HEVE 사용자 지침서

주문 번호 : BA322-90028

2005년5월

본 지침서는 한글 확장 다변 편집기(HEVE) 를 사용하는데 도움이 되는 개별 정보에 대한 설명서 입니다.

개정/갱신	정보 :	본 개정 참조서는 VMS/Hangul AXP V1.5 용 HEVE 사용자 지침서를 대신한 것입니다.
소프트웨어	버전 :	OpenVMS/Hangul I64 Version 8.2 OpenVMS/Hangul Alpha Version 7.3−2

Hewlett−Packard Company
Palo Alto, California

© Copyright 2005 Hewlett−Packard Development Company, L.P.

기밀 컴퓨터 소프트웨어입니다. 소유, 사용 또는 복사를 위해서는 HP로부터 유효한 라이센스를 취득해야 합니다. FAR 12.211 및 12.212에 준거하여 상용컴퓨터 소프트웨어, 컴퓨터 소프트웨어 문서 및 사용 항목의 기술 데이터에 대한 라이센스가 공급업체의 표준 사용라이센스에 따라 미합중국 정부에 부여됩니다.

이 설명서의 내용은 예고 없이 변경될 수 있습니다. HP 제품과 서비스에 대한 보증은 오직 제품 및 서비스와 함께 제공되는 명시적 보증서만을 근거로 합니다. 이 설명서의 어떤 내용도 추가 보증 제정으로 해석할 수 없습니다. HP는 이 문서에 포함된 기술적 오류나 편집상의 오류에 대해 책임을 지지 않습니다.

Intel 및 Itanium 는미국과다른국가에서 Intel Corporation 또는 자회사의 상표 또는 등록 상표입니다.

Printed in Singapore

목 차

서 문 xii

제 **1** 장 **HEVE** 입문 1-1
1.1 서론 1-1
1.1.1 2개국어 도움말 1-2
1.1.2 2개국어 메시지 1-3
1.2 편집 세션의 시작과 종료 1-3
1.2.1 HEVE의 호출 1-3
1.2.2 HEVE 종료 1-9
1.2.2.1 편집 내용의 저장 1-10
1.2.2.2 편집 내용의 취소 1-10
1.3 중단된 편집 세션으로부터의 복구 1-11
1.4 기본적인 편집 작업 1-11
1.4.1 한 위치에서 다른 위치로 원문 이동 1-13
1.4.2 원문 탐색 1-13
1.5 HEVE 명령의 입력 1-13
1.6 HEVE의 도움말 보기 1-15

제 **2** 장 원문의 편집 2-1
2.1 커서 이동 2-1
2.1.1 커서 이동의 자유/고정 설정 2-3
2.1.2 커서 화면표시 모드의 설정 2-4
2.1.3 보이지 않는 표시기의 사용 2-6
2.2 편집 모드의 사용 2-8
2.3 원문 지우기 및 복구 2-10
2.4 원문 선택 2-13
2.4.1 표준 선택의 시작 2-14

2.4.2 표준 선택 후의 편집 작업 2-14
2.4.2.1 선택된 범위의 원문 삭제 또는 복사 ... 2-16
2.4.2.2 삭제 대기 2-16
2.5 상자 편집 2-17
2.5.1 상자 선택의 시작 2-17
2.5.2 상자 선택의 사용 2-18
2.5.2.1 BOX CUT 및 BOX COPY 2-20
2.5.2.2 BOX PASTE 2-22
2.5.2.3 SET BOX [NO]PAD 2-24
2.5.2.4 SET BOX [NO]SELECT 2-26

제 **3** 장 원문 형식화 3-1
3.1 여백 설정 3-1
3.2 단어 줄 바꿔쓰기 설정 3-2
3.3 탭 설정 3-2
3.4 단락 들여쓰기 3-4
3.5 원문 채움 3-6
3.6 원문 들여쓰기 3-9
3.7 원문 조정 3-11
3.8 문자형 및 문자 크기의 변경 3-12
3.9 페이지 매김 3-14

제 **4** 장 원문 탐색 4-1
4.1 원문 찾기 4-2
4.2 원문 대체 4-4
4.3 문자형 및 문자크기 구분 기능의 설정 4-7
4.4 화이트 스페이스로 탭 설정 4-8
4.5 와일드카드를 사용한 탐색 4-8

제 **5** 장 선/상자 그리기 및 기호 5-1

5.1 선/상자 그리기 5-1
5.2 기호 5-6

제 **6** 장 버퍼와 창을 사용한 **2**개 이상의 파일 편집 6-1
6.1 버퍼의 사용 6-2
6.1.1 버퍼의 목록 보기 6-2
6.1.2 버퍼를 현재 창에 집어 넣기 6-2
6.1.3 파일을 버퍼로 읽어 들이기 6-3
6.1.4 버퍼 내용을 파일에 쓰기 6-4
6.1.5 버퍼 삭제 6-5
6.2 창의 사용 6-6
6.2.1 다중 파일 보기 6-6
6.2.2 창 삭제 6-9
6.3 다중 창을 사용한 다중 버퍼 편집 6-10

제 **7** 장 **HEVE**에서의 키 정의 7-1
7.1 HEVE의 명령 실행을 위한 키 정의 7-5
7.2 Learn 순서를 입력하기 위한 키 정의 7-6
7.3 GOLD 키의 정의 7-8
7.4 Motif-스타일 기능 키의 정의 7-9
7.5 마우스 단추의 정의 7-11
7.6 키패드의 설정 7-12
7.7 키 정의 해제 7-13
7.8 정의할 수 없는 키들 7-14

제 **8** 장 **HEVE**의 설정 8-1
8.1 HEVE 내정 설정값 8-3
8.2 속성의 저장 8-9
8.3 HTPU 프로시저를 사용한 HEVE의 확장 8-13
8.4 편집 세션 중의 **HTPU** 프로시저의 컴파일링 .. 8-14

8.5 HEVE 초기화 파일의 사용 8-15
8.6 HTPU 명령 파일의 사용 8-18
8.7 HTPU 섹션 파일의 사용 8-19
8.8 기동 파일의 사용 비교 8-20
8.9 논리명을 사용한 **HEVE** 키패드의 정의 8-21

제 **9** 장 **HEVE** 외의 사항 9-1
9.1 DCL 명령의 실행 9-1
9.2 부프로세스의 작성 9-3

제 **10** 장 **HEVE DECwindows Motif/Hangul** 인터페이스 사용 10-1
10.1 시작하기 전에 10-1
10.2 DECwindows Motif/Hangul 인터페이스를 사용한 HEVE 호출 10-2
10.2.1 메뉴 언어 선택 10-3
10.2.2 한글 문자의 입력 10-4
10.3 DECwindows Motif/Hangul 인터페이스 화면 구성 10-4
10.4 커서이동 및 화면이동 10-7
10.4.1 커서 위치 지정 10-7
10.4.2 명령 창에서의 클릭 10-8
10.4.3 상태 행에서의 클릭 10-8
10.4.4 화면 이동 막대 위에서의 클릭 10-8
10.4.5 마우스를 오프(off)로 설정하기 10-9
10.5 선택 및 클라이언트간 통신 10-10
10.5.1 1차 선택 10-10
10.5.2 2차 선택 10-11
10.5.3 1차 및 2차 이동 10-12
10.5.4 클립보드 10-12
10.6 메뉴를 사용한 편집 10-13

제 **11** 장 저널링 및 복구 11-1
11.1 버퍼 변경 저널링 11-2
11.1.1 버퍼 변경 저널링을 사용한 복구 11-4
11.2 키조작 저널링 및 복구 11-8
11.2.1 키조작 저널링의 가동 11-9
11.2.2 키조작 저널링을 사용한 복구 11-9
11.3 요약 11-10

부록 **A** **HTPU** 프로시저의 예 A-1

부록 **B** **HEVE** 명령 B-1

부록 **C** 지원되는 **HEDT** 기능 C-1

그림 목록

그림 **1-1** **HEVE** 편집 세션의 화면 1-4
그림 **1-2** 새로운 파일 편집할때의 화면 구성 1-6
그림 **1-3** 기존 파일 편집시의 화면 1-7
그림 **1-4** 시스템 버퍼 **$CHOICES$**를 보여 주는 화면 1-9
그림 **1-5** 도움말 보기의 예 1-16
그림 **3-1** **RIGHT INDENT**의 예 3-10
그림 **3-2** **RIGHT INDENT**의 예 3-11
그림 **5-1** **SYMBOL** 명령 입력 후의 화면 5-7
그림 **5-2** 화면에 기호가 표시된 예 5-7

그림 **6-1** **TWO WINDOWS** 명령 사용 후의 화면 6-7

그림 **6-2** **SPLIT WINDOW 3** 명령 사용 후의 화면 6-8

그림 **6-3** **DELETE WINDOW** 명령 사용 후의 화면 6-9

그림 **6-4** **RHYMES.DAT**의 내용을 보여주는 화면 6-10

그림 **6-5** **TWO WINDOWS** 명령을 사용한 후의 화면 6-11

그림 **6-6** **GET FILE SCHEDULE.DAT** 명령을 사용한 후의 화면 6-12

그림 **6-7** **SCHEDULE.DAT**의 **4**번째 행을 선택한 후의 화면 .. 6-13

그림 **6-8** 선택한 원문을 제거한 후의 화면 6-14

그림 **6-9** **RHYMES.DAT** 버퍼에 원문을 삽입한 후의 화면 6-15

그림 **6-10** **ONE WINDOW** 명령을 사용한 후의 화면 6-16

그림 **10-1** **HEVE DECwindows Motif/Hangul** 편집 세션에서의 화면 구성 10-5

표 목록

표 **1-1** 편집 키와 그 기능 1-12

표 **2-1** 커서 이동을 위한 **HEVE** 명령 2-1

표 **2-2** 원문 지우기를 위한 **HEVE** 명령 2-11

표 **2-3** 원문을 복구하기 위한 **HEVE** 명령 2-13

표 **2-4** 표준 선택에서 사용할 수 있는 기능 2-15

표 **2-5** **HEVE**의 상자 선택 명령 2-19

표 **2-6** **BOX CUT** 및 **BOX PASTE**에서의 상자 채우기 모드의 효

과 요약 표 2-25
표 **3-1** 원문 조정 명령 3-12
표 **4-1** **HEVE**의 탐색 명령 4-1
표 **5-1** 선 그리기 모드에서의 제어 키 5-2
표 **5-2** 상자 그리기 모드에서의 제어 키 5-3
표 **7-1** **HEVE** 키 명 7-2
표 **7-2** **Motif-**스타일 키 정의 7-9
표 **7-3** **MB1**에 대한 마우스 단추 정의 7-11
표 **8-1** 모든 버퍼에 대한 **HEVE** 내정 글로벌 설정값 8-3
표 **8-2** **HEVE** 내정 버퍼 특유의 설정값 8-7
표 **8-3** 속성 설정을 위한 **HEVE** 명령 8-9
표 **8-4** 속성 저장 명령의 요약 8-12
표 **10-1** 상태 행의 서로 다른 구성 요소 위에서 **MB1**을 클릭한 효과 10-8
표 **10-2** 파일 풀다운 메뉴 10-14
표 **10-3** 편집 풀다운 메뉴 10-15
표 **10-4** 형식화 풀다운 메뉴 10-16
표 **10-5** 탐색 풀다운 메뉴 10-17
표 **10-6** 열람 풀다운 메뉴 10-18
표 **10-7** 옵션 풀다운 메뉴 10-19
표 **10-8** 선택 팝업 메뉴 10-20

표 **10-9** 무선택**(No-Selection)** 팝업 메뉴 10-21
표 **B-1** 원문편집 B-2
표 **B-2** 상자 편집 B-5
표 **B-3** 커서 이동 및 화면이동 B-6
표 **B-4** 형식화 B-8
표 **B-5** 탐색 B-10
표 **B-6** 창 및 화면 B-11
표 **B-7** 파일 및 버퍼 B-12
표 **B-8** 키 정의 B-15
표 **B-9** 범용 명령 B-16
표 **B-10** 설정 B-17
표 **C-1** **HEDT** 명령과 **HEVE** 명령간의 대응 관계 C-1
표 **B-1** 원문편집 B-2
표 **B-2** 상자 편집 B-5
표 **B-3** 커서 이동 및 화면이동 B-6
표 **B-4** 형식화 B-8
표 **B-5** 탐색 B-10
표 **B-6** 창 및 화면 B-11
표 **B-7** 파일 및 버퍼 B-12
표 **B-8** 키 정의 B-15
표 **B-9** 범용 명령 B-16

표 **B-10** 설정 B-17

표 **C-1** **HEDT** 명령과 **HEVE** 명령간의 대응 관계 C-1

서 문

사용자

이 지침서는 HEVE의 새로운 사용자와 이 편집기의 기능을 재학습하고자 하는 사용자를 위한 것입니다.

구성

이 지침서는 11개의 장과 3개의 부록으로 구성되어 있습니다.

- 제1장은 HEVE의 입문편입니다. 이 장에서는 HEVE의 개요를 제공하고 있으며, HEVE의 호출법과 HEVE의 기본적인 편집 기능을 설명합니다.

- 제2장은 원문의 편집을 다룹니다. 이 장에서는 다양한 편집 키와 커서 이동 명령의 사용과 원문의 이동, 삭제 및 복구와 같은 전형적인 원문 편집 기능의 수행 방법을 설명합니다.

- 제3장은 원문의 형식화를 다룹니다. 이 장에서는 여백과 탭 및 단어 줄 바꿔쓰기 설정을 통하여 원문을 형식화하는 방법을 설명합니다.

- 제4장은 원문의 탐색을 다룹니다. 이 장에서는 편집 버퍼에 특정 원문 집어넣기와 찾아낸 원문의 편집 작업 수행에 대하여 설명합니다.

- 제5장은 선/상자 그리기 및 기호에 관한 장입니다. 이 장에서는 선과 사각 도형 그리기와 특수 기호의 삽입 방법을 설명합니다.

- 제6장은 2개 이상의 파일 편집을 위한 버퍼와 창 사용을 다룹니다. 이 장에서는 2개 이상의 파일을 편집하기 위하여 버퍼를 사용하는 방법을 설명합니다. 그리고 여러 개의 창을 사용하여 동시에 서로 다른 파일들을 편집하고 열람하는 것에 대해 설명합니다.

- 제7장은 HEVE 키 정의를 다룹니다. 이 장에서는 HEVE 명령 실행을 위한 키 정의와 런 순서(learn sequence) 작성 방법을 설명합니다.

- 제8장은 HEVE의 설정을 다룹니다. 이 장에서는 특정 편집 어플리케이션을 위한 사용자 자신의 편집 환경의 설정 방법을 설명합니다.

- 제9장은 HEVE 외적인 것을 다룹니다. 이 장에서는 DCL 명령의 실행과 HEVE 내에서의 부프로세스 생성 방법을 설명합니다.

- 제10장은 HEVE DECwindows Motif/Hangul Interface 사용을 다룹니다. 이 장에서는 DECwindows Motif/Hangul Interface에 대해 설명하며, 또한 한글 풀다운 메뉴와 팝업 메뉴, 그리고 DO 명령과 사전 정의된 키를 사용하여 HEVE 명령을 실행하는 방법을 설명합니다.

- 제11장은 저널링과 복구를 다룹니다. 이 장에서는 시스템 장애로 인하여 편집 세션이 중단되었을 경우에, 저널 파일을 사용하여 편집 작업 내용을 복구하는 방법을 설명합니다.

- 부록 A는 HTPU 프로시저의 예제들을 수록하고 있습니다. 여기에서는 커서가 화면 경계에 도달하였을 때에 화면을 자동으로 이동시킬 수 있는 HTPU 프로시저의 예를 보여줍니다.

- 부록 B는 HEVE 명령을 수록하고 있습니다. 여기에서는 범주에 따른 HEVE 명령의 표와 각 명령에 대한 간단한 설명을 보여줍니다.

- 부록 C는 지원되는 HEDT 기능을 다루고 있습니다. 여기에서는, HEDT 전용 기능과 HEVE가 제공하는 것으로서 상이한 구문에서도 동일하거나 또는 유사한 작업을 수행할 수 있는 등가 기능을 비교합니다.

규약

규약	의미
OpenVMS, VMS	용어 OpenVMS와 VMS는 OpenVMS 운영체제를 뜻합니다.
OpenVMS/Hangul	용어 OpenVMS/Hangul은 한글용 OpenVMS I64와 한글용 OpenVMS Alpha 운영체제를 뜻합니다.

한글	이 지침서에서, 단어 "한글"은 한글,한자 및 기타 그래픽 문자를 포함하여 한글 문자 세트에 정의된 집단적인 명칭으로 사용됩니다.
RETURN	캐리지 리턴.
	예제와 형식에서, 별도로 언급하지 않는 한, 사용자 입력의 각 행의 끝에서는 암시적으로 캐리지 리턴이 발생하는 것으로 합니다. 따라서, 실제 입력시에는 각 행의 끝에서 캐리지 리턴을 눌러야 합니다.
키 기호	예제에서, 키와 순차 키는 PF2 및 CTRL/Z와 같은 기호로 나타냅니다.
CTRL	CTRL/x는 CTRL 키를 누른 상태에서 다른 키를 동시에 누르는 것을 나타냅니다. 예를 들어, CTRL/C, CTRL/Y, CTRL/O와 같습니다.
. . .	수직 생략 부호는 특정 명령에 대해 시스템의 응답 데이터를 화면에 전부 표시하지 않거나, 또는 사용자가 입력해야 하는 것을 생략했다는 것을 뜻합니다. 즉, 논의중인 주제와 직접 관련이 없는 정보를 생략할 때에 사용합니다.
...	수평 생략 부호는 추가의 매개변수, 값, 또

는 정보가 입력될 수 있다는 것을 뜻합니다.

()　　형식 설명에서, 괄호(())는 사용자가 선택할 수 있다는 것을 표시할 때 사용합니다.

[]　　형식 설명에서, 괄호([])는 사용자가 임의로 선택할 수 있다는 것을 나타낼 때에 사용합니다. 이 경우, 사용자는 아무것도 선택하지 않거나 또는 그 중 하나만 선택하거나, 또는 그 중 모두를 선택할 수 있습니다(그러나, 이 괄호가 파일 명세의 디렉토리명 구문에서 사용되거나 또는 할당문의 부문자열 명세에서 사용된 경우에는 선택적이지 않습니다).

{}　　형식 설명에서, 괄호[{}]는 사용자가 택일하여 선택할 수 있다는 것을 나타낼 때에 사용합니다. 즉, 이 괄호 내에 표시된 것 중 반드시 하나를 선택해야 합니다.

관련　책자

- *HTPU*와 *HEVE* 참조서
- *Extensible Versatile Editor Reference Manual*
- *Guide to the DEC Text Processing Utility*
- *DEC Text Processing Utility Reference Manual*
- *Open VMS/Hangul* 사용자 지침서

제 1 장

HEVE 입문

이 장에서는 HEVE의 호출 방법과 HEVE의 기본적인 편집 기능에 대한 설명을 포함하여 HEVE(Hangul Extensible Versatile Editor)의 개요를 제공합니다.

1.1 서론

HEVE는 HTPU(Hangul Text Processing Utility) 상에 구축된 범용 편집기입니다. HEVE는 한글 문자[1] 뿐만 아니라, ASCII 문자[2] 의 편집 기능도 제공합니다. HEVE는 다국 문자 세트(MCS)를 제외한 EVE의 모든 기능을 제공하며, 또한 한글 처리를 위하여 일부의 신규 및 향상된 기능이 보강되었습니다. HEVE를 사용하여 선과 상자를 그릴 수 있고, 또한 선/상자 그리기와 다른 활용을 위하여 편집중인 파일에 특수 기호를 삽입할 수 있습니다. 동일한 편집 세션 중에 서로 다른 파일들을 열람하고 편집할 수 있습니다. 또한, 사용자 자신의 편집 환경에 맞게 HEVE를 손쉽게 조정할 수도 있습니다.

HEVE V3.1에서는, HEVE가 대부분의 HEDT의 기능을 지원하도록 기능이 향상되었습니다. 만일 사용자가 HEDT 편집기에 익숙하다면, HEDT를 에뮬레이트하도록 HEVE를 손쉽게 설정할 수 있습니다.

[1] 한글 문자는 화면에서 2개의 열을 차지합니다.
[2] ASCII 문자는 화면에서 1개의 열을 차지합니다.

또한, 사용자의 워크스테이션에서 DECwindows Motif 창 작업 환경을 사용하여서도 HEVE V3.1을 실행할 수 있습니다. DECwindows Motif 사용자 인터페이스를 통한 HEVE의 사용 설명에 대해서는 제10장에서 별도로 다룹니다.

1.1.1 2개국어 도움말

HEVE에서는 온라인 도움말을 사용할 수 있습니다. 이 도움말은 영어 또는 한글로 표시됩니다. 도움말에 사용할 언어는 HEVE를 호출하기 전에 HANGULGEN 유틸리티를 사용하여 설정할 수 있습니다. HANGULGEN 유틸리티의 사용에 대한 보다 상세한 내용은 OpenVMS/Hangul 사용자 지침서의 제5장을 참조하십시오. 예를 들어, HEVE에서 도움말이 영어로 표시되게 하려면, HEVE를 호출하기 전에 다음의 명령을 사용할 수 있습니다.

```
$ RUN HSY$SYSTEM:HANGULGEN

HANGULGEN> SET/OUTPUT=ASCII_MSG
```

한글 도움말을 사용하려면 다음 명령을 사용하십시오.

```
$ RUN HSY$SYSTEM:HANGULGEN

HANGULGEN> SET/OUTPUT=HANGUL_MSG
```

HEVE의 도움말 기능 사용에 관한 상세한 사항은 1.6절의 'HEVE에서 도움말 보기'를 참조하십시오.

