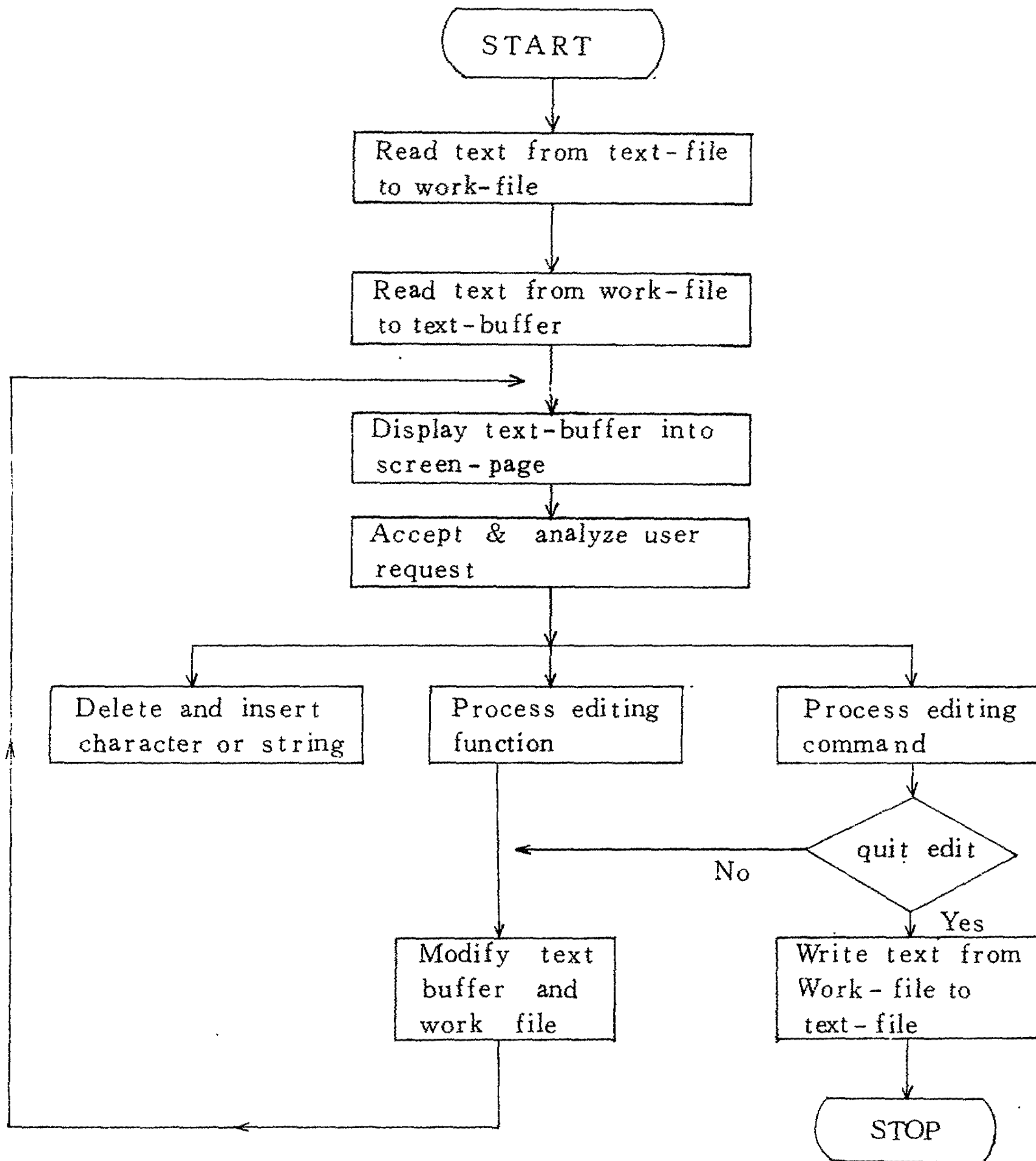
텍스트 버퍼에 들어온 텍스트를 對象으로 편집·교정處理를 하나 하나 遂行하게 되면 그때마다 텍스트 버퍼의 內容이 바뀌게 되고 同時に work-file 에도 update 가 이루어 지도록 하면서 그 結果를 모니터에도 表示하는 處理過程을 反復시킴으로써 시스템과 利用者가 인터액티브(interactive)한 편집·교정機能이 수행되도록 하였다. 이와 같이 텍스트를 work-file 에 두고 스크린 페이지 만큼의 텍스트를 텍스트 라인 單位로 텍스트 버퍼에 옮겨가면서 處理하는 論理的 構成은 大單位의 텍스트를 주기억장치(Main memory)에 전부 불러놓고 편집·교정하고자 할때 全體텍스트크기 만큼의 記憶場所가 할당되어야 한다는 問題點을 없애기 위한 것으로, 텍스트 버퍼에 들어온 텍스트 라인 單位의 레코드 運營은 <表 3>과 같은 形態의 前位 및 後位를 連結하는 레코드呼出連結因子(record access pointer)를 포함하는 segment 로 work-area 內에 편성·利用함으