1.1.2 2개국어 메시지

HEVE에서는 메시지가 영어 또는 한글로 화면에 표시됩니다. HEVE를 호출하기 전에 HANGULGEN 유틸리티를 사용하여 이 메시지에 사용할 언어를 설정할 수 있습니다.

1.2 편집 세션의 시작과 종료

편집 세션을 시작하려면, DCL 명령 EDIT/HTPU를 사용하여 HEVE를 호출하면 됩니다. 편집 세션에서는 새로운 파일이나 기존 파일을 작성하고 편집할 수 있습니다. 편집 세션을 종료할 때는, Do 키(VT100-계열 단말기에서는 PF4 키)를 누른 후 EXIT 또는 QUIT 명령을 입력하면 됩니다. HEVE를 종료하면, 일반적으로 새로운 파일이 생성되거나, 또는 기존 파일의 새로운 버전이 생성됩니다. 이와 다른 방법으로, F10 키(VT200-계열 또는 VT300-계열 단말기에서) 또는 CTRL/Z 키를 눌러서 종료할 수도 있습니다.

1.2.1 HEVE의 호출

HEVE를 호출하려면, 다음과 같이 DCL 명령 행에서 간단히 EDIT/HTPU를 입력하면 됩니다.

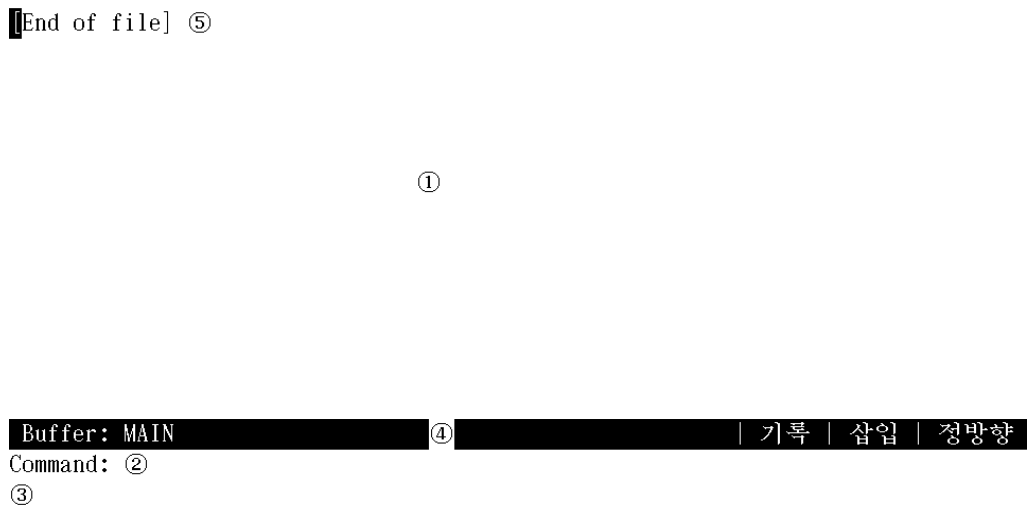
```
$ EDIT/HTPU
```

그러면, HEVE가 이름이 MAIN인 빈 버퍼를 생성합니다. 이후 사용자는 입력을 할 수 있으며, 또한 종료시의 출력 파일을 지정해 줄 수 있습니다. 이와 달리, HEVE 호출시에 단일 또는 일련의 출력

파일을 지정해 줄 수도 있습니다. 이에 관해서는 이 절의 후반부에서 설명합니다.

일단 HEVE를 호출하면, 화면이 다음과 같이 표시됩니다.

그림 **1-1 HEVE** 편집 세션의 화면



1. 주 창
   편집중인 버퍼를 보여 줍니다. 버퍼는 데이터를 일시적으로 저장하거나 처리하는 컴퓨터의 기억장소입니다.

2. 명령 창
   HEVE 명령을 입력하는 창으로서, 1행으로 된 영역입니다. 사용자가 Do 키를 누르면, HEVE가 커서를 명령 창으로 넣습니다. 다음에, HEVE는 'Command:' 프롬프트를 화면에 표시하

여 사용자에게 HEVE 명령의 입력을 요구합니다.

3. 메시지 창

HEVE의 정보 메시지가 표시되는 창으로서, 1행으로 된 영역입니다. 이것은 HEVE 화면의 맨 아래 행입니다.

4. 상태 행

사용자가 열람 또는 편집 중인 버퍼에 관한 정보를 보여 주는 영역입니다.

o 버퍼명

HEVE 호출시에 입력 파일을 지정하지 않으면, HEVE가 MAIN이라는 이름으로 빈 버퍼를 만듭니다. 입력 파일을 지정해 주면, HEVE는 그 파일명과 파일 유형을 버퍼명으로 사용합니다.

o 상태

판독 전용 또는 쓰기

o 편집 모드

원문의 입력, 삽입 또는 겹쳐쓰기 모드를 나타냅니다.

o 탐색 및 다른 작업에서의 방향

정방향 또는 역방향을 나타냅니다.

5. 파일의 끝 표시기

파일의 끝 표시기인 [End of file]은 HEVE 버퍼의 끝을 정의합니다. 이것은 화면에서 보이기만 할뿐, 파일의 부분을 이

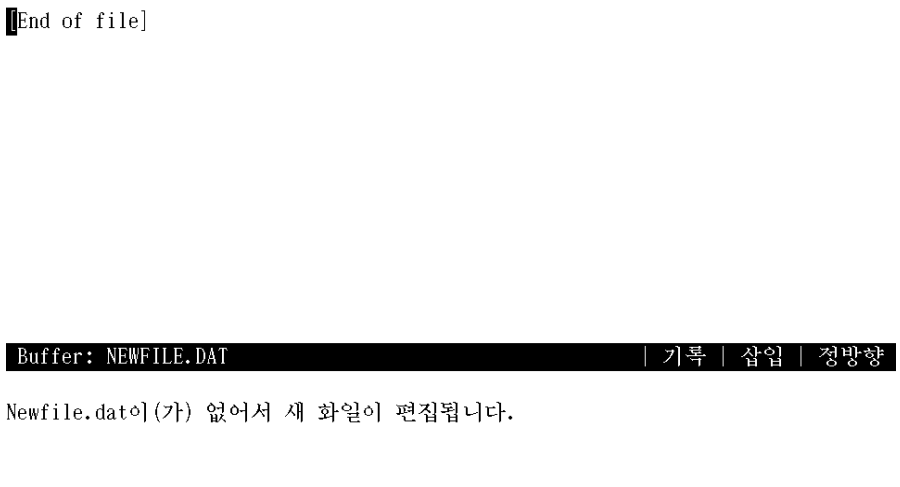
루지는 않습니다. 이 표시기는, 많은 행을 포함하고 있는 버퍼의 맨 앞부분을 볼 때에는 보이지 않을 수도 있습니다.

내정 커서 시작 위치는 주 창의 왼쪽 윗 모서리입니다.
HEVE를 호출하고, 이름이 NEWFILE.DAT인 새로운 파일을 생성하려면, 다음과 같이 명령을 입력하십시오.

```
$ EDIT/HTPU NEWFILE.DAT
```

그러면, HEVE가 NEWFILE.DAT를 위한 새로운 버퍼를 생성하고, 그 파일의 이름과 유형을 버퍼명으로 사용합니다. 이 명령은 그림 1-2와 같은 화면을 생성합니다.

그림 1-2 새로운 파일 편집할때의 화면 구성



이름이 SCHEDULE.DAT인 기존 파일의 편집을 위해 HEVE를 호출하는 경우, 다음의 명령을 입력할 수 있습니다.

```
$ EDIT/HTPU SCHEDULE.DAT
```

이 때, 화면의 편집 창에서 나타나는 파일의 내용은 그림 1-3에서와 같이 보입니다.

그림 1-3 기존 파일 편집시의 화면

```
CENTER LINE
버퍼의 현재 가장자리들 사이에 있는 한 행의 원문을, 행의 처음에 공백을
삽입하여, 중앙에 오도록 합니다.
단계:
1.      중앙에 놓고자 하는 행의 임의의 위치에 커서를 놓으십시오.
2.  CENTER LINE 명령을 사용하십시오(WPS/Hangul 키패드에서는 GOLD-C를
        누르십시오).
[End of file]

Buffer: SCHEDULE.DAT                                | 기록 | 삽입 | 정방향

7 lines read from file DATA$:[USER.CVMS.HANGUL]SCHEDULE.DAT;1
```

또한, HEVE 호출시에 2개 이상의 파일을 동시에 열 수 있습니다. 예를 들어, HEVE를 호출하여 파일 NEWFILE.DAT(새로운 파일), JABBER.DAT(기존 파일) 및 SCHEDULE.DAT(기존 파일)를 편집하는 경우에, 다음의 명령을 사용할 수 있습니다.

```
$ EDIT/HTPU NEWFILE.DAT, JABBER.TXT, SCHEDULE.DAT
```

그러면, HEVE가 목록 내의 각 입력 파일을 위한 버퍼를 생성한 후에, 이들 기존 파일의 내용을 해당 버퍼에 복사합니다.

입력 파일 목록에서 모든 파일을 복사한 후에, HEVE는 목록 내의 첫번째 파일, 즉 위의 예에서는 NEWFILE.DAT에 작업 위치를 설정합니다.

입력 파일의 명세 지정에 와일드카드 문자를 사용할 수 있습니다. HEVE에서의 와일드카드 사용 규칙은 DCL에서의 와일드카드 사용 규칙과 동일합니다. 파일 명세에 와일드카드를 사용하였을 때에, 일치되는 것이 하나만 있으면 그 파일만 화면에 표시되고, 2개 이상의 파일이 일치되면 HEVE가 일치되는 파일의 목록을 화면에 표시한 후에 보다 완전한 파일 명세를 제공하도록 프롬프트합니다. 일치되는 것이 없으면, HEVE가 그러한 내용의 메시지를 화면에 표시하고 이름이 MAIN인 빈 버퍼로 들어갑니다. 예를 들어, 다음 명령을 사용하여 HEVE를 호출한 경우에,

```
$ EDIT/HTPU *.DAT
```

일치되는 파일이 2개 이상 있으면, HEVE가 그림 1-4에서와 같이 이름이 $CHOICES$인 시스템 버퍼에 일치 가능한 파일 목록을 넣어서, 사용자가 그 중에서 원하는 것을 선택할 수 있도록 합니다.

그림 1-4 시스템 버퍼 **$CHOICES$**를 보여 주는 화면

```
[End of file]


Buffer: MAIN                                          | 기록 | 삽입 | 정방향
Data$:[user.cvms.hangul]rhymes.dat;1
Data$:[user.cvms.hangul]schedule.dat;1


Buffer: $CHOICES$
Command: get file DATA$:[USER.CVMS.HANGUL]*.DAT;
불확실한 화일명: DATA$:[USER.CVMS.HANGUL]*.DAT;
```

일치되는 파일이 없으면, HEVE가 다음과 같은 메시지를 화면에 표시한 후 MAIN 버퍼로 들어갑니다.

```
No files matching: *.txt
```

HEVE의 일부 내정 설정값들은 사용자 단말기 설정값에 따릅니다. 예를 들어, HEVE의 내정 화면 폭은 단말기 설정값(DCL 명령 SET TERMINAL/WIDTH에 따른 값)과 동일합니다. 내정 오른쪽 여백값은 단말기에서 설정한 폭보다 1열 작습니다. 따라서, 단말기에서 설정한 폭이 80열이면 내정 오른쪽 여백은 79열입니다.

1.2.2 HEVE 종료

두 개의 다른 명령 EXIT와 QUIT을 사용하여 HEVE의 편집 세션을 정상적으로 종료할 수 있습니다.

기존 원문에서 편집한 원문을 저장하고 세션을 종료하려면, Do 키(VT100-계열 단말기에서는 PF4 키)를 누른 후에 EXIT 명령을 입력하십시오. 이와 달리, F10 키(VT200-계열 또는 VT300-계열 단말기에서)를 누르거나 또는 CTRL/Z를 눌러서 종료할 수도 있습니다.

1.2.2.1 편집 내용의 저장

현재 버퍼를 수정하면, HEVE가 원래 버전과 동일한 파일명과 파일 유형에 버전 번호만 1을 증가시킨 새로운 버전의 파일을 생성합니다. 예를 들어, 파일명이 FUN.DAT;1인 파일을 수정한 후에 EXIT 명령을 수행하면, 이름이 FUN.DAT;2인 파일이 출력됩니다.

1.2.2.2 편집 내용의 취소

편집 내용을 저장하지 않고 세션을 종료하려면, Do 키를 누른 후 QUIT 명령을 입력하십시오. 예를 들어, 이름이 FUN.DAT인 파일을 수정한 후에 QUIT 명령을 입력하면, HEVE가 다음과 같이 사용자에게 수정한 내용을 포기할 것인지를 물어옵니다.

```
Buffer modifications will not be saved, continue quitting [Yes]?
```

이때, 편집한 내용을 저장하지 않고 종료하려면 Y를 입력하십시오. 그렇지 않고 마음을 바꾸어, 이대로 종료하지 않으려면 N을 입력하십시오. 그러면 HEVE가 현재 버퍼 작업 상태를 유지합니다.

1.3 중단된 편집 세션으로부터의 복구

시스템에 장애가 발생하거나 또는 사용자가 CTRL/Y를 누르면, 편집 세션이 비정상적으로 종료합니다. 이 경우에, 내정된 절차에 의해 저널 파일이 출력되는데, 이 파일은 이러한 중단 시점 이후에 사용자가 편집 작업을 복구할 수 있도록 편집 내용을 기록합니다.

이 경우, 한번에 한개의 버퍼를 복구할 수 있으며, 다른 편집 세션에서도 버퍼를 복구할 수 있습니다. 예를 들어, 다음의 명령은 HEVE를 호출하여 파일명이 JABBER.TXT인 파일의 원문을 복구합니다.

```
$ EDIT/HTPU JABBER.TXT /RECOVER
```

저널링과 복구에 관한 사항은 이 지침서의 제11장을 참조하십시오.

1.4 기본적인 편집 작업

일단 HEVE를 호출하면 미니 키패드 상의 편집 키, 특정 기능 키, 및 HEVE가 사전에 정의해 놓은 특정 제어 순차 키를 사용하여 간단히 커서의 이동, 원문의 삭제, 버퍼 내에서의 원문 이동 등과 같은 기본적인 편집 작업을 수행할 수 있습니다.

다음 표는 편집 키와 그 기능들을 보여 줍니다.

표 **1-1** 편집 키와 그 기능

키	기능	키	기능
CTRL/A	모드변경(삽입/겹쳐쓰기)	FIND	지정된 문자열 찾기
CTRL/B	이전 HEVE 명령 재호출	INSERT HERE	현재 커서 위치에 선택된 범위의 원문 삽입
CTRL/E	행의 끝으로 커서 이동	REMOVE	선택된 범위의 원문 삽입
CTRL/H	행의 처음으로 커서 이동	SELECT	일정 범위의 원문 선택
CTRL/J	현재 단어의 삭제	PREV SCREEN	이전 화면
CTRL/L	페이지 구분점 삽입	NEXT SCREEN	다음 화면
CTRL/U	행의 처음까지 삭제	F10	HEVE 종료
CTRL/W	화면 재생	F11	방향변경(정 방향 또는 역 방향)
CTRL/Z	HEVE 종료	F12	행 단위로 커서 이동
DELETE	이전 문자 삭제	F13	현재 단어 삭제
↑	위로 이동	F14	모드변경(삽입/겹쳐쓰기)
↓	아래로 이동	→	오른쪽으로 이동
←	왼쪽으로 이동		

1.4.1 한 위치에서 다른 위치로 원문 이동

원문을 버퍼 내에서 원하는데로 이동하거나 또는 일정 범위의 원문을 삭제하려면, HEVE의 오려붙이기(cut and paste) 기능을 사용하십시오. 이때, 일정 범위의 원문을 이동 또는 삭제하기 전에, 먼저 범위를 선택하여야 합니다. 선택은 미니 키패드의 Select 키를 누른 다음에, 커서를 이동하여 원문을 선택하면 됩니다. 그러면, 선택된 원문이 역상으로 강조표시됩니다. 그 다음에 Remove 키를 누르면 선택된 원문이 버퍼에서 제거됩니다.

Insert Here 키는 임의의 커서 위치로 원문을 삽입합니다. Select, Remove, Insert Here 키를 순서대로 누르면, 원문을 원하는 위치로 이동시킬 수 있습니다.

1.4.2 원문 탐색

버퍼내에서 지정된 문자열을 탐색하려면, 미니 키패드 상의 Find 키를 누르십시오. 그러면 HEVE가 상태 행에 표시되어 있는 현재 탐색 방향에 따라 Forward Find: 또는 Reverse Find: 프롬프트를 화면에 표시합니다. 이 프롬프트에서 탐색할 문자열을 입력할 수 있습니다.

탐색 방향은 F11 키를 눌러서 전환시킬 수 있습니다.

1.5 HEVE 명령의 입력

HEVE 명령은 Command: 프롬프트에서 입력할 수 있습니다. HEVE

명령을 입력할 때에는 항상 다음의 3단계를 수행해야 합니다.

1. Do 키 (또는 VT100 계열 단말기에서는 PF4 키)를 누르십시오. 그러면, HEVE가 Command: 프롬프트를 화면에 표시합니다.

2. 위의 프롬프트가 표시된 후에, HEVE 명령을 입력하십시오.

3. Return 키 또는 Do 키를 누르십시오.

그러면 HEVE가 입력된 명령을 수행하거나 또는 다음에 조치해야 할 내용을 프롬프트로 화면에 표시합니다.

HEVE 명령을 입력할 때에 약어를 사용할 수 있는데, 이 때는 모호하지 않도록 입력해야 합니다. 사용자가 입력한 명령과 2개 이상의 명령이 일치하는 경우에는, HEVE가 일치되는 명령의 목록을 화면에 표시하여 사용자가 원하는 것을 선택할 수 있도록 합니다. HEVE는 명령을 해석할 때에 대/소문자를 구분하지 않습니다. 명령 행에서는 대/소 문자를 구분없이 사용할 수 있습니다.

마지막으로 입력한 명령을 반복할 때에는, Do 키를 두 번 누르면 그 명령이 입력됩니다. 따라서, 해당 명령을 다시 입력하는 번거로움을 피할 수 있습니다.

사용가능한 HEVE 명령과 각 명령에 관한 간략한 설명에 대한 표를 부록 B에 수록하여, 신속한 참조가 가능하도록 하였습니다.
HEVE의 고급 편집 기능에 관한 상세한 정보는 후속 장들에서 논의합니다.

1.6 HEVE의 도움말 보기

HEVE는 다음과 같이 3가지 종류의 도움말 주제를 가지고 있습니다.

- 명령 주제 - HEVE 명령을 설명하고, 해당 명령에 대한 사용 예제와 단계 목록, 그리고 명령에 연계된 모든 키의 목록(EDT 키 또는 WPS 키) 및 기타 정보를 제공합니다.

- 키패드 주제 - HEVE 명령에 연계되어 있지 않은 EDT 키패드와 WPS 키패드의 키를 설명합니다.

- 정보 주제 - 내정값, 키 명칭 및 초기화 파일 등과 같은 편집기의 동작 또는 기능 영역을 설명합니다.

임의의 시점에서 도움말을 얻는 방법에는 다음 세가지가 있습니다.

- 도움말 메뉴를 사용하는 방법(DECwindows Motif 인터페이스에서만)
  DECwindows Motif 인터페이스에 관한 보다 상세한 사항은 이 지침서의 제10장을 참조하십시오.

- Help 키(VT100-계열 단말기에서는 PF2 키)를 사용하는 방법
  Help 키는 HELP KEYPAD로 정의되어 있으며, 이 키패드는 키가 정의된 내용에 따라 미니 키패드, 키패드 또는 둘 다를 보여주는 키패드 도형을 화면에 그립니다. 그 다음에 도움말을 원하는 키를 누르면 됩니다. 이 도형에서는 <Gold> 키 기능(있

는 경우)이 역상으로 표시됩니다. Gold 키에 관한 보다 상세한 사항은 이 지침서의 7.3절을 참조하십시오.

- Do 키(VT100-계열 단말기에서는 PF4 키)를 누르고 명령 행에 Help 명령을 입력하는 방법
  HELP 명령 다음에 도움말을 원하는 주제를 입력하면 됩니다. 주제를 입력하지 않으면, HEVE가 주제 목록을 화면에 표시합니다.

예를 들어, 다음 명령은 Find 명령에 대한 도움말이 표시됩니다.

```
Command: HELP FIND
```

이 경우, Find 명령의 도움말은 이름이 HELP인 시스템 버퍼로 들어가고, 그림1-5에서와 같이 화면으로 나타납니다.