로써 온라인 프로그램의 task를 最小化해야 하는 必要條件을 갖추도록 하였다. 이와 같이 텍스트 버퍼의 텍스트 라인 構造는 각 텍스트 라인을 하나의 노드(node)로 하는 兩方連結리스트로 構成하였다.

〈表 3〉 텍스트 라인의 構造

record-id	text	forward pointer	backward pointer	editing function
pic'(6)9'	char(72)	pic'(6)9'	pic'(6)9'	pic'(6)9'

〈圖 1〉 편집·교정을 위한 論理的 흐름圖

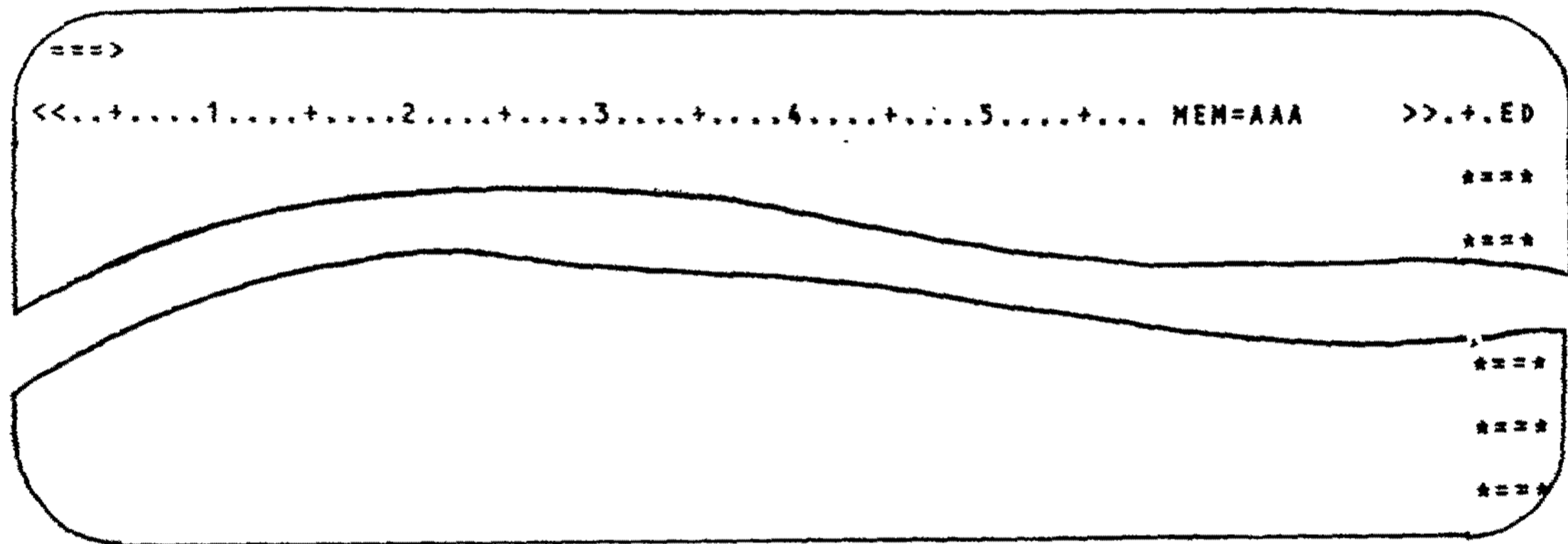


以上과 같은 條件下에서 WORDPRO의 editor 는 <圖 1>와 같은 편집·교정을 위한 論理的 흐름을 갖는다.