그림 1-5 도움말 보기의 예

```
FIND

 사용자가 지정한 원문 문자열을 찾아 발견하면 이(found text)를 강조표시
 합니다. 이 기능은 2 바이트 문자를 지원합니다.

 키:    HEVE        VT100       EDT             WPS/Hangul
        ------------------------------------------------------------
        FIND        PF1         GOLD-PF3        키패드의 GOLD-,

 단계:

    1.  FIND 명령을 사용하십시오(위의 키 목록을 참조하십시오).

    2.  찾고자 하는 원문을 입력하십시오. 문자형(case)에 상관없이 동일한
        문자열을 버퍼내에서 찾으려고 하면 모두 소문자로 지정하십시오.
        문자형까지 정확히 일치되는 것만 찾으려면 소문자와 대문자의
        혼합형이나 대문자를 사용하십시오. 2 바이트 문자에 있어서,
        FULL_FORM 문자열은 단일 바이트의 ASCII 형식내의 동일 문자열에
        의해 찾아집니다.

 예제:
 Buffer: HELP                    To see more, use: | Prev Screen | Next Screen
목록을 보려면 주제명이나 ?를 입력하십시오.(끝내려면 RETURN을 누르십시오.):
```

대부분의 도움말 주제는 관련 명령이나 다른 주제의 목록을 가지고 있습니다. 한 주제의 도움말을 보고 있는 중에 다른 주제의 도움말을 보려면, 원하는 주제명을 입력하고 Return 키를 누르면 됩니다. 그리고 주제의 목록을 보려면 물음표(?)를 입력하고 Return 키를 누르면 됩니다.
길게 입력한 도움말 주제를 화면 이동하여 보려면 Prev Screen 키와 Next Screen 키를 누르십시오.

도움말을 종료하고 편집을 재개하려면 Return 키를 누르십시오.

제 2 장

원문의 편집

신규 및 기존 파일의 편집을 위하여 HEVE 명령을 사용할 수 있습니다. 다양한 편집 키와 명령을 사용하여 커서 이동이나 원문의 이동, 삭제 및 복구 등과 같은 전형적인 원문 편집을 수행할 수 있습니다.

2.1 커서 이동

HEVE를 호출하고 화면에 파일 또는 버퍼를 불러낸 다음에는, 파일 내에서 자유롭게 커서를 이동할 수 있습니다.
다음 표에 커서 이동을 위한 모든 HEVE 명령과 그 기능들을 요약해 놓았습니다.

표 **2-1** 커서 이동을 위한 **HEVE** 명령

명령	기능
MOVE UP	커서를 한번에 1행 단위로 위로 이동합니다.
MOVE DOWN	커서를 한번에 1행 단위로 아래로 이동합니다.
MOVE LEFT	커서를 한번에 1문자 단위로 왼쪽으로 이동합니다.
MOVE LIGHT	커서를 한번에 1문자 단위로 오른쪽으로 이동합니다.

표 **2-1** (계속) 커서 이동을 위한 **HEVE** 명령

명령	기능
MOVE BY WORD	현재 버퍼 내에서, 커서를 한번에 1단어 단위로 이동합니다.
MOVE BY LINE	현재 버퍼 내에서, 커서를 한번에 1행 단위로 이동합니다.
MOVE BY PAGE	현재 버퍼 내에서, 커서를 한번에 1페이지 단위로 이동합니다. 페이지의 경계는 버퍼의 FF(용지이송) 표시점 또는 상단/하단입니다. 현재 방향에 FF 표시가 없으면, 커서가 버퍼의 상단 또는 하단으로 이동합니다.
END OF LINE	커서를 현재 행의 끝으로 이동합니다.
START OF LINE	커서를 현재 행의 처음으로 이동합니다.
TOP	커서를 현재 버퍼의 상단으로 이동합니다.
BOTTOM	커서를 현재 버퍼의 하단으로 이동합니다.
LINE	커서를 행 번호에 의해 지정된 행의 처음으로 이동합니다. 예를 들어, LINE 10 명령을 사용하면, 커서가 현재 버퍼 내의 10번째 행의 처음으로 이동합니다.
WHAT LINE	현재 행의 번호를 보여줍니다.
MARK	현재 버퍼에 보이지 않는 표시기를 삽입합니다 2.1.3 절을 참조하십시오.
GO TO	지정된 표시기 위치에 커서를 위치시킵니다. 2.1.3 절을 참조하십시오.

표 **2-1** (계속) 커서 이동을 위한 **HEVE** 명령

명령	기능
CHANGE DIRECTION	상태 행에 표시된 버퍼의 방향(정방향 또는 역방향)을 변경합니다. 버퍼의 방향은 MOVE BY LINE, MOVE BY PAGE, 및 MOVE BY WORD 명령에 영향을 미칩니다.
FORWARD	버퍼의 방향을 정방향(즉, 오른쪽 또는 아래쪽 방향)으로 설정합니다.
REVERSE	버퍼의 방향을 역방향(즉, 왼쪽 또는 윗쪽 방향)으로 설정합니다.
SET CURSOR BOUND	커서 이동을 고정시킵니다. 2.1.1 절을 참조하십시오.
SET CURSOR FREE	내정값입니다. 버퍼 내에서 커서 이동을 자유롭게 합니다. 2.1.1 절을 참조하십시오.
SET DISPLAY CURSOR	내정값입니다. 커서 이동을 화면에 표시되게 합니다. 2.1.2 절을 참조하십시오.
SET NODISPLAY CURSOR	커서 이동을 화면에 표시되지 않게 합니다. 2.1.2 절을 참조하십시오.

상이한 커서 모드(자유커서/고정커서, 화면표시/화면비표시)에 따라, 커서는 서로 다른 동작을 합니다. 이에 대해서는 이하의 소절들에서 보다 자세히 논의됩니다.

2.1.1 커서 이동의 자유/고정 설정

커서 이동을 자유 또는 고정 모드로 설정할 수 있습니다. 자유 커서 모드에서는, 커서를 버퍼 내의 어느 곳으로나 이동시킬 수 있으며 또한 문자가 그 곳에 있는지 여부에 관계없이 원문을 삽입할 수

있습니다. 이것이 내정값입니다.

고정 커서 모드에서는 커서 이동이, HEDT 및 WPS 편집기에서와 같이, 원문의 모양 또는 흐름에 따릅니다. 예를 들어, 행의 끝에 위치해 있을 때 오른쪽 화살표 키를 누르면, 커서가 다음 행의 처음 위치로 이동합니다.

커서 이동을 자유 커서 모드로 설정하려면, SET CURSOR FREE 명령을 사용하고, 고정 커서 모드로 설정하려면, SET CURSOR BOUND 명령을 사용하여야 합니다.

편집 세션 내내 또는 대부분에서 고정 커서 모드로 사용하려면, HEVE 초기화 파일에 SET CURSOR BOUND 명령을 넣어 놓으십시오. 사용자 자신의 편집 환경의 설정에 관한 보다 상세한 사항은 이 지침서의 8.5절을 참조하십시오.

2.1.2 커서 화면표시 모드의 설정

또 다른 커서 모드인 '커서 화면표시' 모드도 커서 동작에 영향을 미칩니다.

커서 화면표시 모드가 가동 정지되어 있으면, 커서가 V3.1 이전 버전에서와 같이 동작합니다. 커서가 지나가는 문자의 문자 경계가 정렬되어 있지 않은 경우에, 상향 또는 하향 화살표 키를 누르면 커서가 왼쪽으로 이동합니다.

예를 들어, 버퍼 내에 있는 원문이 아래의 예와 같고, 커서의 시작 위치가 표시된 것과 같을 때, 하향 화살표 키를 3번 누르면, 최종

커서 위치는 처음 위치와 동일한 열에 머무르지 않고, 그림에서와 같이 왼쪽으로 이동합니다.

```
          |--- Starting cursor position
          v
찾고자 하는 원문을 입력하십시오
  찾고자 하는 원문을 입력하십시오
 찾고자 하는 원문을 입력하십시오
찾고자 하는 원문을 입력하십시오
          ^
          |--- End cursor position
[End of file]
```

커서 화면표시 모드가 가동되어 있으면, 커서가 왼쪽으로 이동되는 문제가 해결됩니다. 위의 예에서, 커서 화면표시 모드가 설정되어 있으면, 동일하게 하향 화살표 키를 3번 눌렀을 때에 최종 커서 위치는 다음과 같습니다.

```
          |--- Starting cursor position
          v
찾고자 하는 원문을 입력하십시오
  찾고자 하는 원문을 입력하십시오
 찾고자 하는 원문을 입력하십시오
찾고자 하는 원문을 입력하십시오
          ^
          |--- End cursor position
[End of file]
```

커서 화면표시 모드가 'on'으로 설정되어 있는 경우에 하향 화살표 키를 n번 누른 다음에 상향 화살표 키를 n번 누르면, 커서가 원래의 위치로 복귀합니다. 그러나, 커서 화면표시 모드가 가동되어 있지 않으면 그렇게 되지 않습니다.

커서 화면표시 모드가 'on'으로 설정되어 있으면, 화면에 표시되는 커서는 항상 한글 문자의 첫번째 열에 정렬됩니다.
커서 화면표시(내정값) 모드를 가동하려면 SET DISPLAY CURSOR 명령을 사용하고, 가동 중지시키려면 SET NODISPLAY CURSOR 명령을 사용하십시오.

2.1.3 보이지 않는 표시기의 사용

보이지 않는 표시기를 사용하여 버퍼 내의 현재 위치에 표시를 하면, 나중에 해당 위치로 복귀할 수 있습니다. 이 방법은 큰 파일을 편집할 때에 매우 유용한 것으로서, 편집 세션 중 나중에 특정한 커서 위치로 복귀하는 것이 용이합니다.

현재 위치에 표시하는 것은 MARK 명령 다음에 표시기명을 입력하면 됩니다. 명령 행에 표시기명을 입력하지 않으면, HEVE가 표시기명을 입력하도록 Mark name: 프롬프트를 화면에 표시합니다. 이때 표시기명을 입력하지 않고 그냥 Return 키만 누르면 명령이 취소됩니다.

표시기명에는 영숫자, 구두문자, 공백, 탭을 포함하여 1자 이상을 사용할 수 있으며, 또한 한글 문자도 사용할 수 있습니다. 표시기명에서는 대/소 문자를 구분하지 않습니다. 그리고, 와일드카드 문자도 표시기 문자로 사용할 수 있습니다. 예를 들어, 표시기명으로

'*'를 사용할 수 있습니다.

지정된 표시기 위치에 커서를 위치시키려면, GO TO 명령 다음에 표시기명을 입력하면 됩니다. 표시기명을 입력하지 않으면, HEVE가 표시기명을 입력하도록 Go To: 프롬프트를 화면에 표시합니다. 이 때, 표시기명을 입력하지 않고 Return 키만 누르면 명령이 취소됩니다.

표시기명을 입력할 때는 약어를 사용할 수 있는데, 입력된 약어와 일치되는 것이 여러개 있으면 HEVE가 일치되는 표시기명의 목록을 화면에 표시하여 선택할 수 있도록 해줍니다.

SHOW 명령을 사용하면 현재 버퍼 내의 표시기의 명들을 볼 수 있습니다. 예를 들어, 버퍼명이 DEMO.TXT인 버퍼에서 편집 중이고, MARK 명령을 사용하여 이름이 CHAP-1과 CHAP-2인 두 개의 표시기를 만들어 놓았을 때에 SHOW 명령을 사용하면, 화면에 다음과 같이 표시됩니다.

```
HEVE HG V3.1 1992-10-18 18:41
DEMO.TXT 버퍼에 관한 정보

입력 화일: DATA$:[USER.CVMS]DEMO.TXT;1
출력 화일: DATA$:[USER.CVMS]DEMO.TXT;1
저널 화일: DATA$:[USER.CVMS]DEMO_TXT.TPU$JOURNAL;1

    변경되지 않음              왼쪽 가장자리가 1(으)로 설정되었습니다.
    모드: 삽입                 오른쪽 가장자리가 79(으)로 설정되었습니다.
    단락 들여쓰기: 없음        WPS/Hangul 단어랩 들여쓰기: 없음
    기록                       변경 가능한
    방향: 정방향               윈도우 넓이가 80(으)로 설정되었습니다.
    0 행

탭멈춤은 매 8 열마다 설정되어 있습니다.

단어랩: on

표시기:
Chap-1                              Chap-2
```

2.2 편집 모드의 사용

HEVE는 삽입 및 겹쳐쓰기 두 가지 편집 모드를 가지고 있습니다. 버퍼의 편집 모드는 상태 행에 표시됩니다.

현재 버퍼가 삽입 모드로 설정되어 있으면, 입력시에 새로운 원문이 현재 입력 위치에 삽입되고 기존 원문은 오른쪽 또는 아래로 밀려납니다.

현재 버퍼가 겹쳐쓰기 모드로 설정되어 있으면, 입력시에 새로운 원문이 현재 입력 위치에 있는 기존 원문을 대체합니다. 따라서, 겹쳐쓰기 모드를 "대체 모드" 또는 "중복타자 모드"라고도 합니다. 한글 문자는, 동일한 폭의 열만큼 겹쳐쓰기됩니다. 반면에, 예를 들어 하나의 한글 문자에 하나의 ASCII 문자가 겹쳐쓰기 되는 경우와 같이 한글 문자의 일부분만이 겹쳐쓰기 되는 경우에는, 한글 문자의 겹쳐쓰기 되지 않은 부분이 등가의 공백으로 대체됩니다.

커서 화면표시 모드가 가동 중지되어 있고 버퍼가 삽입 모드에 있을 때 커서가 현재 한글 문자에 위치해 있으면, 한글 문자 내에서의 커서 위치에 관계 없이 새로운 원문은 항상 한글 문자의 첫번째 열에 삽입됩니다. 예를 들어, 커서 화면표시 모드가 가동 중지되어 있고 현재 버퍼가 삽입 모드에 있을 때, 버퍼 내의 원문이 아래의 예와 같고 커서가 현재 한글 문자의 2번째 열에 위치해 있는 경우,

```
      Once upon a midnight dreary,
       사용자가 지정한 지원문 문자열을 찾아
       사 as I pondered weak 문자열을 찾아
      weary, over many a quaint
 [End of file]
```

현재 커서 위치에 문자 a를 삽입하게 되면, 그 문자는 아래와 같이 삽입됩니다.

```
      Once upon a midnight dreary,
       사용자가 지a정한 지원문 문자열을 찾아
       사 as I pondered weak 문자열을 찾아
      weary, over many a quaint
 [End of file]
```

커서 화면표시 모드가 가동 중지되어 있고 버퍼가 겹쳐쓰기 모드에 있을 때 커서가 현재 한글 문자에 위치해 있으면, 한글 문자 내에서의 커서 위치에 관계 없이 새로운 원문은 현재 커서 위치에 정확하게 겹쳐쓰기 됩니다. 예를 들어, 커서 화면표시 모드가 가동 중지되어 있고 버퍼가 겹쳐쓰기 모드에 있을 때, 버퍼 내의 원문이 다음의 예와 같고 커서가 현재 한글 문자의 2번째 열에 위치해 있는 경우,

```
      Once upon a midnight dreary,
       사용자가 지정한 지원문 문자열을 찾아
       사 as I pondered weak 문자열을 찾아
      weary, over many a quaint
[End of file]
```

문자 b를 입력하면, 현재 커서 위치에 문자 b가 입력되고 겹쳐쓰기 되지 않은 한글 문자 부분은 공백으로 대체됩니다.

```
      Once upon a midnight dreary,
       사용자가 지 b정한 지원문 문자열을 찾아
       사 as I pondered weak 문자열을 찾아
      weary, over many a quaint
[End of file]
```

2.3 원문 지우기 및 복구

HEVE 편집기에서는 파일에서 원문 지우기와 편집 세션 중 실수의 교정을 용이하게 할 수 있습니다. 실수로 원문을 지운 경우에는, 가장 최근에 지운 원문을 현재 커서 위치에 복구할 수 있습니다.

버퍼에서 원문을 지우려면, 커서를 지울 원문의 위치로 이동시킨 후에 해당 HEVE 명령을 입력하면 됩니다.
다음 표는 원문을 지울 때 쓰이는 HEVE 명령을 보여줍니다.

표 **2-2** 원문 지우기를 위한 **HEVE** 명령

명령	기능
DELETE	커서 왼쪽의 문자를 지웁니다.
ERASE CHARACTER	현재 문자를 지웁니다.
ERASE PREVIOUS WORD[1]	커서 위치에 따라, 이전 단어 또는 현재 단어를 지웁니다. 커서가 단어 사이에 있거나 또는 단어의 처음에 위치해 있으면, 이전 단어를 지웁니다. 커서가 단어 내에 위치해 있으면, 현재 단어를 지웁니다. 커서가 행의 처음에 위치해 있으면, 행 구분 표시를 지우고 현재 행을 위로 올립니다.
ERASE WORD[1]	현재 단어를 지우거나, 또는 단어 사이에 있을 때에는 다음 단어를 지웁니다.
ERASE START OF LINE	행의 맨 처음부터 현재 커서 위치까지의 모든 문자를 지웁니다.
ERASE LINE	현재 커서 위치부터 현재 행의 끝까지 지우고, 다음 행을 현재 행의 끝에 추가시킵니다.

[1]한글 문자의 단어 정의는 영어 문자에서의 단어 정의와 다릅니다. 한글 문자에서의 단어는 단일 행에서 일련의 동일한 부류의 문자들로 정의됩니다. 이러한 문자 세트는 다음의 다섯가지, 즉: ASCII, FULL_FORM, HANGUL, NON_HANGUL(구두점 표시 등) 및 UNSPECIFIED로 분류됩니다. 문자 분류 정의에 관한 상세한 사항은 "HTPU/HEVE 참조서"를 살펴보십시오.

예를 들어, 버퍼 내에 있는 원문이 아래의 예와 같고 커서 위치가 표시된 것과 같을 때,

```
    사용자가 지지정한 지원문 문자열을 찾아
[End of file]
```

ERASE WORD 명령을 사용하면, HEVE는 일련의 한글 문자를 지웁니다.

```
    사용자가 지원문 문자열을 찾아
[End of file]
```

현재 커서 위치에서 ERASE CHARACTER 명령을 사용하면, 현재 문자만 지워집니다.

```
    사용자가 지정한 지원문 문자열을 찾아
[End of file]
```

다음의 표는 원문을 복구하기 위한 HEVE 명령을 보여줍니다.

표 **2-3** 원문을 복구하기 위한 **HEVE** 명령

명령	기능
RESTORE	마지막으로 ERASE 명령을 사용하여 지웠던 것을 현재 커서 위치에 복구합니다. 마지막으로 지운 것이 ERASE LINE을 사용하여 지운 것이면, 이 RESTORE 명령은 RESTOR LINE과 동일한 효과를 가져옵니다. RESTORE는 단일 문자를 복구하지 않습니다.
RESTORE CHARACTER	마지막으로 DELETE 또는 ERASE CHARACTER 명령을 사용하여 지웠던 것을 현재 커서 위치에 복구합니다.
RESTORE WORD	마지막으로 ERASE WORD 또는 ERASE PREVIOUS WORD 명령을 사용하여 지웠던 것을 현재 커서 위치에 복구합니다.
RESTORE LINE	마지막으로 ERASE LINE 또는 ERASE START OF LINE 명령을 사용하여 지웠던 것을 현재 커서 위치에 복구합니다.

2.4 원문 선택

HEVE에서는 표준 선택과 상자 선택 두가지 선택 방식이 있습니다. 표준 선택 방식은 연속적인 범위의 원문을 행으로 선택합니다.

상자 선택 방식은 원문을 열로 선택합니다. 상자 선택과 상자 편집 작업에 관한 상세한 사항은 2.5절을 참조하십시오.

2.4.1 표준 선택의 시작

표준 선택을 시작하려면 SELECT 명령을 사용하거나 또는 미니 키패드에서 Select 키를 누르십시오. 그 다음에 커서를 이동하여 원문을 선택하십시오. 그러면, 커서가 지나가는 부분의 원문이 역상으로 강조 표시됩니다. 그러나 공백 행은 강조 표시되지 않습니다.

커서를 정방향(즉, 오른쪽 또는 아래쪽)으로 이동시키면, 선택 범위가 시작 위치에서 시작하여 최종 커서 위치의 왼쪽 문자에서 끝납니다. 커서를 역방향(즉, 왼쪽 또는 윗쪽)으로 이동시키면, 선택 범위가 시작 위치의 왼쪽 문자에서 시작하여 마지막 커서 위치의 문자에서 끝납니다.

RESET 명령을 사용하거나 또는 SELECT 명령을 다시 한번 반복하면 선택이 취소됩니다.

2.4.2 표준 선택 후의 편집 작업

일정 범위의 원문을 선택한 후에는, HEVE 명령이나 HEDT 키패드 키, 또는 WPS 키패드 키를 사용하여 선택된 범위를 편집할 수 있습니다.

이러한 편집을 위한 기능과 명령 또는 편집 키를 표로 나타내었습니다.

표 **2-4** 표준 선택에서 사용할 수 있는 기능

기능	명령
대/소문자 변경	UPPERCASE WORD LOWERCASE WORD CAPITALIZE WORD HEDT CHNGCASE(GOLD-KP1) WPS UPPER CASE(KP3) WPS LOWER CASE(GOLD-KP3)
문자 크기 변경	FULLFORM WORD HALFFORM WORD
형식화	FILL(FILL RANGE) CONVERT TAB
대체	HEDT REPLACE(GOLD-KP9) WPS REPLACE(GOLD-" 또는 GOLD-')WPS Subs(GOLD-ENTER)
삭제	REMOVE 또는 CUT BOX REMOVE 또는 BOX CUT BOX CUT INSERT BOX CUT OVERSTRIKE HEDT APPEND(KP9) WPS CUT(MINUS)
복사	STORE TEXT 또는 COPY BOX COPY WPS COPY(GOLD-MINUS)
기타 편집	DELETE(삭제 대기) FIND SELECTED OPEN SELECTED SPELL

2.4.2.1 선택된 범위의 원문 삭제 또는 복사

선택된 범위의 원문을 삭제하려면, REMOVE 또는 CUT 명령을 사용하거나, 또는 단순히 미니 키패드 상의 Remove 키를 누르십시오. 그러면, HEVE가 강조 표시된 원문을 화면에서 삭제하여 이름이 INSERT HERE인 시스템 버퍼에 집어 넣습니다. 커서는, 삭제 후 즉시 동일한 원문을 버퍼에 재삽입하기 쉽도록 선택 범위 맨 앞에 위치합니다.

선택된 범위의 원문은 STORE TEXT 또는 COPY 명령을 사용하여 원래의 원문을 버퍼로부터 삭제하지 않고 INSERT HERE 버퍼로 복사해 넣을 수 있습니다. 그러면, HEVE가 강조표시를 취소하고 선택된 원문을 INSERT HERE 버퍼에 복사해 넣습니다.

INSERT HERE 또는 PASTE 명령을 사용하거나, 또는 단순히 미니 키패드의 Insert Here 키를 눌러서, INSERT HERE 버퍼 내의 어느 커서 위치에나 원문을 삽입할 수 있습니다. INSERT HERE 버퍼에 포함된 원문은, 새로운 부분의 원문을 선택한 후 그 새로운 원문을 REMOVE 또는 STORE TEXT 명령을 사용하여 INSERT HERE 버퍼에 집어 넣지 않는 한, 몇번이든지 어느 커서 위치에나 삽입할 수 있습니다.

2.4.2.2 삭제 대기

삭제 대기 기능이 가동되면, DELETE 명령을 사용하거나(또는 Delete 키 <x 를 누르거나), 또는 새로운 원문을 입력할 때 선택한 원문이 지워집니다. 이것은 원문의 블럭을 신속히 지우거나 대체할 때에 유용합니다.

삭제 대기 작업으로 지운 원문은 RESTORE SELECTION 명령을 사용하여 복구할 수 있습니다.

삭제 대기 기능은 $RESTORE$SELECTION$ 버퍼를 사용합니다. 따라서, INSERT HERE 버퍼의 내용에 영향을 미치지 않고, 삭제 대기 기능과 RESTORE SELECTION 명령을 사용하여 원문을 오려붙이기(cut and paste)할 수 있습니다.

삭제 대기 기능은 SET PENDING DELETE 명령을 사용하여 가동시킬 수 있습니다.

삭제 대기 기능을 가동 중지시키려면 SET NOPENDING DELETE 명령을 사용하십시오. 이것이 내정값입니다.

2.5 상자 편집

HEVE에서는 표준 방식의 행으로 뿐만 아니라, 원문을 열로도 선택하여 편집할 수 있습니다. 상자 편집 작업은, 특히 목록이나 또는 표의 일부분을 편집하는데 유용합니다.

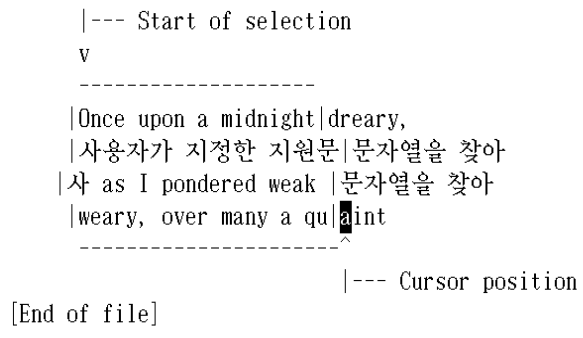
2.5.1 상자 선택의 시작

다른 것이 선택되어 있지 않은 상태라면, BOX SELECT 명령을 사용하여 상자 선택을 시작할 수 있습니다. 만일 다른 종류의 선택 즉, 상자 또는 표준 방식으로 기존에 선택한 것이 있는 경우에는, 이 명령이 그 선택을 취소시킵니다. 커서가 지나가는 부분은, 원문의 상자 부분뿐 아니라, 모든 원문이 굵은체로 강조 표시됩니다.

그 다음에, 커서를 원문의 선택 상자로 이동하여, 상자 선택의 대각편 모서리로 갑니다. 선택 상자의 아래 오른쪽 모서리에 있는 문자는 선택에 포함되지 않습니다. 상자 선택에서는, 상자의 오른쪽 열에 있는 모든 문자도 포함되지 않습니다.

상자의 경계에 한글 문자가 있으면, 이들 문자도 상자 선택에 포함됩니다.

다음 그림에서는, 정방향 상자 선택이 점선으로 표시되고, 현재 커서 위치는 "quaint"의 "a"에 위치해 있습니다. 상자의 왼쪽과 오른쪽에 걸쳐 있는 한글 문자는 상자 선택에 포함됩니다.



2.5.2 상자 선택의 사용

2.4.2에서 열거한, 행으로 선택한 범위에서 동작하는 기능들은 상자 선택에서도 동작됩니다.

아래의 표는 HEVE의 상자 선택 명령들입니다.

표 **2-5** **HEVE**의 상자 선택 명령

상자 선택 명령	기능
BOX SELECT	상자 선택을 시작합니다.
BOX CUT	상자 선택한 것을 잘라냅니다.
BOX PASTE	삽입 버퍼에 상자 내용을 삽입합니다.
BOX COPY	상자 선택한 것을 복사합니다.
SET BOX [NO]SELECT	상자 선택 모드를 가동/중시시킵니다.
SET BOX [[NO]PAD	상자 작업을 겹쳐쓰기/삽입 모드로 설정합니다.
RESTORE BOX SELECTION	삭제 대기된 운문 상자를 복구합니다.
BOX CUT OVERSTRIKE	상자 선택한 것을 겹쳐쓰기 모드로 잘라냅니다.
BOX CUT INSERT	상자 선택한 것을 삽입모드로 잘라냅니다.
BOX PASTE OVERSTRIKE	상자 선택한 것을 삽입 버퍼에 겹쳐쓰기 모드로 삽입합니다.
BOX PASTE INSERT	상자 선택한 것을 삽입 버퍼에 삽입 모드로 삽입합니다.

2.5.2.1 BOX CUT 및 BOX COPY

버퍼에서 선택한 원문을 제거한 상자 형태로 INSERT HERE 버퍼로 이동시키려면, BOX CUT 명령을 사용하십시오.

선택한 원문을 제거하지 않은 상자 형태로 INSERT HERE 버퍼로 복사하려면, BOX COPY 명령을 사용하십시오.

이때, 선택 방식이 상자 선택일 필요는 없습니다. 단지, Select 키를 눌러서 선택을 시작한 다음에 커서를 이동하여 원문을 선택하면 됩니다. 그 다음에, BOX CUT 또는 BOX COPY 명령을 사용하여 원문 상자를 오려내거나 복사하십시오. 이들 명령은, 원문 선택의 시작점과 끝점을 상자 형태로 지정해 주면, 자동적으로 표준 선택을 상자 선택으로 변환시킵니다. 이와는 달리, BOX SELECT 명령을 사용하여 상자 선택을 시작한 다음에, Remove 키를 눌러서 원문 상자를 오려낼 수도 있습니다.

내정에 의해, BOX CUT 명령은 원문의 열 정렬을 상자의 오른쪽에 맞추기 위하여, 오려낸 영역을 공백으로 채웁니다. 예를 들어, 버퍼에 있는 원문이 아래와 같을 때, 한글 문자가 선택 상자의 왼쪽 또는 오른쪽 경계에 걸치게 되므로, 원문 선택을 사각형의 상자 형태로 할 필요가 없습니다.

```
        |--- Start of selection
        v
    abcd사용자가 지지정한 지원문 문자열을efghk
    abc사용자가 지지정한 지원문 문자열을defghk
    abcde사용자가 지지정한 지원문 문자열을fghk
    adcd사용자가 지지정한 지원문 문자열을efghk
                                ^
                                |--- Cursor position
[End of file]
```

BOX CUT 명령을 사용하고 나면, HEVE가 원문 상자를 버퍼에서 제거하여 버퍼에 공백을 채워 넣고, 다음과 같이 커서를 제거된 원문의 왼쪽 모서리에 위치시킵니다.

```
        |--- Cursor position
        v
    abcd사                      자열을efghk
    abc사                        열을defghk
    abcde                        자열을fghk
    adcd사                      자열을efghk

[End of file]
```

BOX CUT 이후에 버퍼에 빈 공간을 남겨 두지 않으려면, BOX CUT INSERT 명령을 사용하십시오. 이 명령은 원문 상자를 버퍼에서 제거한 후에, 빈 간격을 메우기 위하여 원문을 상자의 오른쪽으로 이동시킵니다.

따라서, 위의 예와 같은 경우에, BOX CUT INSERT 명령을 실행한 후의 버퍼는 아래와 같이 됩니다.

```
        |--- Cursor position
        v
    abcd사자열을efghk
    abc사열을defghk
    abcde자열을fghk
    adcd사자열을efghk

[End of file]
```

2.5.2.2 BOX PASTE

상자 선택 또는 표준 선택 작업 결과로 생긴, INSERT HERE 버퍼의 내용을 버퍼 내의 현재 커서 위치에 집어 넣으려면, BOX PASTE 명령을 사용하십시오. 이 때, INSERT HERE 버퍼의 내용이 사각형 형태의 원문일 필요는 없습니다.

BOX PASTE와 PASTE 명령간의 차이는, BOX PASTE 명령은 INSERT HERE 버퍼 내의 각 행을, 각 행의 끝에 있는 라인 피드(LF) 문자를 무시하고, 별개체로 버퍼에 삽입하거나 겹쳐쓰기한다는 것입니다.

내정에 의해, BOX PASTE 명령은 INSERT HERE 버퍼 내의 원문의 각 행을 버퍼 내의 대응되는 "상자 위치"로 기존 원문에 겹쳐쓰기합니다. 예를 들어, 버퍼에서 제거된 원문 상자가 아래와 같이 현재 INSERT HERE 버퍼에 있고,

```
    ONCE UPON A MIDNIGHT--
    사용자가 지지정한 지원
   사 AS I PONDERED WEAK 사
    WEARY, OVER MANY A QUA
[End of file]
```

다음 페이지의 원문이 사용자 버퍼에 있는 원문이라면

```
   aaaaaaaaaaaaaattttttttttttttttt
   aaaaaaaaaaaaaattttttttttttttttt
   aaaaaaaaaaaaaattttttttttttttttt
   aaaaaaaaaaaaaattttttttttttttttt
[End of file]
```

BOX PASTE 명령 사용 후에는 기존 원문이 아래와 같이 겹쳐쓰기됩니다.

```
   aaaaONCE UPON A MIDNIGHT--tttt
   aaaa사용자가 지지정한 지원tttt
   aaa사 AS I PONDERED WEAK 사ttt
   aaaaWEARY, OVER MANY A QUAtttt
[End of file]
```

이때, 기존 원문에 겹쳐쓰기되지 않게 하려면, BOX PASTE INSERT 명령을 사용하여 기존 원문이 상자의 오른쪽으로 밀려나게 하십시오.

위의 예에서, BOX PASTE INSERT 명령을 사용하면, 결과가 다음과 같이 됩니다.

```
    aaaaONCE UPON A MIDNIGHT--aaaaaaaaattttttttttttttttt
    aaaa사용자가 지지정한 지원aaaaaaaaattttttttttttttttt
    aaa사 AS I PONDERED WEAK 사aaaaaaaaaattttttttttttttttt
    aaaaWEARY, OVER MANY A QUAaaaaaaaaaattttttttttttttttt
[End of file]
```

2.5.2.3 SET BOX [NO]PAD

상자 채우기 모드는 상자 선택 작업을 삽입 모드로 할 것인지 또는 겹쳐쓰기 모드로 할 것인지를 결정합니다.

상자 채우기 모드(box pad mode)를 가동되게 하려면, SET BOX PAD 명령을 사용하십시오. 이것이 내정값입니다. 이 모드에서는 BOX CUT과 BOX PASTE 명령이 각각 2.5.2.1절과 2.5.2.2절에서 설명한 내정 동작과 동일하게 동작합니다.

상자 채우기 모드를 가동 중지시키려면, SET BOX NOPAD 명령을 사용하십시오. 그 후의 BOX CUT 및 BOX PASTE 명령의 동작은 현재 버퍼의 편집 모드(삽입 또는 겹쳐쓰기)에 따릅니다. 버퍼가 삽입 모드에 있을 때에는, BOX CUT 명령이 원문 상자를 제거하고 상자 오른쪽의 원문을 왼쪽으로 당깁니다(BOX CUT INSERT와 동일한 효과를 갖습니다). BOX PASTE 명령은 상자를 삽입한 후, 기존 원문을 오른쪽으로 밀어냅니다(BOX PASTE INSERT와 동일한 효과를 갖습니다). 현재 모드가 겹쳐쓰기 모드이면, SET BOX PAD 모드에서와 동일하게 동작합니다.

다음 표에 BOX CUT 및 BOX PASTE 명령에서의 상자 채우기 모드의 효과를 요약해 놓았습니다.

표 **2-6 BOX CUT** 및 **BOX PASTE**에서의 상자 채우기 모드의 효과 요약 표

설정값	BOX CUT 및 BOX PASTE에서의 효과
SET BOX PAD (내정값)	BOX CUT에서는, 원문 상자를 오려낸 후, 오려낸 영역에 공백을 채워서 원문의 열 정렬을 상자 오른쪽에 맞춥니다. BOX PASTE에서는, 기존 원문에 겹쳐쓰기 형태로 원문 상자를 삽입합니다. 이들 효과는 삭제 대기 후 RESTORE BOX SELECTION을 사용하여 상자 선택을 지울 때에도 적용됩니다.
SET BOX NOPAD	상자 편집은 버퍼의 편집 모드에 따릅니다 삽입 모드에서는, BOX CUT가 원문 상자를 오려낸 후에 상자 오른쪽의 원문을 왼쪽으로 당겨서 공백을 없앱니다. BOX PASTE는 기존 원문을 오른쪽으로 밀어내고 원문 상자를 삽입합니다. 겹쳐쓰기 모드에서는, 이들 효과가 SET BOX PAD에서와 동일합니다. 이들 효과는 삭제 대기 후 RESTORE BOX SELECTION을 사용하여 상자 선택을 지울 때에도 적용됩니다.

현재의 상자 채우기 모드에 관계없이, 겹쳐쓰기 명령(BOX CUT OVERSTRIKE 및 BOX PASTE OVERSTRIKE)이나 또는 삽입 명령(BOX CUT INSERT 및 BOX PASTE INSERT)을 사용하여, 겹쳐쓰기 또는 삽입 동작이 되게 할 수 있습니다.

2.5.2.4 SET BOX [NO]SELECT

HEVE에서는 표준 선택 모드와 상자 선택 모드를 전환시킬 수 있습니다.

상자 선택 모드는 SET BOX SELECT 명령으로 가동됩니다. 이 모드에서는, 표준 선택 명령(및 연계 키)들이 표준 원문 선택 대신에, 원문 상자를 선택, 오려내기, 복사 및 삽입합니다. 예를 들어, 상자 선택 모드가 가동되어 있을 때, Select 키를 누르면 BOX SELECT 명령을 사용한 것과 동일한 효과를 갖습니다. 상자 선택 모드에서는 표준 선택 방식의 작업이 없습니다. 일반적으로, SET BOX SELECT는 상자 편집용 키를 정의/재정의하지 않고서 표 편집과 같이 열 단위로 오려내기와 삽입 작업을 빈번하게 할 경우에 사용합니다.

상자 선택 모드를 가동 중지시키려면, SET BOX NOSELECT 명령을 사용하십시오. 이것이 내정 모드입니다. 상자 선택 또는 표준 선택이 기동중일 때에는 SET BOX NOSELECT 명령을 사용할 수 없습니다.

제 3 장

원문 형식화

HEVE는 여백, 탭 및 단어 줄 바꿔쓰기 설정을 통하여 원문을 형식화하는 명령들을 제공합니다. HEVE에서는 원문 조정, 원문에서 필요 없는 공백 제거, 원문의 문자형 변경, 선택된 범위의 원문 들여쓰기 및 페이지 구분점의 삽입이 가능합니다. 이하의 소절들에서는 이러한 점들에 대하여 상세히 논합니다.

3.1 여백 설정

왼쪽 및 오른쪽 여백은 SET LEFT MARGIN 및 SET RIGHT MARGIN 명령 다음에 원하는 여백값을 지정하여 설정합니다. 여백값을 지정해 주지 않으면, HEVE가 사용자에게 값을 입력하도록 프롬프트합니다. 예를 들어, 현재 버퍼의 왼쪽 여백을 5로 설정하려면 다음의 명령을 사용하십시오.

```
Command: SET LEFT MARGIN 5
```

왼쪽 여백은 0보다는 크고 오른쪽 여백보다는 작아야 합니다. 왼쪽 여백의 내정값은 1입니다. 오른쪽 여백은 왼쪽 여백보다 커야 합니다. 내정에 의해, 오른쪽 여백은 단말기의 폭 설정값보다 1만큼 작습니다. 여백 설정은 버퍼에 따라 다릅니다. 즉, 서로 다른 버퍼들에서는 서로 다른 여백 설정값을 갖습니다.

3.2 단어 줄 바꿔쓰기 설정

내정에 의해, HEVE는 원문 입력시 커서가 오른쪽 여백을 지나갈 때, 오른쪽 여백을 넘어가는 단어는 자동적으로 다음 행의 왼쪽 여백 위치로 넘겨 줍니다. 즉, 자동적으로 단어 줄 바꿔쓰기를 합니다. 자동 단어 줄 바꿔쓰기는 SET WRAP 명령으로 가동되며, SET NOWRAP 명령을 사용하면 가동 중지됩니다.

이 명령은, 폭이 넓은 다중열 표 또는 점차적으로 들여쓰기되는 프로그램 문장과 같이, 오른쪽 여백을 넘어가는 매우 긴 행들을 편집할 때 유용합니다.

단어 줄 바꿔쓰기는 버퍼 속성입니다. 즉, 한 버퍼에서는 SET NOWRAP을 사용하고, 다른 버퍼에서는 SET WRAP으로 사용할 수 있습니다.

3.3 탭 설정

HEVE에서는, Tab 키를 눌렀을 때의 효과를 결정하는 탭 멈춤, 버퍼에서 탭이 나타나는 방식, 그리고 탭 모드를 설정할 수 있습니다.
탭 멈춤은 SET TAB AT 또는 SET TAB EVERY 명령을 사용하여 설정합니다.

- SET TAB AT
  현재 버퍼에서 탭을 멈추고자 하는 열을 설정합니다.

- SET TAB EVERY
  동일한 간격으로 탭 멈춤을 설정합니다. 내정 탭 멈춤 값은 8입니다.(즉, 9, 17, 25열... 등입니다.)

탭 멈춤은 버퍼에 따라 다르게 설정할 수 있습니다. 탭 멈춤을 변경하면, 버퍼 내의 기존의 모든 탭 문자에 영향을 줍니다.
편집 중 탭 문자의 모양을 설정하려면, SET TAB INVISIBLE 또는 SET TAB VISIBLE 명령을 사용할 수 있습니다.

- SET TAB INVISIBLE(내정값)
  탭 문자를 보이지 않게 합니다. 즉, 탭 문자가 빈 공백으로 나타납니다.

- SET TAB VISIBLE
  탭 문자를 화면에 나타나게 합니다. 즉,(수평 탭 문자) 다음에 TAB 문자의 화면 폭만큼을 채워주는 어미점(trailing dots)이 표시됩니다.

이 설정값은 모든 버퍼에 영향을 미칩니다.
탭 모드는 SET TAB INSERT, SET TAB MOVEMENT 또는 SET TAB SPACES 명령을 사용하여 설정합니다.

- SET TAB INSERT(내정값)
  Tab 키를 누르면, 탭 문자를 삽입하고 커서와 행의 모든 기존 원문을 다음 탭 멈춤 위치로 이동시킵니다.

- SET TAB MOVEMENT

  Tab 키를 누르면, 아무것도 삽입하지 않고 커서만 다음 탭 멈춤 위치로 이동시킵니다. 이때, 다음 탭 멈춤이 복수 바이트 문자 내에 있으면, HEVE는 커서를 다음 문자의 문자 경계로 이동시킵니다.

  이것은 표나 다중열 목록과 같은 탭-정렬 원문 작업에서 유용합니다.

- SET TAB SPACES

  Tab 키를 누르면, 탭 문자 대신에 해당 숫자 만큼의 공백을 삽입하고 커서와 행의 모든 기존 원문을 다음 탭 멈춤 위치로 이동시킵니다. 이것은 서로 다른 장치에 인쇄 또는 화면표시될 원문을 편집할 때 유용합니다. 왜냐하면, 프린터나 화면표시 장치의 탭 멈춤 설정에 관계없이 탭 간격 조정이 동일하게 되기 때문입니다.

이 설정값은 모든 버퍼에 영향을 주지만, 기존 탭에는 영향을 주지 않습니다.

이 외에, CONVERT TAB 명령은 탭 문자를 범위, 상자, 또는 버퍼에서의 해당 숫자만큼의 공백으로 대체합니다. 이것은 원문이 서로 다른 탭 멈춤값으로 장치에 인쇄 또는 화면표시되도록 HEVE에서 상이한 탭 멈춤을 설정하였지만 표, 목록 및 기타 탭-정렬 원문이 편집시에 동일하게 표시되도록 할 때에 유용합니다.

3.4 단락 들여쓰기

단락 들여쓰기는 작성 또는 재형식화하는 각 단락의 첫번째 행에서

원하는 만큼의 공백을 추가하거나 차감합니다. 예를 들어, 통신문 작성시 각 단락을 5칸씩 들여쓰거나 또는 목록 작성을 위하여 부(-)의 들여쓰기 값을 줄 수도 있습니다.