(2) Editor 의 機能 및 命令語

本 研究에서 利用되는 모니터의 全體的인 画面은 <圖 2>와 같으며 여기에서 圖解된 바와 같이 editing-command를 받아들이기 위한 monitor 상단의 command field와 모니터에 표시되는 각 텍스트 라인의 오른쪽 75~79column까지의 位置에 存在하는 function field 및 텍스트 버퍼內的 각 텍스트 라인을 利用者에게 보이기 위한 text-field로 나뉘어진다. Command field는 편집·교정시스템에 對한 全般的인 制御 및 text-field에 表示될 텍스트 라인의 位置를 制御하기 위한 領域이며, function field는 편집·교정의 기본 單位가 되는 텍스트 라인 單位의 處理를 위한 領域이다.

<圖 2> Editing을 위한 모니터 構成



<表 4>는 editing-command와 editing function의 syntax 및 그 機能이 要約되어 있는데, 例를 들어 command-field에 'C/강/남/20'과 같은 文字 또는 單語 變更(change) 명령을 주면 text-field의 最上段 텍스트 라인부터 다음 20個의 텍스트 라인에 存在하는 '강'이라는 文字가 '남'이라는 文字로 變更된다.

Editor는 以上の 機能 以外에 텍스트의 등록사항을 알리기 위한 Directory 機能, 각 text에 대한 情報를 알리며 또 內容을 바꾸기 爲한 Status 기능과 利用者에게 시스템의 利用方法을 알리기 위한 Help 기능 등이 있다.

<表 4>

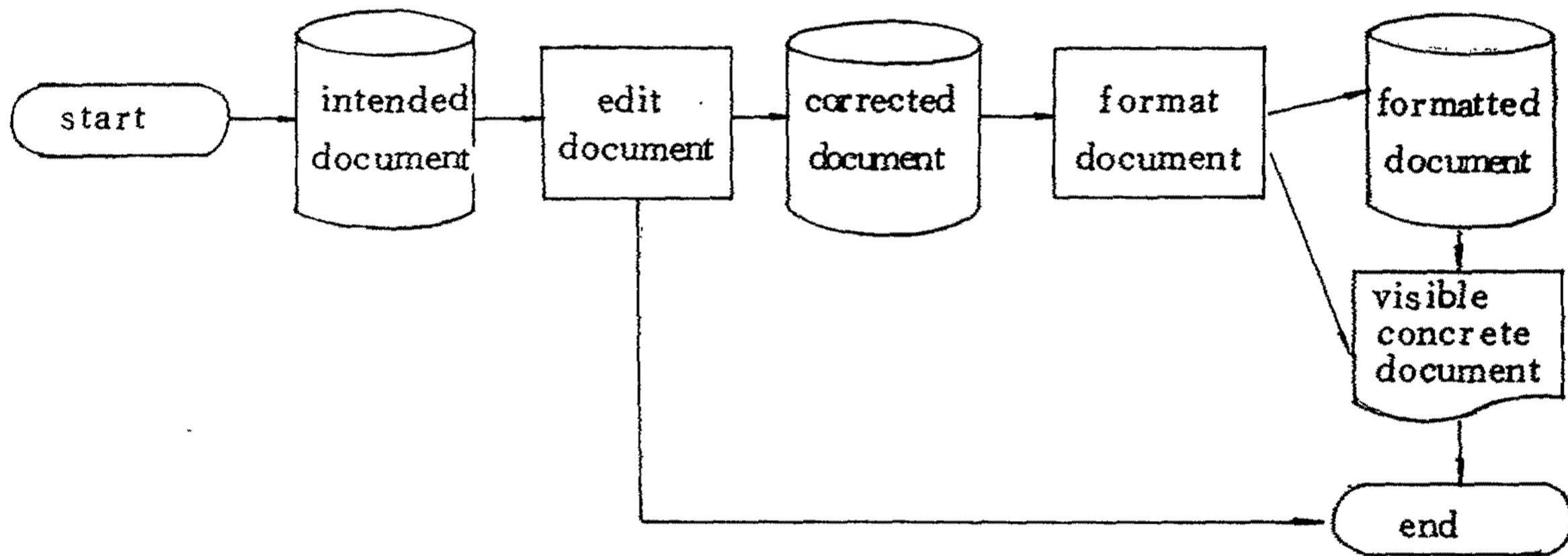
Editor의 機能 要約

I. Editing Command		
기능구분	Syntax	기능 요약
文字·單語變更	C/a/b/n	畫面에 表示된 最上段 텍스트라인 부터 n줄에 포함된 모든 a文字(혹은 단어)를 b文字(혹은 단어)로 변환代置(Change)
文字·單語찾기	Fa	畫面에 表示된 最上段 텍스트라인 부터 最初로 나타나는 a文字(혹은 단어)를 포함하는 텍스트라인을 畫面 最上段에 表示(Find)
Screen page 제 어	FOn	現在 表示된 스크리페이지의 n page 앞의 텍스트가 모니터에 표시(Forth)
	BAn	現在 表示된 스크린페이지의 n페이지 뒤의 텍스트가 모니터에 표시(Back)
	T	텍스트의 始作部分부터 모니터에 表示(Top)
	B	텍스트의 끝部分이 모니터에 表示(Bottom)
편집·교정 완료	Q	現在 進行中인 텍스트의 편집·교정을 완료
II. Editing function		
	Cn	해당 텍스트라인을 임시 場所(temporary area)에 임시 보관(Copy)
	Mn	n個의 해당 텍스트라인을 現위치에서 삭제하고 임시 場所에 보관(Move)
	Dn	n個의 텍스트라인을 삭제>Delete)
	I	Copy 또는 Move 명령으로 임시 場所에 임시보관된 텍스트를 지정된 위치에 삽입(Insert)
	An	n個의 blank 텍스트라인을 새로히 생성(Append)
	Sn	現位置의 텍스트라인과 내용이 같은 n個의 텍스트라인을 새로히 生成(Same)
	L	이 function이 표기된 텍스트라인을 畫面의 最上段에 表示(Locate)
III. 文字·單語의 삽입·삭제		
삽 입	PF12	文字·혹은 單語를 삽입하고자 하는 위치에 커서(cursor)를 옮겨놓은 후 PF12 function-Key와 함께 文字·單語를 入力
삭 제	PF13	삭제하고자 하는 文字 혹은 單語를 space key를 over-type한 후 삭제가 끝나는 位置에서 PF13 function key를 key in

3. Formatter

Formatter는 editor에 의하여 편집·교정이 완료된 텍스트를 사용자가 요구하는書式으로組版·印刷하는機能을 가지며 editor와는 <圖 3>과 같은作業進行이 이루어진다.

<圖 3> WORDPRO의 job 처리 順序



이때 editor에 의하여處理된 corrected document는 순수한 텍스트와 텍스트를 formatting하기 위한 포매팅제어명령(formatting control command)으로構成되며, formatter에 의해 처리된 formatted document는 사용자가 formatting과 동시에 여러개의 visible concrete document를 필요로 하거나 또는 此後에 다시 formatting을 하지 않고 formatted hard copy를 얻고자 할때 利用될 수 있다.