현재 버퍼의 단락 들여쓰기는 SET PARAGRAPH INDENT 명령 다음에 원하는 단락 들여쓰기 값을 지정하여서 설정합니다. 이 명령 행에 단락 들여쓰기 값을 지정해 주지 않으면, 그 값을 입력하도록 HEVE가 프롬프트합니다. 내정값은 0입니다(또는 들여쓰기를 하지 않습니다). 이 명령은 기존의 원문에 영향을 주지 않습니다. 이 설정값은 새로운 단락 또는 FILL 명령으로 재형식화하는 단락에만 적용됩니다.

단락 들여쓰기는 왼쪽 여백과 관계됩니다. 예를 들어, 왼쪽 여백이 1이고 단락 들여쓰기가 4로 설정되어 있다면, 각 단락의 최초행은 제5열에서 시작하고 나머지 행들은 제1열에서 시작됩니다. 이 예에서 원문의 공백은 다음과 같습니다.

```
    xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxx
```

음수의 들여쓰기 값을 사용하면, 점표시 목록 작성에 사용하는 "매달린" 단락을 설정할 수도 있습니다. 예를 들어, 왼쪽 여백을 8로 설정하고 단락 들여쓰기 값을 -3으로 지정해 주면, 각 단락의 최초행은 제5열에서 시작됩니다. 이렇게 하면, 아래와 같이 원문 왼쪽에 번호 또는 "점" 표시가 붙은 형식의 목록을 만들 수 있습니다.

```
xxxxxxxxxxxxxxxxxxxxxxxx
   xxxxxxxxxxxxxxxxxxxx
   xxxxxxxxxxxxxxxxxxxx
```

단락 들여쓰기와 왼쪽 여백값의 합은 오른쪽 여백값보다 작아야 하고, 1 이상이어야 합니다.
단락 들여쓰기는 버퍼 특유의 속성입니다. 따라서, 편집할 각 버퍼에 서로 다른 설정값을 줄 수 있습니다.

3.5 원문 채움

원문 채움(filling text)을 수행하면, 버퍼의 현재 여백에 맞추어 한 행에 최대 갯수의 단어가 수용되게 합니다. 이것은 특히, 단어를 지우거나 또는 원문을 삽입(paste)한 후에, 또는 여백을 변경한 후에 원문 줄 올리기(rewrap)할 경우에 유용합니다. 채움은 선택된 범위 또는 원문 상자, 또는 현재 단락을 대상으로 할 수 있습니다.

- 선택된 범위 채움
  몇 개의 행 또는 단락, 또는 버퍼 전체를 선택한 후에 FILL 또는 FILL RANGE 명령을 사용하여 선택된 원문을 재형식화할 수 있습니다. 범위를 채울 때에는 버퍼의 현재 여백을 사용합니다.

- 상자 채움
  먼저 원문의 상자를 선택한 후에 FILL 또는 FILL RANGE 명령을 사용합니다. 상자를 채우면, 버퍼의 여백에 관계없이 상자의 폭을 여백으로 사용하여 상자 내의 원문을 줄 올리기합니다. 따라서, 인접한 원문에 영향을 주지 않고서 목록이나 표의 부분을 재형식화할 수 있습니다.

- 단락 채움
  원문을 선택하지 않으면, FILL 명령이 현재 여백을 사용하여 현재 단락을 재형식화한 후에, 커서를 그 단락의 끝으로 이동시킵니다. HEVE에서는, 단락이 다음의 어느 하나와 경계를 이룹니다.

  - 공백 행
  - 버퍼의 상단 또는 하단
  - 페이지 구분점(행 시작 위치에 있는 폼 피드)
  - 행 시작 위치에 있는 DIGITAL Standard Runoff(DSR) 명령(.LE; 등)
  - 행 시작 위치에 있는 DECdocument 태그(<LE> 또는 <EMPHASIS> 등)

  내정에 의해, HEVE는 RUNOFF 명령 또는 DECdocument 태그로 행이 시작되도록 재형식화하지 않습니다. SET FILL TAGS 명령을 사용하면, HEVE가 RUNOFF 명령 또는 DECdocument 태그를 행 시작 위치에 채우도록, 즉 그것들을 단락 경계의 목록에서 제거하도록 할 수 있습니다. RUNOFF 명령 또는 DECdocument 태그를 행 시작 위치에 단락 경계로 복구시키려면, SET FILL NOTAGS 명령을 사용하십시오.

범위 또는 상자를 채우면 공백 행과 페이지 구분점이 단락 경계로 유지되어, 재형식화할 때에 여러개의 단락 또는 버퍼 전체를 선택할 수 있습니다.

단락 들여쓰기 설정값이 0이 아닌 경우에는, 범위가 단락의 맨 앞 또는 행 시작 위치에서 시작되지 않는 한, FILL 또는 FILL

RANGE 명령을 사용하여 범위를 재형식화할 수 없습니다. 그렇지 않으면, 다음의 메시지가 표시됩니다.

```
Invalid range enclosure specified.
```

HEVE는 일부 문자들을 예외 문자로 정의하는데, 대부분의 복수 바이트 구두점이 이에 속합니다. 예외 문자에는 FILL_NOT_BEGIN 문자와 FILL_NOT_END 문자의 두 가지 유형이 있습니다. FILL_NOT_BEGIN 문자는 FILL 명령 사용시 행의 처음 위치에 나타나지 않게 할 문자입니다. FILL_NOT_END 문자는 FILL 명령 사용시 행의 끝에 나타나지 않게 할 문자입니다. FILL_NOT_BEGIN 문자와 FILL_NOT_END 문자는, 각각 다음 명령을 사용하여, 메시지 창에 나타나게 할 수 있습니다.

```
Command: HTPU MESSAGE (GET_INFO (SYSTEM, "FILL_NOT_BEGIN"))
Command: HTPU MESSAGE (GET_INFO (SYSTEM, "FILL_NOT_END"))
```

FILL_NOT_END 문자가 행의 끝에 나타나지 않게 하면, 이 유형의 문자들이 다음 행으로 이동됩니다. FILL_NOT_BEGIN 문자가 행의 처음 위치에 나타나지 않게 하면, 이 유형의 문자들이 이전 행으로 이동됩니다. 따라서, 오른쪽 여백의 끝까지 채워지지 않는 행이 생길 수도 있습니다.

HEVE는, FILL 명령이 사용되었을 때, 단 하나의 예외 문자만 오른쪽 여백을 넘어갈 수 있도록 합니다.

3.6 원문 들여쓰기

HEVE는 선택된 범위의 행의 재형식화를 위한 LEFT INDENT와 LIGHT INDENT 명령을 제공하여, 선택된 범위의 왼쪽 또는 오른쪽 여백이 지정된 열까지 들여쓰기되도록 합니다.

선택된 범위의 원문을 왼쪽 들여쓰기하려면, 먼저 왼쪽 들여쓰기 될 원문의 첫번째 행의 들여쓰기 할 열에 커서를 위치시켜야 합니다. 그 다음에, 들여쓰기 할 원문을 선택하고 LEFT INDENT 명령을 사용합니다. 이 명령은, 예를 들어, 목록 내의 단락을 형식화할 때에 유용합니다. 이 명령의 효과는 HEDT의 NDTL 명령의 효과와 유사합니다.

예를 들어, 다음 원문에서 "o" 기호 다음의 원문을 들여쓰기하려면, 먼저 커서를 "o" 기호 다음의 문자 "T"에 위치시킨 후 Select 키를 누르십시오. 그 다음에, 아래 화살표 키를 두번 눌러서 들여쓰기 할 원문을 선택합니다.

```
Usage notes :
o The mode of the buffer is shown in the status line. It also
affects some commands and keys such as DELETE, ERASE CHARACTER,
and RESTORE CHARACTER.
```

그 다음에, LEFT INDENT 명령을 사용하여 원문을 들여쓰기하면, 결과가 다음과 같이 됩니다.

Usage notes :

o The mode of the buffer is shown in the status line. It also affects some commands and keys such as DELETE, ERASE CHARACTER, and RESTORE CHARACTER.

선택된 범위의 원문을 1열만큼 오른쪽 들여쓰기하려면, 먼저 커서를 들여쓰기 될 원문이 시작되는 열에 위치시킨 다음에 범위를 선택하고 RIGHT INDENT 명령을 사용하십시오. 이 명령은 예를 들어, 도표에서 단락 중간에 일부의 공백을 넣을 때에 유용합니다. 이 명령의 효과는 HEDT의 NDTR 명령의 효과와 유사합니다.

예를 들어, 아래와 같은 단락 중간에 도표를 삽입하려면, 먼저 커서를 들여쓰기가 수행될 열에 위치시킨 후 원문을 선택한 다음에 RIGHT INDENT 명령을 사용하십시오.

그림 **3-1 RIGHT INDENT**의 예

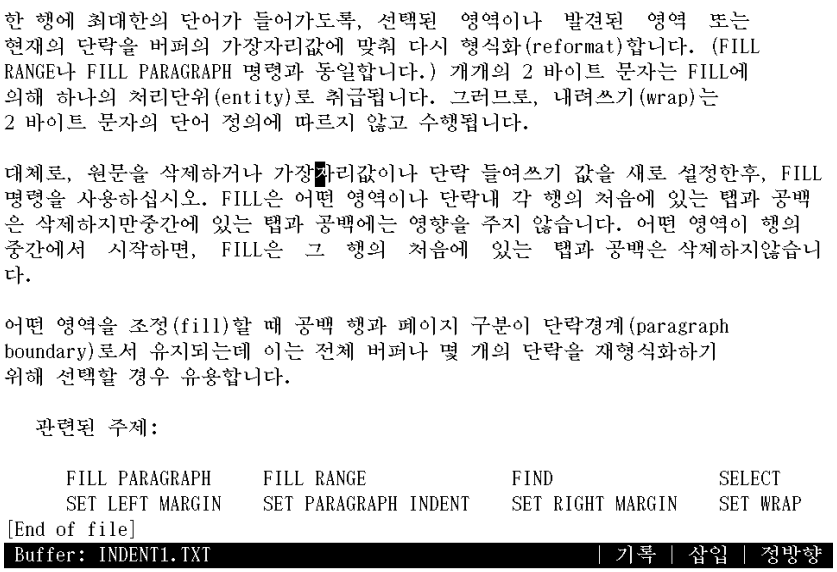
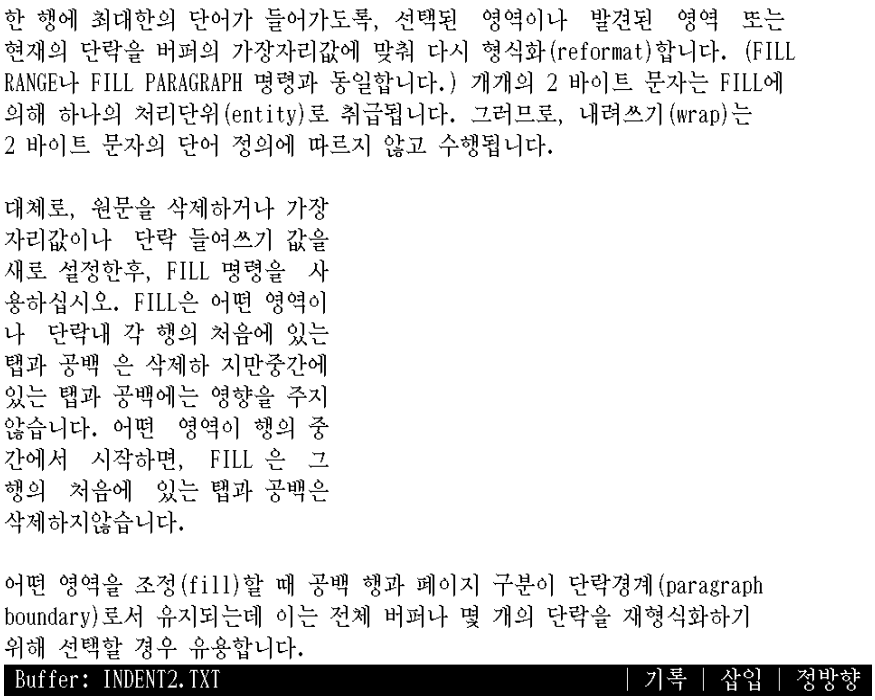


그림 **3-2 RIGHT INDENT**의 예



3.7 원문 조정

HEVE에서는 버퍼의 여백 설정에 따라 원문을 조정할 수 있습니다. 다음 페이지의 표는 HEVE의 원문 조정 명령과 그 기능들을 나타낸 것입니다.

표 **3-1** 원문 조정 명령

명령	기능
CENTER LINE	행의 처음에 공백을 삽입하여, 현재 행을 버퍼의 왼쪽 여백과 오른쪽 여백 사이의 중앙에 위치시킵니다.
LEFT ADJUST	원문(또는 선택된 범위의 행)을 이동시켜서, 원문을 왼쪽 여백에 맞추어 정렬시킵니다. 이 효과는 HEDT의 ADJL 명령의 효과와 유사합니다.
RIGHT ADJUST	원문(또는 선택된 범위의 행)을 이동시켜서, 원문을 오른쪽 여백에 맞추어 정렬시킵니다. 이 효과는 HEDT의 ADJL 명령의 효과와 유사합니다.

3.8 문자형 및 문자 크기의 변경

HEVE에서는 ASCII 원문의 문자형과 문자 크기[1] 를 변경할 수 있습니다. 이러한 변경 작업은 범위, 상자 선택, 또는 단일 단어에 대해 수행할 수 있습니다. 범위를 선택하지 않으면, 현재 단어에 수행됩니다. 이 명령은 한글 문자에 대해서는 효과가 없다는 점을 유의하십시오.

원문의 문자형을 변경하는 명령으로는 CAPITALIZE WORD, UPPERCASE WORD 또는 LOWERCASE WORD가 있습니다.

[1]전자 문자는 2개의 열을 차지하는 영숫자 문자이고, 반자 문자는 1개의 열을 차지하는 ASCII 문자를 말합니다.

- CAPITALIZE WORD
  현재 단어 또는 범위나 상자의 각 단어의 첫번째 문자를 대문자로 만들고, 다른 문자는 소문자로 만듭니다. 예를 들어, 다음 원문의 첫번째 행을 대문자로 만들기 위해, 단어 'digital'에 커서를 이동시킨 후, SELECT 명령을 사용하여 선택을 시작한 후,

  ```
  digital equipment corporation
  maynard, massachusetts
  ```

  커서를 행의 끝으로 이동시킨 다음, CAPITALIZE WORD 명령을 사용하면 결과가 다음과 같이 됩니다.

  ```
  Digital Equipment Corporation
  maynard, massachusetts
  ```

- UPPERCASE WORD
  범위, 상자, 또는 현 단어의 모든 문자를 대문자로 만듭니다.

- LOWERCASE WORD
  범위, 상자, 또는 현 단어의 모든 문자를 소문자로 만듭니다.

원문의 문자 크기를 변경하려면 FULLFORM WORD 또는 HALFFORM WORD 명령을 사용하십시오.

- FULLFORM WORD
  범위, 상자, 또는 현재 단어의 모든 ASCII 문자를 전자 문자

로 만듭니다.

- HALFFORM WORD
  범위, 상자, 또는 현재 단어의 모든 ASCII 문자를 반자 문자로 만듭니다.

3.9 페이지 매김

파일의 매 54행마다 "소프트" 페이지 구분점을 삽입하여 54행의 페이지를 만들 수 있습니다. 소프트 페이지 구분점을 만들면 폼피드(FF) 다음에 널(null) 문자가 생깁니다.

소프트 페이지 구분점을 삽입하려면 PAGINATE 명령을 사용하십시오. 그러면, HEVE가 이전 페이지 구분점 또는 버퍼의 상단으로 되돌아가서 다음 54행 내에 페이지 구분점이 있는지 점검합니다. 이때, 54행 내에 페이지 구분점이 있으면, HEVE가 그것을 지우고 아래로 내려가서 54행 페이지를 표시하기 위하여 소프트 페이지 구분점을 삽입한 후에, 커서를 다음 행의 맨 앞에 위치시킵니다.

새로운 페이지를 시작할 위치에 하드 페이지 구분점을 삽입하려면, INSERT PAGE BREAK 명령을 사용하십시오. 하드 페이지 구분을 만들면 폼피드만 생깁니다.

제 4 장

원문 탐색

HEVE에서는 편집 버퍼 내의 특정 원문을 찾아서 편집 작업을 할 수 있습니다. 원문 탐색 명령은 ASCII 문자뿐만 아니라, 한글 문자도 지원합니다. 표 4-1에 HEVE의 탐색 명령을 수록하였습니다.

표 **4-1 HEVE**의 탐색 명령

명령	기능
FIND	원문의 지정된 문자열을 탐색합니다.
WILDCARD FIND	와일드카드 문자를 사용하여 원문의 패턴을 탐색합니다.
REPLACE	원문의 지정된 문자열을 탐색하여, 찾은 원문을 다른 입력 문자열로 대체합니다.
GLOBAL REPLACE	원문의 지정된 문자열을 탐색하여, 찾아낸 모든 문자열들을 다른 문자열로 대체합니다.
SET FIND CASE [NO]EXACT	HEVE가 탐색 문자열 내의 문자형을 대조할 방법을 결정합니다.
SET FIND FORM [NO]EXACT	HEVE가 탐색 문자열 내의 문자크기(전자 또는 반자)를 대조할 방법을 결정합니다.
SET FIND GENERAL	SET FIND CASE NOEXACT 및 SET FIND FORM NOEXACT와 등가입니다.

표 **4-1** (계속)**HEVE**의 탐색 명령

명령	기능
SET FIND EXACT	SET FIND CASE EXACT 및 SET FIND FORM EXACT와 등가입니다.
SET FIND [NO]WHITESPACE	FIND 및 WILDCARD FIND 명령이 공백, 탭 및 행 구분점을 "빈 공백"으로 처리하도록 설정합니다. 따라서, 두 개 이상의 단어들이 임의 개의 공백이나 탭 또는 행 구분점으로 분리되어 있어도, 한 개의 문자열로 탐색되게 합니다.

4.1 원문 찾기

편집 키패드 내의 원문을 찾으려면, FIND 명령을 사용하거나 또는 단순히 미니 키패드의 Find 키(또는 VT100-계열 단말기에서는 PF1 키)를 누르십시오. 그러면, HEVE가 찾을 원문, 즉 탐색 문자열을 입력하도록 프롬프트합니다. 이때, 탐색 문자열은 모두 7-비트 ASCII 문자이거나, 또는 모두 한글 문자, 또는 ASCII 문자와 한글 문자가 혼합되어도 됩니다.

탐색 문자열에 ASCII 문자가 있으면, 해당 ASCII 문자들을 모두 소문자로 간주하여 문자열들을 찾습니다. 대소문자가 혼합되어 있으면, 문자형이 동일한 문자열들만을 찾습니다(한글 문자에서는 문자형(대/소 문자) 개념이 없습니다).

다음과 같은 예를 들어 보겠습니다.

탐색 문자열	HEVE가 찾은 결과
digital 디지탈	Digital 디지탈 DIGITAL 디지탈 digital 디지탈 ｄｉｇｉｔａｌ 디지탈 Ｄｉｇｉｔａｌ 디지탈 ＤＩＧＩＴＡＬ 디지탈
Digital 디지탈	Digital 디지탈 Ｄｉｇｉｔａｌ 디지탈
DIGITAL 디지탈	DIGITAL 디지탈 ＤＩＧＩＴＡＬ 디지탈
ＤＩＧＩＴＡＬ 디지탈	DIGITAL 디지탈 ＤＩＧＩＴＡＬ 디지탈

위의 경우에, HEVE는 상태 행에 표시된 버퍼의 현재 방향(정방향 또는 역방향)으로 탐색합니다. 편집기가 현재 방향에서 문자열을 찾지 못하고, 반대 방향에서 찾았으면, HEVE가 탐색 방향을 변경할 것인지를 프롬프트합니다. 이때, 반대 방향으로 탐색하게 하려면, Y를 입력하면 됩니다.

원문을 찾으면, HEVE가 커서를 찾아낸 원문의 맨 앞에 위치시킨 후에 문자열을 굵은체로 강조표시합니다. 찾은 원문에 대해서는, 다음과 같은, 선택한 범위에 사용하는 명령의 대부분을 사용할 수 있습니다.

- CAPITALIZE WORD, UPPERCASE WORD, LOWERCASE WORD, FULLFORM WORD 및 HALFFORM WORD와 같은 문자형 및 문자 크기 변경 명령(문자형 변경 명령은 반자 및 전자 ASCII 문자

모두를 지원합니다).

- CUT 또는 REMOVE
- COPY 또는 STORE TEXT
- FILL 또는 FILL RANGE
- OPEN SELECTED

예를 들어, 찾아낸 범위를 CUT 명령을 사용하여 제거하여 INSERT HERE 버퍼에 저장할 수 있습니다. 강조표시를 취소하려면, 커서를 찾아낸 원문밖으로 이동시키거나 또는 RESET 명령을 사용하십시오.

동일한 문자열을 다시 또 찾으려면, FIND NEXT 명령을 사용하거나 또는 Find 키를 두번 누르십시오.

4.2 원문 대체

원문을 탐색하여 대체하려면, REPLACE 명령을 사용하십시오. 이 명령은 특히, 긴 파일 전체에서 철자가 틀린 단어를 전부 찾아내어 교정하려고 할 때에 유용합니다. 이 명령을 사용하면, HEVE가 대체할 문자열과 새로운 문자열을 각각 입력하도록 프롬프트합니다. 기존 문자열과 새로운 문자열에 한글 문자 또는 영어와 한글이 혼합된 문자를 사용할 수 있습니다.

HEVE는 기존 문자열(대체할 문자열)을 탐색할 때에, 버퍼의 현재

방향으로 먼저 시작합니다. 문자열을 찾으면 HEVE가 찾은 문자열을 강조표시하고, 다음의 프롬프트를 화면에 표시하여 그 문자열을 새로운 문자열로 대체할 것인지를 묻습니다.

```
Replace? Type Yes, No, All, Last, or Quit:
```

이 프롬프트에서, 다음 중의 하나로 응답하여야 합니다.

응답	효과
Yes	이번에 찾은 문자열 하나만 대체하고, 다음 문자열을 탐색합니다(이것이 내정값입니다. 이 경우 Return 키만 눌러도 됩니다).
No	이번에 찾은 문자열을 건너 뛰고, 다음 문자열을 탐색합니다.
All	반대 방향으로 기존 문자열을 탐색하도록 하지 않는 한, 더 이상 프롬프트하지 않고 찾아지는 모든 문자열을 대체합니다.
Last	이번에 찾은 문자열을 대체하고, 탐색을 중지합니다.
Quit	취소합니다.(이번에 찾은 문자열을 건너 뛰고 중지합니다)

프롬프트 응답없이 기존 문자열이 찾아지는 대로 모두 새로운 문자열로 대체시키려면, GLOBAL REPLACE 명령을 사용하십시오. 이 명령은 REPLACE 명령 사용시 프롬프트에 ALL을 응답한 것과 동일합니다.

REPLACE 명령은 먼저 FIND 명령을 사용하여 대체될 문자열을 탐색합니다. REPLACE 명령에서의 SET FIND 명령의 효과는 FIND의

효과와 동일합니다. 상세한 사항은 FIND 명령을 참조하십시오.

목표 문자열이 탐색되면(찾아지면), 탐색된 문자열이 다음 범주 중의 하나에 속하고 새로운 문자열의 알파벳 문자가 모두 소문자이면서 반자인 경우에, 탐색된 문자열의 문자형과 문자 크기에 따라 대체됩니다.

- 모든 알파벳 문자가 소문자이면서 반자
- 모든 알파벳 문자가 대문자이면서 반자
- 모든 알파벳 문자가 소문자이면서 전자
- 모든 알파벳 문자가 대문자이면서 전자
- 모든 알파벳 문자가 소문자이면서 반자(첫번째 문자가 대문자이고 반자인 경우는 제외)
- 모든 알파벳 문자가 소문자이면서 전자(첫번째 문자가 대문자이고 전자인 경우는 제외)

HEVE는, 대체될 문자열(기존 문자열)의 문자형과 문자 크기를 유지한 채로, 대체 문자열(새로운 문자열)을 현재 버퍼에 넣습니다. 그렇지 않으면, 대체 문자열의 문자형과 문자 크기가 new-string 매개변수에 제공한 것과 꼭같게 됩니다. SET FIND 명령의 설명도 함께 참조하십시오.

4.3 문자형 및 문자크기 구분 기능의 설정

탐색될 문자열을 소문자로만 찾아서 대체하려면, SET CASE EXACT 명령을 사용하십시오. 그 다음에, 탐색 문자열을 모두 소문자로 입력하면, HEVE가 대문자가 포함된 문자열은 건너 뛰고 소문자로만 탐색합니다. 예를 들어, SET FIND CASE NOEXACT가 설정되어 있으면, Title, TITLE, 및 title, ｔｉｔｌｅ, ＴＩＴＬＥ 및 Ｔｉｔｌｅ이 모두 탐색 문자열 title과 일치하는 것으로 탐색됩니다. 그러나, SET FIND CASE EXACT가 설정되어 있다면 title, ｔｉｔｌｅ만이 탐색 됩니다. 내정값은 SET FIND CASE NOEXACT입니다.

또한, HEVE가 SET FIND FORM [NO]EXACT 명령을 사용하여 탐색 문자열 내의 문자의 문자크기(전자 또는 반자)를 대조하는 방법을 결정할 수 있습니다. SET FIND FORM NOEXACT가 가동되어 있는 경우에는, HEVE가 문자열을 대조할 때에 전자 및 반자 ASCII 문자를 구분하지 않습니다. 예를 들어, Name과 Ｎａｍｅ을 구분하지 않는 것 등입니다.

문자형과 문자크기의 구분을 설정하는 명령으로는 SET FIND EXACT와 SET FIND GENERAL이 있습니다.
SET FIND CASE EXACT와 SET FIND FORM EXACT를 동시에 가동시키면, SET FIND EXACT 명령의 효과와 동일합니다. 이와 유사하게, SET FIND GENERAL은 SET FIND CASE NOEXACT와 SET FIND FORM NOEXACT를 동시에 가동시킨 것과 동일한 효과를 갖습니다.

내정 설정값은 SET FIND GENERAL입니다. 즉, SET FIND CASE NOEXACT와 SET FIND FORM NOEXACT를 동시에 가동시킨 것과 같습니다.

이들 설정은 FIND, REPLACE 및 WILDCARD FIND에 적용되고, HALF_FORM 및 FULL_FORM ASCII 문자에만 효과가 있습니다.

4.4 화이트 스페이스로 탭 설정

HEVE를 수정하여 FIND 및 WILDCARD FIND 명령이 공백, 탭 및 행 구분점을 "화이트 스페이스"로 처리하게 할 수 있습니다. 따라서, 2개 이상의 단어들이 몇 개의 공백이나 탭, 또는 행 구분점으로 분리되어 있어도 하나의 문자열로 탐색할 수 있습니다.

탐색시에 공백과 탭 모두를 "화이트 스페이스"로 처리되게 하려면, SET FIND WHITESPACE 명령을 사용하십시오. 내정값은 SET FIND NOWHITESPACE입니다. 예를 들어, 다음 명령은 "가나"가 공백이나 탭에 의해 분리되어 있거나, 또는 "가"는 행의 끝에 있고 "나"는 다음 행의 처음에 있어도 하나의 문자열로 탐색합니다.

```
Command: SET FIND WHITESPACE
Command: FIND 가나
```

4.5 와일드카드를 사용한 탐색

와일드카드는 원문의 패턴을 탐색하게 합니다. 예를 들어, 문자열

이 행의 처음에 나타날 때에만 탐색하게 하거나 또는 문자들이 모음 다음에 올 때에만 탐색하게 할 수 있습니다. 설정값에 따라 VMS 와일드카드를 사용하거나 또는 ULTRIX 와일드카드를 사용할 수 있습니다. 내정 설정값은 VMS입니다. 이 설정값은 SET WILDCARD ULTRIX 또는 SET WILDCARD VMS 명령을 사용하여 수정할 수 있습니다.

와일드카드를 사용하여 탐색하려면, WILDCARD FIND 명령을 사용하십시오. 그러면, HEVE가 찾을 원문의 패턴을 입력하도록 프롬프트합니다. 탐색 패턴에는 와일드 카드와 문자열을 모두 사용할 수 있습니다.

대조 패턴(matching pattern)을 찾으면, HEVE가 찾은 원문을 강조표시합니다.
그 다음에는, 범위를 찾았을 때와 같이, COPY, FILL, OPEN SELECTED, REMOVE, 및 UPPERCASE WORD 등의 명령을 사용하여 찾은 원문을 편집할 수 있습니다.

사용가능한 와일드카드의 패턴 목록을 보려면, SHOW WILDCARDS 명령을 사용하십시오. 다음의 예는, 대부분이 VMS 와일드카드에서 사용되는, 상이한 탐색 패턴을 보여 줍니다.

패턴	대조결과
가%나	"가다나라"에서와 같이, "가" 다음에 임의의 단일 문자와 "나"가 나타납니다. 백분율 기호(%)는 행에서 임의의 단일 영어 또는 한글 문자를 대조합니다. ULTRIX에서는, "가.나"와 같이 마침표가 등가 기능을 합니다.
가*나	"가다라나"에서와 같이 "가" 다음에 임의의 문자(들)과 "나"가 같은행에 나타납니다. 또는 문자열이 "가%나"에 의해 찾아집니다. 별표(*)는 동일한 행에서 임의개의 영어 또는 한글 문자를 대조합니다.
digital**디지탈	"digital디지탈"이 임의개의 원문 즉, 탭, 공백, 행 구분, 또는 기타 원문 문자에 의해 분리됩니다.
digital\>디지탈	"digital디지탈"이 행 구분에 의해 분리됩니다. 즉, "digital"은 한 행의 끝에 있고, "디지탈"은 다음 행의 시작 위치에 있게 됩니다. 역 슬래시와 우각괄호(\>)는 행의 끝을 대조합니다. 역슬래시와 좌각괄호(\<)는 행의 처음을 대조합니다. ULTRIX에서는, 행의 끝과 행의 처음을 대조하는데에 각각 카렛(^)과 달러 기호($)를 등가적으로 사용합니다.

제 5 장

선/상자 그리기 및 기호

HEVE는 선 그리기와 상자 그리기 기능을 제공합니다. DRAW LINE과 DRAW BOX 명령을 사용하면, 커서를 움직이는데로 선과 사각형을 손쉽게 그릴 수 있습니다. 또한 SYMBOL 명령을 사용하면, 화살표 기호와 같은 특수 기호를 커서 위치에 삽입할 수 있습니다.

5.1 선/상자 그리기

DRAW LINE 및 DRAW BOX 명령은 간단한 사용자 위주의 인터페이스를 제공합니다. 이 그리기 기능들은 몇개의 간단한 전환 키로 제어됩니다. 그리기 상태는 제어키가 눌려지는데에 따라 갱신됩니다. DRAW LINE 명령을 사용하면, HEVE는 먼저 선 그리기 모드로 들어갑니다. 그러면, 상태 행이 선 그리기 모드에 있다는 것을 나타내도록 변경됩니다.

[End of file]

Draw_Line: Move[Draw]->SELECT Bold[Thin]Erase->INSERT Draw Box->REMOVE

선 그리기 모드에서는, 상향, 하향, 좌향, 우향 화살표 키를 사용하여 커서를 이동시키면서 선을 그리거나 지울 수 있습니다. 미니 키패드의 Remove 키를 누르면, 상자 그리기 모드로 전환됩니다. 선 그리기 모드에서의 제어키는 다음과 같습니다.

표 **5-1** 선 그리기 모드에서의 제어 키

기능	전환 키
선 그리기/커서 이동	미니 키패드의 Select 키 또는 숫자 키패드에서는 KP1 키
굵은 선/가는 선 /지우기	미니 키패드의 Insert Here 키 또는 숫자 키패드에서는 KP2 키
상자 그리기로 전환	미니 키패드의 Remove 키 또는 숫자 키패드에서는 KP3 키
선 그리기 종료	Do 키 또는 Return 키

커서, 그리기 또는 이동의 모드를 전환하려면, 미니 키패드의 Select 키(또는 KP1 키)를 누르십시오. 선 그리그 문자의 유형, 즉 굵은 선/가는 선, 또는 선 지우기를 선택하려면 미니 키패드의 Insert Here 키(또는 KP2 키)를 누르십시오. 선 그리기 모드를 종료하려면, Do 키 또는 Return 키를 누르십시오.

DRAW BOX 명령을 사용하면, HEVE가 상자 그리기 모드로 들어갑니다. 상태 행은 상자 그리기 모드를 나타내도록 변경됩니다.

선 그리기 모드에 있을 때에, 미니 키패드의 Remove 키를 누르면 상자 그리기 모드로 들어갑니다. 그리고 상태 행이 갱신됩니다.



[End of file]

Draw_Box: Exit Draw Box->SELECT Cancel Draw Box->REMOVE

상자 그리기 모드에서는, 상향, 하향, 좌향, 우향 화살표 키를 사용하여 커서를 이동시키면서 상자를 그릴 수 있습니다. 상자 그리기 모드에서의 제어 키는 다음과 같습니다.

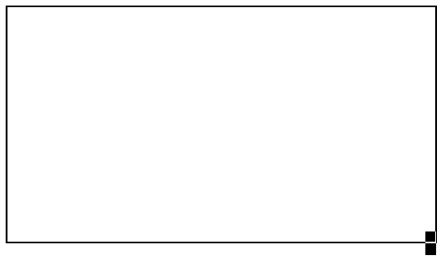
표 **5-2** 상자 그리기 모드에서의 제어 키

기능	전환 키
상자 그리기 종료	미니 키패드의 Select 키 또는 숫자 키패드에서는 KP1 키
상자 그리기 취소	미니 키패드의 Remove 키 또는 숫자 키패드에서는 KP3 키
상자 및 선 그리기 모두 종료	Do 키 또는 Return 키

상자 그리기를 종료하고, 선 그리기 모드로 복귀하려면, Select 키 또는 KP1 키를 누르십시오. 상자 그리기를 취소하고 선 그리기 모드로 복귀하려면, Remove 키 또는 KP3 키를 누르십시오. 상자 그리기와 선 그리기 모드를 모두 종료하려면, Do 키 또는 Return 키를 누르십시오. 선 그리기 문자는 항상 화면의 홀수번째 열에 정렬됩니다. 각각의 선 그리기 문자는 화면에서 2개의 열을 차지합니다.

아래의 예는, HEVE에서의 선 그리기와 상자 그리기 기능을 사용하여 2개의 열로된 표를 그리는 것을 보여 줍니다.

먼저, 커서를 표의 왼쪽 상단 모서리로 이동시킨 후에 DRAW LINE 명령을 사용합니다. 다음에, Insert Here 키 또는 KP2 키를 눌러서 선의 굵기를 선택합니다. 이때, 가는 선을 선택했다고 가정하겠습니다. 그 다음에, Remove 키 또는 KP3 키를 눌러서 상자 그리기 모드로 변경한 후에 커서를 표의 오른쪽 하단 모서리로 이동시킵니다.

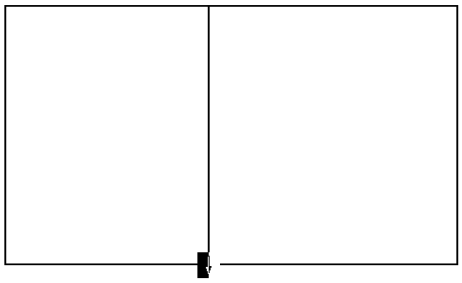


[End of file]

Draw_Box: Exit Draw Box->SELECT Cancel Draw Box->REMOVE

그리고 나서, Select 키 또는 KP1 키를 눌러서 선 그리기 모드로 복귀합니다. 선 그리기 모드에서는, Select 키를 누르면 커서가 이동 모드로 변경되어, 커서가 표의 상단으로 이동하여 원하는 열 위치에 정지합니다.

그 다음, 커서를 그리기 모드로 변경한 다음에 커서를 표의 하단으로 내립니다.



[End of file]

Draw_Line: Move[Draw]->SELECT [Bold]ThinErase->INSERT Draw Box->REMOVE

Insert Here 키를 눌러서 선을 지우도록 선 유형을 전환하여, 선 그리기를 취소할 수 있습니다. 선 지우기는 커서가 선을 따라 이동할 때 선 그리기 문자를 지웁니다.

선의 교차점은 자동적으로 병합됩니다.

그런 다음에, 표의 표제를 넣기위해 수평 선을 그립니다.



[End of file]

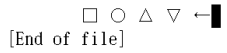
Draw_Line: Move[Draw]->SELECT [Bold]ThinErase->INSERT Draw Box->REMOVE

이제, Do 키 또는 Return 키를 눌러서, 선 그리기 모드를 종료하고 표를 채울 수 있습니다. 표를 채울 때에는 먼저 버퍼를 겹쳐쓰기 모드로 변경합니다. 그렇지 않으면, 표가 삽입된 원문에 의하여 훼손됩니다.

5.2 기호

SYMBOL 명령을 사용하면, HEVE가 사용가능한 기호 세트와 색인을 (ASCII 순서로) 화면의 아래 부분에 표시합니다. 다음 페이지에 그 예를 보였습니다.

그림 **5-1** **SYMBOL** 명령 입력 후의 화면



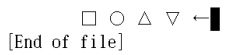


그 다음에, 아래의 예와 같이 프롬프트에서 원하는 기호의 색인을 입력하여서, 기호를 선택할 수 있습니다.

```
Symbols: [] {} <\ </ <-
```

아래에 색인 입력 후 기호가 화면에 표시되는 결과를 나타내었습니다. 이 명령은 특수 기호를 사용하여 도형을 그릴 때 유용합니다.

그림 **5-2** 화면에 기호가 표시된 예





제 6 장

버퍼와 창을 사용한 2개 이상의 파일 편집

버퍼는 편집 세션 중에 편집한 내용을 일시적으로 저장하는 저장 영역입니다. HEVE는 버퍼를 작성하고, 다루고, 삭제하는 명령을 제공합니다. 버퍼에 포함되어 있는 정보는 버퍼가 창과 연계되어 있어야만 화면에서 볼 수 있습니다.

창은 편집한 버퍼를 화면표시하거나 또는 명령이나 메시지와 같은 기타 정보를 화면표시하기 위하여 HEVE가 사용하는 화면 영역입니다. 그리고 창은 화면 영역에 들어갈 수 있는 만큼의 원문을 화면표시하므로, 버퍼에 들어 있는 정보의 일부분만을 화면에 표시합니다. HEVE는 창을 분리하고, 다루고, 삭제하는 명령을 제공합니다.

편집 세션 중에는, 2개 이상의 파일을 편집하기 위하여 여러개의 버퍼를 사용할 수 있으며, 원한다면 원문 블록을 다루기 위한 일시적인 저장 장소로 사용할 수도 있습니다. HEVE는 창에 표시되는 버퍼를 전환시킬 수 있는 명령을 제공합니다. 원문을 어느 한 파일에서 다른 파일로 복사하려 할 때에는 특히, 다중 버퍼가 유용합니다.

그리고, 다중 창을 사용하면 동시에 다른 파일들 또는 동일한 파일의 다른 부분들을 편집하거나 볼 수 있습니다. 이것은 서로 다른 버전의 초고를 비교하거나, 또는 파일들간에서 오려붙이기 하거나, 또는 다른 부분을 편집할 때에 긴 문서의 한 부분을 참조할 때에 유용합니다.

6.1 버퍼의 사용

6.1.1 버퍼의 목록 보기

편집 세션 중에 작성한 모든 버퍼의 목록을 화면에 표시하려면, SHOW BUFFERS 명령을 사용하십시오. 이때, 목록을 영상이동하면서 보려면 Next Screen 키를 누르거나 또는 미니 키패드의 Prev Screen 키를 누르십시오. 현재 창 내의 버퍼를 화면표시하려면, 커서를 버퍼명 위치로 이동시킨 후에 Select 키를 누르십시오. 버퍼를 삭제하려면, 커서를 버퍼명 위치로 이동시킨 후에 Remove 키를 누르십시오.

HEVE가 작성한 모든 버퍼, 즉 시스템 버퍼의 모든 목록을 화면에 표시하려면, SHOW SYSTEM BUFFERS 명령을 사용하십시오. 이때, 영상이동을 하여서 목록을 볼 수도 있고, 아니면 커서를 버퍼명 위치로 이동시킨 후 Select 키를 눌러서 원하는 버퍼를 지정하여 해당 버퍼의 목록만을 볼 수도 있습니다. 그러면, HEVE가 버퍼를 현재 창으로 집어넣습니다.

주의
시스템 버퍼는, 일부 명령 작업 수행에 꼭 필요하므로, 삭제하지 마십시오.

6.1.2 버퍼를 현재 창에 집어 넣기

BUFFER 명령 다음에 버퍼명을 지정하여서, 지정된 버퍼를 현재 창

에 집어 넣을 수 있습니다. 버퍼명을 지정해 주지 않으면, HEVE가 프롬프트합니다. 버퍼명을 지정할 때에는 약어를 사용할 수 있습니다.

버퍼가 이미 있을 때에는, HEVE가 커서를 해당 버퍼의 최종 위치로 복귀시킵니다. 버퍼가 없을 때에는, HEVE가 새로운 버퍼를 작성하여 커서를 그 버퍼의 상단에 위치시킵니다.

또한, BUFFER 명령을 사용하여 MESSAGES 버퍼, HELP 버퍼, DCL 버퍼, 또는 INSERT HERE 버퍼와 같은 HEVE의 시스템 버퍼들을 볼 수도 있습니다. 예를 들어, HEVE는 화면의 하단에 나타나는 MESSAGES 창을 사용하여 편집 세션 중에 오류 메시지와 정보 메시지를 표시합니다. MESSAGES 창은 MESSAGES 버퍼 내의 최종 메시지를 화면에 표시합니다. BUFFER MESSAGES 명령을 사용하면, MESSAGES 버퍼의 내용을 화면에 표시할 수 있습니다. 이 명령은 DECTPU 프로시저를 컴파일할 때에 유용합니다.

6.1.3 파일을 버퍼로 읽어 들이기

파일을 HEVE의 버퍼로 읽어 들이는 데에는 다음의 4가지 방법이 있습니다.

- 파일 명세를 지정하여 HEVE를 호출하는 방법.

- INCLUDE FILE 명령 다음에, 현재 버퍼에 포함시킬 파일의 이름을 지정하는 방법. 이 경우, HEVE는 지정된 파일의 전체 내용을 현재 버퍼 내의 커서가 위치한 행 바로 앞에 읽어 들입니다. INCLUDE FILE 명령은 상태 행의 버퍼명을 변경하지 않습

니다.