(1) formatter의 論理構造 및 機能

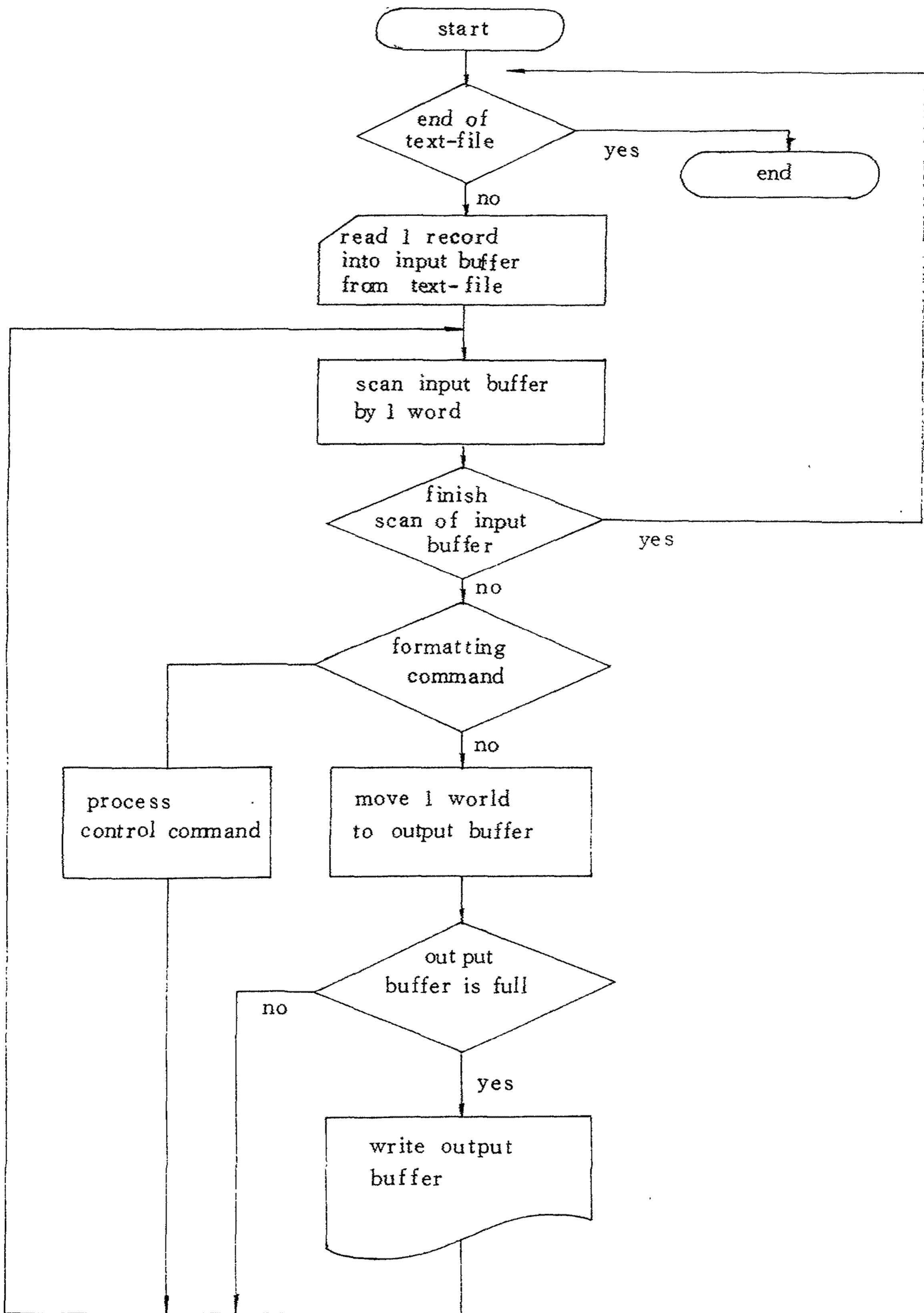
Formatting의 對象이 되는 corrected document는 formatting을 制御하기 위한 포매팅제어명령이 텍스트의 中間에 삽입되어 있는 혼합된 形態로 構成되며, 이는 <圖 4>와 같은 論理構造下에서 실질적인 formatting이 遂行된다.

以上の 論理的 構造下에서 다음과 같은 formatting 機能이 遂行된다.

- 各 line에 對한 機能: line center, line break, line indent
- 여러 line에 걸친 機能: left margin, right margin, line length, line justification

<圖 4>

formatting을 위한 論理的 흐름도



- line space 에 對한 機能 : single space, double space, leave blank space
- page control 을 위한 機能 : top margin, page header, page length, begin new page, page number

IV. 結 論

本 研究에서 大型시스템의 온라인制御機能과 通信機能에 work station의 入出力 機能을 利用하여 2 바이트 한글·漢字코드體系를 갖는 WORDPRO의 설계에 있어서 다음과 같은 結論을 얻을 수 있었다.

첫째, 多數의 利用者가 同一한 시스템下에서 同時에 機能수행이 可能하다.

둘째, 實質적으로 텍스트의 크기 및 個數가 시스템에 制約을 받지 않으며 거의 無限하게 可能하다.

셋째, 主記憶裝置에서 할당되는 텍스트 버퍼의 크기는 monitor 에 표시되는 텍스트의 크기와 同一하므로 편집·교정을 위한 主記憶裝置의 할당을 極小化할 수 있다.

〈參考文獻〉

1. Norman Meyrowitz & Andries Van Dam, "Interactive Editing Systems," *Computing Surveys*, Vol.14, No.3, Sept.1982.
2. Richard Furuta 外, "Document Formatting Systems : Surveys, Concepts, and Issues", *Computing Surveys*, Vol.14, No.3, Sept.1982.
3. Michael Levison, "A Programmable Text-editing System", *Software Practice and Experience*, Vol.12, 1982.
4. 한국 I.B.M. 주식회사, 「IBM한글사용 지침서」, K:GCHG-0020-0, Mar.1982.
5. 삼성전관공업주식회사, *SED Terminal-NE3270 KH Emulator*, Mar.1983.
6. SC33-0077-3, CICS/VS version release5-Application Programmer's Reference Manual, Jul.1981.
7. SC24-5144-1, *Using VSE/VSAM-Commands and Macros*, Dec.1979.
8. GC38-0282-1, *Advanced Communications Function for VTAM*, Aug.1977.