- GET FILE 또는 OPEN 명령 다음에 사용할 파일명을 지정하는 방법. 이 경우, HEVE는 새로운 버퍼를 작성하고, 지정된 파일(있는 경우)의 내용을 버퍼로 읽어 들입니다. 상태 행의 버퍼 명은 GET FILE 또는 OPEN 명령에서 지정해 준 파일명과 동일합니다.

- GET WILDCARDED FILES 명령 다음에 파일 명세를 지정하는 방법. 와일드카드를 사용할 수 있습니다. 이 명령은, 책의 여러 장이나 프로그램의 여러 모듈들을 위하여 여러 개의 파일을 편집하는 것과 같이, 단일 명령을 사용하여 2개 이상의 파일을 여는 것을 제외하고는, GET FILE 명령과 유사합니다.

6.1.4 버퍼 내용을 파일에 쓰기

현재 버퍼의 내용을 파일에 쓰려면, WRITE FILE 명령을 사용하십시오. WRITE FILE 명령에 파일 명세를 포함시킬 수 있습니다. 파일 명세를 지정해 주지 않으면, HEVE가 입력 파일 명세를 사용하여 파일을 작성합니다. BUFFER 명령을 사용하여 현재 버퍼를 작성한 경우에는, HEVE가 쓰려고 하는 대상 파일의 명세를 입력하도록 프롬프트합니다.

HEVE의 시스템 버퍼에 쓰려고 하면, HEVE가 출력 파일명을 물어옵니다. 예를 들어, 정의된 키의 목록 사본을 보관하기 위하여 HELP 버퍼의 내용을 파일에 쓰려면, 다음과 같이 명령을 사용하십시오.

```
Command: HELP KEYS
도움말을 종료하려면 RETURN 키를 누르십시오.
Command: BUFFER HELP
Command: WRITE FILE KEYDEFS.LIS
```

6.1.5 버퍼 삭제

버퍼를 삭제하려면 DELETE BUFFER 명령 다음에 버퍼명을 지정하여 사용하십시오. 버퍼명을 지정해 주지 않으면, HEVE가 프롬프트 합니다. 버퍼명은 정확하게 일치하여야 합니다. 그리고, 약어나 와일드카드를 사용할 수 없습니다.

다른 방법으로, SHOW BUFFERS 명령을 사용하여 작성한 버퍼의 목록을 화면표시한 다음에 버퍼명을 입력하지 않고, 커서를 삭제할 버퍼명에 위치시키고 Remove 키를 눌러서 삭제할 수도 있습니다. 이 방법은, 버퍼명이 매우 긴 경우에 유용합니다. 삭제될 버퍼가 비었거나 수정되지 않은 경우에는, HEVE가 그것을 삭제합니다. 그렇지 않으면, HEVE가 작업 내용을 실수로 파괴하는 것을 방지하기 위하여 다음 중의 한가지를 선택하도록 묻습니다.

응답	효과
Delete_only	지정된 버퍼를 삭제합니다.
Write_first	버퍼 내용을 먼저 저장하고(쓰고), 삭제합니다.
Quit	작업을 취소합니다. 버퍼는 삭제되지 않습니다. 이것이 내정 선택값입니다. 단순히 Return 키만 눌러도 효과는 동일합니다.

6.2 창의 사용

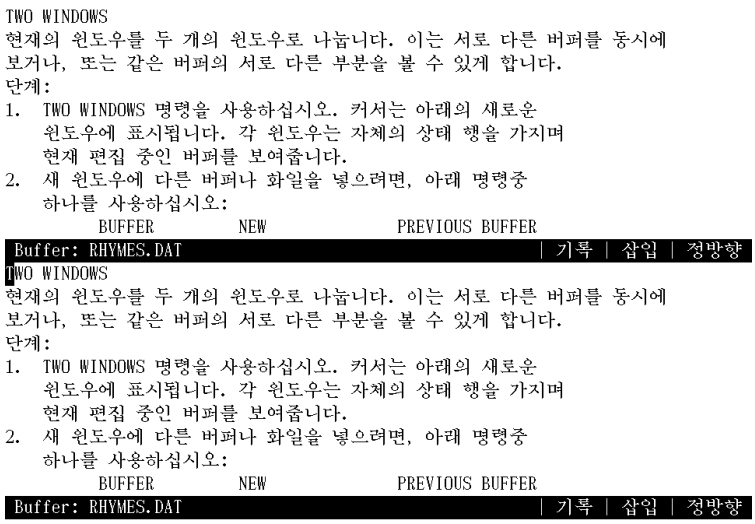
6.2.1 다중 파일 보기

동일한 세션 내에서 2개 이상의 파일을 편집중에 있는 경우에는, HEVE의 주 창을 여러 개의 보다 작은 창들로 분리하여서 서로 다른 파일들을 동시에 볼 수 있습니다. 또한, 매우 긴 문서를 편집하는 것과 같은 경우에도 주 창을 분리하여 동일한 파일의 서로 다른 부분들을 볼 수 있습니다.

창을 분리하면, HEVE가 항상 커서를 맨 아래 창에 위치시킵니다. 각각의 창은 자신의 상태 행을 가지며, 현재 편집중인 버퍼를 화면 표시합니다.

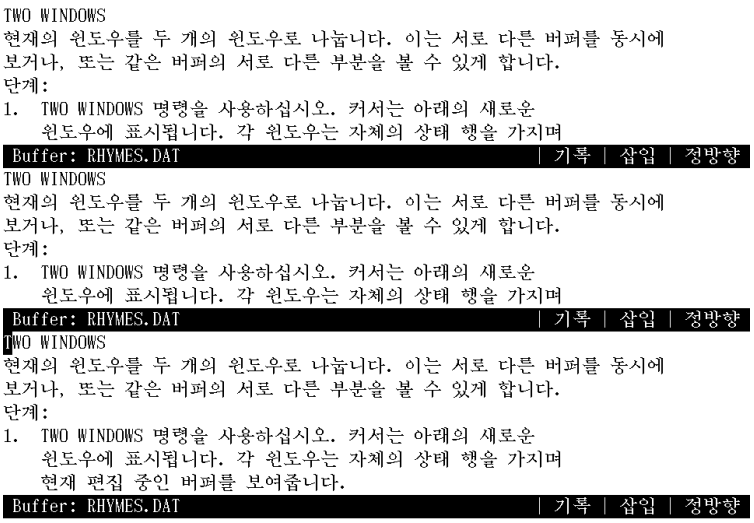
주 창을 2개의 창으로 분리하려면, TWO WINDOWS 명령을 사용하십시오. 예를 들어, 현재 버퍼가 RHYMES.DAT인 경우에, TWO WINDOWS 명령을 사용하면 화면이 다음 페이지와 같이 나타납니다.

그림 **6-1 TWO WINDOWS** 명령 사용 후의 화면



창을 3개 이상의 동일한 크기의 창들로 분리하려면, SPLIT WINDOW 명령 다음에 원하는 창의 수를 지정하십시오. 숫자를 지정하지 않으면, SPLIT WONDOW 명령이 TWO WINDOWS 명령과 같은 효과를 냅니다. HEVE에서 허용되는 최대 창의 수는 사용하는 단말기의 크기와 유형에 따라 다릅니다. 따라서, 21-행짜리 주 창의 경우(VT220 단말기에서)에는 11개까지의 창을 가질 수 있습니다(이때, 각 창에는 1행의 원문을 표시하는 상태행을 각각 1개씩 갖습니다). 예를 들어, 처음에 버퍼 RHYMES.DAT의 내용을 화면표시하는 창을 한 개 갖고 창을 3개의 창으로 분리하기 위하여 SPLIT WINDOWS 3 명령을 사용하면 화면 구성이 다음과 같이 됩니다.

그림 **6-2 SPLIT WINDOW 3** 명령 사용 후의 화면



그 다음에, 다음 명령들 중의 하나를 사용하여 한 개의 다른 파일 또는 버퍼를 현재 창에 넣을 수 있습니다.

```
BUFFER
NEW
NEXT BUFFER 또는 PREVIOUS BUFFER
OPEN 또는 GET FILE
OPEN SELECTED
SHOW BUFFERS 또는 SHOW SYSTEM BUFFERS
```

창들 간에서 이동할 때에는 NEXT WINDOW 또는 PREVIOUS WINDOW 명령을 사용하십시오. 그러면, 커서가 다음 창 또는 이전 창에서 마지막으로 있었던 위치로 돌아갑니다.

서로 다른 창들에 동일한 버퍼 또는 파일이 나타난다 하더라도, 각

창에서의 위치는 다를 수 있습니다. 이것은, 긴 문서를 편집할 때와 같이, 동일한 버퍼 또는 파일의 서로 다른 부분을 보거나 편집할 때에 유용합니다.

6.2.2 창 삭제

현재 창, 즉 커서가 현재 위치해 있는 창을 삭제하려면, DELETE WINDOW 명령을 사용하십시오. 예를 들어, 그림 6-1에서와 같이 2개의 창을 가지고 있을 때 DELETE WINDOW 명령을 사용하여 현재 창을 삭제하면, 다른 창으로 사용가능한 공간을 전부 채운 후에 커서(▯)를 남아 있는 창의 마지막 위치에 나타냅니다.

그림 **6-3 DELETE WINDOW** 명령 사용 후의 화면

```
TWO WINDOWS
현재의 윈도우를 두 개의 윈도우로 나눕니다. 이는 서로 다른 버퍼를 동시에
보거나, 또는 같은 버퍼의 서로 다른 부분을 볼 수 있게 합니다.
단계:
1.  TWO WINDOWS 명령을 사용하십시오. 커서는 아래의 새로운
    윈도우에 표시됩니다. 각 윈도우는 자체의 상태 행을 가지며
    현재 편집 중인 버퍼를 보여줍니다.
2.  새 윈도우에 다른 버퍼나 화일을 넣으려면, 아래 명령중
    하나를 사용하십시오:
        BUFFER          NEW             PREVIOUS BUFFER
        GET FILE        OPEN            SHOW BUFFER
        NEXT BUFFER     OPEN SELECTED   SHOW SYSTEM BUFFERS

[End of file]

Buffer: RHYMES.DAT                                  | 기록 | 삽입 | 정방향

13 lines read from file DATA$:[USER.CVMS.HANGUL]RHYMES.DAT;1
```

창을 삭제하여도, 해당 창 내의 버퍼는 삭제되지 않는다는 점을 유의하십시오. 버퍼를 삭제하려면, DELETE BUFFER 명령을 사용하십시오.

현재 창은 놓아 두고 다른 모든 창들을 삭제하려면, ONE WINDOW 명령을 사용하십시오.

6.3 다중 창을 사용한 다중 버퍼 편집

편집 세션중에, 2개 이상의 파일을 편집하기 위해 여러개의 버퍼를 사용할 수 있습니다. 이 작업 도중에는 다중창을 사용하여 동시에 여러개의 버퍼를 보거나 편집할 수 있습니다.

다음의 예는, 두 개의 창을 사용하여 두 개의 파일을 편집하면서, 원문을 한 파일에서 다른 파일로 이동하는 방법을 보여줍니다. 이 예에서는, 먼저 HEVE를 호출하여 파일 RHYMES.DAT를 편집합니다.

그림 **6-4 RHYMES.DAT**의 내용을 보여주는 화면

```
TWO WINDOWS
현재의 윈도우를 두 개의 윈도우로 나눕니다. 이는 서로 다른 버퍼를 동시에
보거나, 또는 같은 버퍼의 서로 다른 부분을 볼 수 있게 합니다.
단계:
1.  TWO WINDOWS 명령을 사용하십시오. 커서는 아래의 새로운
    윈도우에 표시됩니다. 각 윈도우는 자체의 상태 행을 가지며
    현재 편집 중인 버퍼를 보여줍니다.
2.  새 윈도우에 다른 버퍼나 화일을 넣으려면, 아래 명령중
    하나를 사용하십시오:
        BUFFER          NEW             PREVIOUS BUFFER
        GET FILE        OPEN            SHOW BUFFER
        NEXT BUFFER     OPEN SELECTED   SHOW SYSTEM BUFFERS

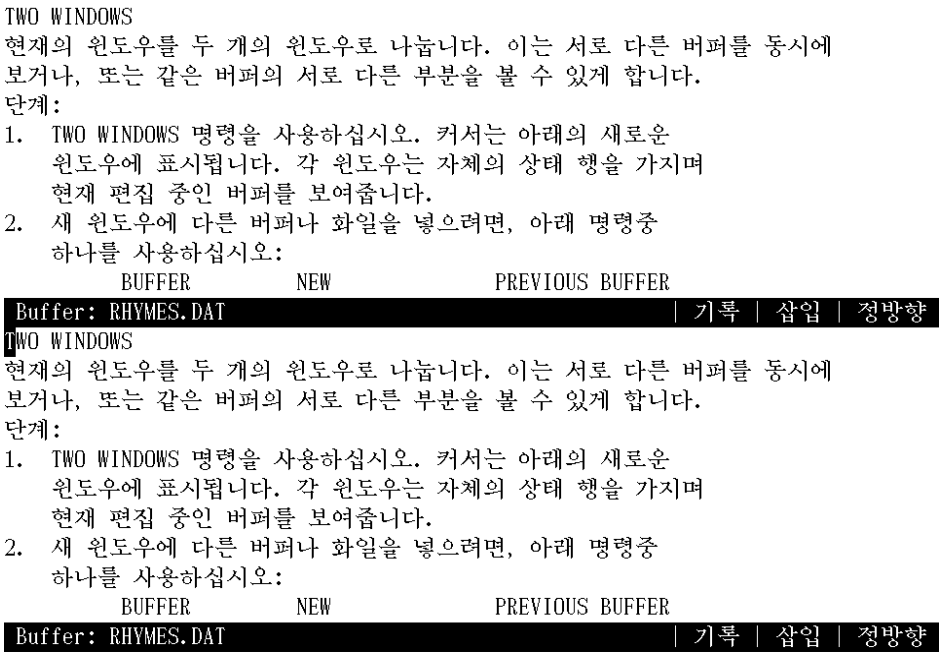
[End of file]

Buffer: RHYMES.DAT                                  | 기록 | 삽입 | 정방향

13 lines read from file DATA$:[USER.CVMS.HANGUL]RHYMES.DAT;1
```

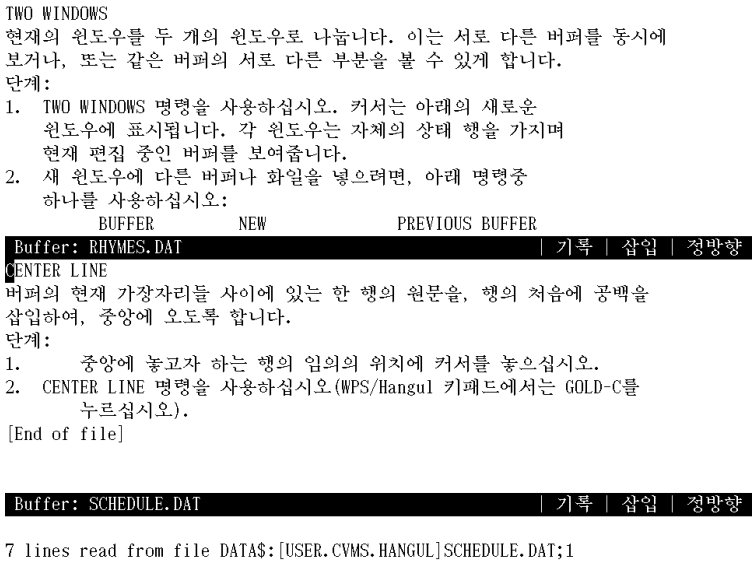
TWO WINDOWS 명령을 사용하여 화면 상에 두 개의 창을 생성합니다.

그림 **6-5 TWO WINDOWS** 명령을 사용한 후의 화면



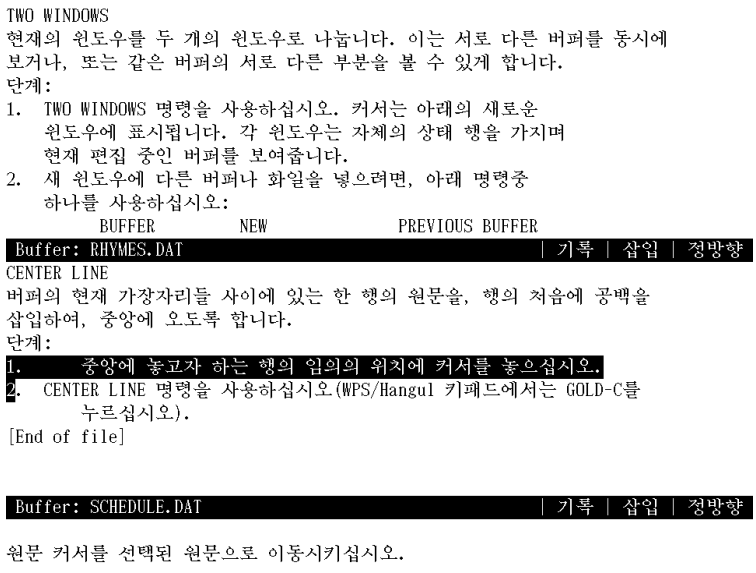
그 다음에, GET FILE SCHEDULE.DAT 명령을 사용하여 새로운 버퍼를 생성하고, 화면의 하단 창에 SCHEDULE.DAT의 내용을 표시합니다.

그림 **6-6 GET FILE SCHEDULE.DAT** 명령을 사용한 후의 화면



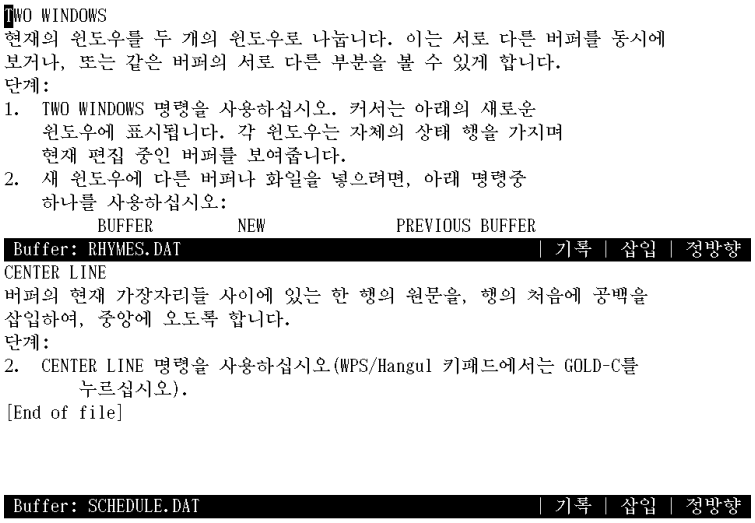
현재 하단 창에서, 커서를 4번째 행에 있는 숫자 "1"에 이동합니다. 다음에 Select 키를 누른 후, 하향 화살표 키를 한번 눌러서 SCHEDULE.DAT의 4번재 행을 선택합니다.

그림 **6-7 SCHEDULE.DAT**의 **4**번째 행을 선택한 후의 화면



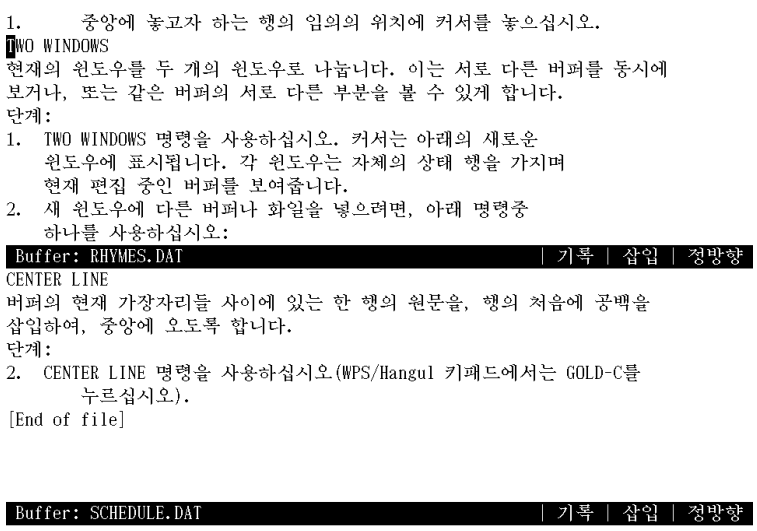
Remove 키를 눌러서 선택된 원문을 INSERT HERE 버퍼로 집어 넣습니다. 다음에, NEXT WINDOW 명령을 사용하여 커서를 다른 창으로 이동합니다.

그림 **6-8** 선택한 원문을 제거한 후의 화면



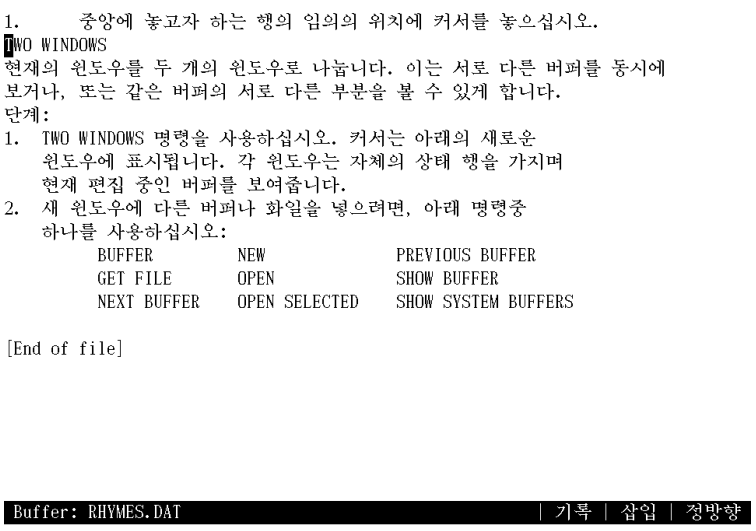
커서를 RHYMES.DAT 버퍼의 상단으로 위치시킨 후, SCHEDULE.DAT 파일에서 제거한 원문을 Insert Here 키를 눌러서 RHYMES.DAT 버퍼로 집어넣습니다.

그림 **6-9 RHYMES.DAT** 버퍼에 원문을 삽입한 후의 화면



ONE WINDOW 명령을 사용하여 현재 창을 제외하고 다른 모든 창을 제거합니다.

그림 **6-10 ONE WINDOW** 명령을 사용한 후의 화면



편집 세션을 종료하면, HEVE가 현재 버퍼의 내용을 파일에 기입한 후, 다른 버퍼의 내용을 파일에 기입할 것인지를 묻습니다.

제 7 장

HEVE에서의 키 정의

사용자는 HEVE 명령을 실행하는 키를 정의하거나 또는 빈번히 사용하는 일련의 키조작(런 순서라고 부릅니다)을 정의할 수 있습니다. 키를 정의하거나 또는 정의를 해제할 때에는 명령어 행에 키 명을 입력해야 합니다.

HEVE의 키 명은 일반적으로 키의 레이블과 동일합니다. 다음 페이지의 표 7-1에 HEVE의 키 명과 키보드 또는 키패드 상의 키 레이블을 수록해 놓았습니다. 일부 키들은 일부 단말기에서 나타나지 않을 수도 있습니다.(예를 들어, VT100-계열 단말기들은 F1- F20 키들을 가지고 있지 않습니다.
VT200 및 VT300 계열 단말기들은 BS 키와 LF 키를 가지고 있지 않습니다)

표 **7-1 HEVE** 키 명

키	키 명 [1]
F7...F20	F7...F20
Help	HELP 또는 F15
Do	DO 또는 F16
Find	FIND 또는 E1
Insert Here	INSERT_HERE 또는 E2
Remove	REMOVE 또는 E3
Select	SELECT 또는 E4
Prev Screen	PREV_SCREEN 또는 E5
Next Screen	NEXT_SCREEN 또는 E6
↑	UP
←	LEFT
↓	DOWN
→	RIGHT

[1]HEVE에서 정의된 키 명은 HTPU 내장 프로시저의 키 명과 다를 수 있습니다. 이들 키 명을 HTPU 내장 프로시저에서 사용하지 마십시오. HTPU 내장 프로시저에서 사용하는 정확한 키 명에 대해서는 DEC Text Processing Utility Reference Manual에서 "Keywords Used for Key Names" 표를 참조하십시오.

표 7-1 (계속) **HEVE** 키 명

키	키 명
PF1...PF4	PF1...PF4
0...9(숫자 키패드)	KP0...KP9
-(숫자 키패드의)	MINUS
.(숫자 키패드의)	PERIOD
,(숫자 키패드의)	COMMA
<x 또는 Delete	DELETE
Tab 또는 TAB	TAB 또는 Ctrl/l
Backspace	BS 또는 Ctrl/H
Linefeed	LF 또는 Ctrl/J

키 명은 혼동되지 않는 범위 내에서 약어를 사용할 수 있습니다. 예를 들어, G Rem은 GOLD[1]-Remove에 대하여 유효한 약어 키 명이 됩니다. 그리고, G R은 GOLD-R의 약어가 될 수 있습니다. 키 정의에서, 키 명은 대/소문자를 구분하지 않습니다.

GOLD 키는 다른 키(한글 문자 키 포함)와 함께 사용하도록 정의한 키로서, 가능한 키 정의의 수를 두 배로 늘려 줍니다. 예를 들어, F20 키에 하나의 명령 실행과 GOLD 키 순차를 정의해 놓으면, 이 F20 키는 다른 명령을 실행하는 것이 됩니다. GOLD 키의 설정에 대한 보다 상세한 사항은 7.3절을 참조하십시오.

[1]GOLD 키 정의는 7.3절을 참조하십시오.

제어 키는 Ctrl, Control, 또는 곡절 액센트 문자(^)를 사용하여 지정합니다. 예를 들어, Ctrl/A, Control/A 및 ^A는 모두 동일합니다.

DECwindows Motif/Hangul 인터페이스에서는, Shift 기능 키 및 Alt 조합 키를 정의할 수 있습니다. 이동(Shift) 기능 키 조합은 주 키보드 상의 Shift 키를 누른 상태에서 기능 키(F14, Remove, PF4, x, 또는(MB2[1]와 같은) 마우스 단추)를 누르는 것입니다. DECwindows Motif/Hangul 인터페이스에서는 키보드 상의 Compose 키가 Alt 키로 작용합니다.

Alt 키는 기능 키, 이동 가능 키, 제어 키, 마우스 단추, 또는 GOLD 순차 키와 조합하여 사용할 수 있습니다.
한글 입력 모드를 시작되게 하려면, Compose/Space[2] 순차키를 사용하십시오.

제어 키, 이동 가능 키, Alt 조합 키, 또는 GOLD 순차 키를 지정할 때에는, 예를 들어 Ctrl/N, GOLD-B, SHIFT_E3와 같이, 키 명칭 내에 구분 문자로서 대시(-), 사선(/), 밑줄(_)을 사용합니다.
HEVE는 규약으로서, (SHOW KEY 또는 HELP KEYS 명령에서) 키 명을 화면에 표시할 때에, 제어 키와 이동 기능 키, 그리고 Alt 조합 키에는 사선을 사용하고, GOLD 순차 키에는 대시를 사용합니다. 따라서, Ctrl에서와 같이 하나의 키를 누른 상태에서 다른 키

1 마우스의 두번째 단추를 말합니다.
2 LK401 키보드에서는, Compose 키 만을 사용합니다.

를 눌러야 하는 동시적인 키 조합을 나타낼 때에는 사선으로 표시하고, GOLD-Help에서와 같이 하나의 키를 누른 다음에 다른 키를 눌러야 하는 순서적인 키 조합을 나타낼 때에는 대시를 사용합니다.

7.1 HEVE의 명령 실행을 위한 키 정의

HEVE에서는, HEVE의 명령에 대해 단일 키 또는 제어 순차 키를 할당할 수 있습니다. 자주 사용하는 HEVE 명령을 입력하기 위한 편집 키를 만들 수도 있습니다.
키를 정의하려면, DEFINE KEY 명령을 사용한 다음에 Return 키를 누릅니다. 그러면, HEVE가 키에 연계시킬 명령을 입력하도록 프롬프트를 화면에 표시합니다. 이 프롬프트에서 명령을 입력하십시오. 그러면, HEVE가 정의할 키를 누르도록 다시 프롬프트를 화면에 표시합니다. 이러한 방법 외에, 아래의 예와 같이 명령 행에서 키 명을 입력할 수도 있습니다.

명령	효과
`DEFINE KEY=ctrl/f include file`	CTRL/F를 INCLUDE FILE로 정의합니다.
`DEFINE KEY=gold-remove hedt kp9`	GOLD-REMOVE(미니 키패드의)를, SET KEYPAD HEDT를 사용하지 않고 HEDT Append 키 KP9 로 정의합니다. 이와 유사하게, WPS Ruler 키에 대해 GOLD-R을 정의하는 것과 같이, WPS의 키패드 기능을 할당할 수 있습니다.

DEFINE KEY는 HEVE의 내정 키, HEDT 키, WPS 키, DECwindows Motif/Hangul 인터페이스 키, 또는 사용자 자신의 키 여부를 불문하고, 키의 현재 정의를 무시합니다.
키 정의는 편집 세션동안 또는 그 정의를 변경하기 전까지 유효합니다.

키 정의는 편집 세션을 종료하면 상실됩니다. 키 정의를 향후의 편집 세션에서도 유지되게 하려면, 다음 중의 한가지 방법을 사용해야 합니다.

- SAVE EXTENDED HEVE 명령을 사용하여, 키 정의를 한 편집 세션에서 다음 편집 세션으로 저장하는 방법

- DEFINE KEY 명령을 초기화 파일에 넣어 놓는 방법

사용자 자신의 편집 환경을 설정하는 것에 대한 보다 상세한 사항은 제8장을 참조하십시오.

7.2 Learn 순서를 입력하기 위한 키 정의

HEVE에서는 단일 키 또는 제어 키에, 런 순서(learn sequence)라고 하는, 일련의 키조작을 할당할 수 있습니다. 런 순서를 사용하면, 단순히 한 개의 키만을 눌러서, 동일한 키 조작을 버퍼에 몇번이고 입력할 수 있습니다. 이는 일련의 편집 작업이나 또는 자주 사용되는 원문의 입력을 하는데 유용합니다.

런 순서를 작성하려면, LEARN 명령을 사용하십시오. 그 다음에,

HEVE가 런(learn)할 편집 사항을 작성하십시오. 이에는 이미 정의되어 있는 키를 누르는 것을 포함하여, 명령이나 원문, 또는 둘 다를 입력할 수 있습니다. 그러나, 메뉴를 사용하는 것과 같은 마우스 작업은 런 순서에 포함시킬 수 없습니다. 런 순서를 종료하거나 또는 "기억"시키려면, REMEMBER로 정의되어 있는 CTRL/R을 누르십시오(REMEMBER 명령을 입력하지 마십시오. 그렇지 않으면, 그 명령 자신이 런 순서에 포함됩니다). 그 다음에는 HEVE가 정의할 키를 누르도록 프롬프트합니다. 기능 키, 제어 키, GOLD 순차 키는, 그것들이 작성 중인 런 순서에서 사용되지 않은 경우에는, 해당 키를 누를 수 있습니다. 런 순서를 포기하려면, 프롬프트에서 Return 키를 누르십시오. 그러면 재정의되지 않습니다.

런 순서를 불러 낼 때에는, 해당되는 한 개의 키만을 누르면 됩니다. 그러면, HEVE가 일련의 키 조작을 하나씩 하나씩 수행합니다. 런 순서는, 해당 순서가 작성된 시점에서의 키 정의보다는, 키 조작을 기록합니다. 따라서, 런 순서를 불러 내기 전에, 그 순서 내에서 사용되는 모든 키의 정의나 또는 그 순서에 적용되는 모든 설정을 점검하십시오.

키 정의는 편집 세션 동안, 또는 그 키를 정의 해제하거나 재정의할 때까지 효과가 지속됩니다. 키 정의를 향후의 세션에까지 적용되게 저장하려면, SAVE EXTENDED HEVE 명령을 사용하여 섹션 파일을 작성하십시오. 런 순서는 초기화 파일에 넣을 수 없습니다.

7.3 GOLD 키의 정의

GOLD 키를 생성하면, 동일한 편집 키에 두 개의 정의를 할당할 수 있습니다. 한가지 편집 기능은 편집 키를 누르면 수행됩니다. 다른 기능은, 먼저 GOLD 키를 누른 다음에 편집 키를 누르면 수행됩니다. GOLD 키를 정의하려면, SET GOLD KEY 명령을 사용하십시오. 명령 행에서 키 명을 입력하거나 또는 GOLD 키로 사용할 키를 누르도록 HEVE가 프롬프트하게 할 수도 있습니다. 예를 들어, 다음의 명령은 PF1 키를 GOLD 키로 설정합니다.

```
Command: SET GOLD KEY PF1
```

일단, GOLD 키를 설정한 후에야, HEVE가 사용자를 위하여 사전 정의한 GOLD 순차 키를 사용할 수 있습니다. 이들 GOLD 키는 흔히 "잠자는" 키로도 불립니다. 왜냐하면, 이들 키가 내정에 의해 정의된 것이기는 하지만 GOLD 키를 설정해 주어야 "깨어나기" 때문입니다.

GOLD 순차 키는 기능 키, 제어 키, 또는 타이핑 키(한글 문자 포함)를 사용하여 정의할 수도 있습니다. 예를 들어, 다음의 명령은 CENTER 명령에 대해 GOLD 순차 키와 한글 문자 "가"를 정의합니다.

```
Command: DEFINE KEY=GOLD-가 CENTER
```

키 정의는 편집 세션 동안, 또는 그 키를 정의 해제하거나 재정의

할 때까지 효과가 지속됩니다. GOLD 키 정의를 향후의 세션에까지 저장하려면, SAVE EXTENDED HEVE 명령을 사용하여 섹션 파일을 작성하십시오. 또는, SET GOLD KEY 명령을 초기화 파일에 넣어 놓으십시오.

7.4 Motif-스타일 기능 키의 정의

표 7-2에 SET FUNCTION KEYS MOTIF 명령을 사용하여 기동할 수 있는 Motif DECwindows-스타일 키 정의를 수록해 놓았습니다.

표 **7-2 Motif-**스타일 키 정의

키	Motif-스타일 정의
Ctrl/space	SELECT
Ctrl//	SELECT ALL
Ctrl/\	RESET
Ctrl/up	이전 단락
Ctrl/down	다음 단락
Alt/right	END OF LINE
Ctrl/right	HEDT/WPS 다음 단어
Ctrl/left	HEDT/WPS 이전 단어
Alt/Ctrl/right	BOTTOM

표 **7-2** (계속) **Motif-**스타일 키 정의

키	Motif-스타일 정의
Alt/left	START OF LINE
Alt/Ctrl/left	TOP
Alt/Shift/E2	RESTORE
Alt/E3	1차 삭제
Ctrl/E3	HEDT 행의 끝까지 삭제
Shift/E3	STORE TEXT
Alt/Shift/E3	1차 복사
Ctrl/Shift/E4	1차 선택 복구(재강조표시)
Ctrl/E5	Page left(1창 폭만큼 왼쪽으로 이동)
Ctrl/E6	Page right(1창 폭만큼 오른쪽으로 이동)
Shift/delete	삭제 대기 또는 HEDT 커서 오른쪽까지의 문자 삭제

이들 명령은 이들 정의가 HEVE 내정, HEDT 키패드, WPS-Plus 키패드, 또는 DEFINE KEY나 LEARN 명령에 의한 정의에 관계 없이, 키에 대한 어떠한 정의도 무시하지 않습니다. Motif 기능 키를 가동시키기 전에 작성한 런 순서는, 런 순서 내의 해당 키가 현재 정의(또는 재정의)되어 있으면, 적절하게 동작하지 않습니다.

이동 기능 키는 DECwindows Motif/Hangul 인터페이스 상에서는 동작하지만, VT220 또는 VT100과 같은 문자 셀 단말기 인터페이스에서는 동작하지 않습니다.

키를 가동 중지하려면, SET FUNCTION KEYS NOMOTIF 명령을 사용하십시오.

7.5 마우스 단추의 정의

수정이 불가능한 MB1[1] 단추와 Shift-MB1 단추를 제외한 모든 마우스 단추를 정의할 수 있습니다. 표 7-3은 정의되지 않은 MB1 단추를 눌렀을 때의 결과를 보여줍니다. 미정의 마우스 단추의 예를 들면 GOLD-MB1이 있습니다.

표 **7-3 MB1**에 대한 마우스 단추 정의

동작	결과
1번 클릭	커서를 지적한 위치로 이동하고, 모든 선택 범위 또는 탐색 범위를 취소합니다.
2번 클릭	커서를 지적한 위치로 이동하고, 해당 위치의 모든 단어를 선택합니다.
3번 클릭	커서를 지적한 위치로 이동하고, 해당 위치의 모든 행을 선택합니다.
4번 클릭	커서를 지적한 위치로 이동하고, 해당 위치의 모든 단락을 선택합니다.
5번 클릭	커서를 지적한 위치로 이동하고, 모든 버퍼를 선택합니다.
끌기	MB1을 누른 곳에서부터 MB1에서 손가락을 뗀 곳까지의 원문을 선택합니다.
시프트/끌기	선택 범위를 연장하거나 축소합니다.

[1]마우스의 첫번째 단추입니다.

마우스에서의 다운스트로크, 업스트로크, 클릭 및 끌기를 각각 별도의 키로 정의할 수 있습니다. 정의한 마우스 단추를 누르면, HEVE가 먼저 커서 위치 이동 작업을 수행하고(지적한 위치로 커서를 이동한 다음에, 선택 범위 또는 탐색 범위를 취소), 그 다음에 해당 마우스 단추에 연계되어 있는 명령을 실행합니다. 예를 들어, 다음 명령에서 GOLD-MB3[1]의 다운스트로크(누르기)는 PASTE 명령, 그리고 업스트로크(떼기)는 FILL 명령을 정의합니다.

```
Command: DEFINE KEY=GOLD-M3DOWN PASTE
Command: DEFINE KEY=GOLD-M3UP FILL
```

따라서, GOLD-MB3를 눌렀다가 떼면, HEVE는 먼저 지적한 위치로 커서를 이동시킨 다음에 마지막으로 삭제하거나 복사한 원문을 삽입한 후, 커서 위치의 단락을 채웁니다.

7.6 키패드의 설정

HEVE의 다른 편집기 또는 단말기의 키패드를 에뮬레이트하기 위하여 숫자 키패드와 일부 다른 키들을 정의하려면 SET KEYPAD 명령을 사용하십시오. 예를 들어, VT100-계열 단말기에서 HEVE를 사용하는 것에 익숙해 있는 사용자라면, SET KEYPAD VT100 명령을 사용하여 키패드를 설정하십시오.

HEDT 키패드를 설정하려면, SET KEYPAD HEDT 명령을 사용하십시오. 이것은 내정 설정값입니다. 숫자 키패드와 일부 다른 키들은

[1]마우스의 세번째 단추입니다.

HEDT를 에뮬레이트 하기 위하여 정의됩니다. 예를 들어, PF1은 GOLD 키로, PF4는 행 삭제, GOLD-PF4는 RESTOR LINE, GOLD-↑는 LEFT ADJUST, 그리고 GOLD-↓는 RIGHT ADJUST로 정의됩니다.

WPS 키패드를 설정하려면, SET KEYPAD WPS 명령을 사용하십시오. 이것은 WPS를 에뮬레이트하기 위한 숫자 키패드와 일부 다른 키들을 정의합니다. 예를 들어, PF1은 GOLD 키, 그리고 GOLD-R은 WPS Ruler 키로 정의됩니다. 또한, WPS 키패드를 자동적으로 설정되게 설정해 놓으면, 커서가 연계됩니다. WPS 키패드 기능은 클립보드에서는 사용할 수 없습니다.

그리고 HEVE 키패드를 선택하기 위한 HEVE$KEYPAD 논리명을 정의할 수도 있습니다. 보다 상세한 사항은 8.9절을 참조하십시오.
HEDT 또는 WPS에 의한 정의와 다르게 정의하면, 그 정의 내용이 HEDT 또는 WPS 정의를 무시하고 적용됩니다. 이후에, 이들 키 정의를 해제해 주면, HEDT 또는 WPS 정의가 재작용됩니다. HEDT 또는 WPS 키패드를 설정한다 해도 HEDT 또는 WPS의 모든 명령이나 기능을 완전히 구현하지는 않습니다. 예를 들어, HEDT 행 모드 명령은 입력할 수 없습니다.

7.7 키 정의 해제

UNDEFINE KEY 명령을 사용하여, DEFINE KEY 또는 LEARN 명령을 사용하여 정의된, 키의 현재 정의를 취소할 수 있습니다. 이를 위해서는, 명령 행에서 키 명을 입력하거나 또는 HEVE가 정의를 해제할 키를 누르도록 프롬프트하게 하여야 합니다.

이때, 지정된 키가 이미 SET KEYPAD 명령으로 정의되어 있는 경우에, 해당 키패드 설정이 효력을 발생하고 있으면 이전 정의가 복구됩니다. 예를 들어, 다음 명령은 HEDT 정의를 무시하고 HEDT 키패드를 설정한 다음에 KP9 키를 CENTER LINE으로 재정의합니다. 이후에, UNDEFINE KEY 명령을 사용하면 해당 정의를 취소하고, 그 키의 HEDT 정의(Append)를 복구합니다.

```
Commnad: SET KEYPAD HEDT
Command: DEFINE KEY=KP9 CENTER LINE
Command: UNDEFINE KEY KP9
```

7.8 정의할 수 없는 키들

HEVE에서는 사용자가 정의할 수 없는 키들이 일부 있습니다. 이들 키는 다음과 같습니다.

F1-F6
Compose 키
Return 또는 Ctrl-M
Break
Escape 또는 Ctrl-[
Lock 또는 Caps Lock
No Scroll
Set-Up
Shift

이 외에, HEVE에서는 다음 키들을 사용자가 정의할 수 있습니다.

1 타이핑 키(제어 키 또는 GOLD 키와 같이, 수정자와 함께 조합 키로 사용할 때는 제외)

2 DO 키로 정의된 키(이 키가 유일하게 DO 키로 정의되어 있는 키인 경우만 해당)

3 현재 GOLD 키로 설정되어 있는 키

제 8 장

HEVE의 설정

HEVE를 사용할 때에는, 특정한 편집 어플리케이션을 위하여 사용자 자신의 편집 환경을 설정해야 할 경우도 있습니다. 예를 들어, 서식문이나 다른 종류의 문서 편집을 위하여, HEVE의 내정 설정값을 무시하고 왼쪽 및 오른쪽 여백이나 단락 들여쓰기를 달리 설정해야 할 경우 등입니다.

HEVE의 모든 명령들이 한 개의 키에 연계되어 있는 것이 아니기 때문에, 사용자가 원하는 키에 HEVE 명령을 실행하도록 정의하여, 향후의 편집 세션에서 정의된 해당 키를 사용하는 것은 매우 유용합니다.

보다 숙련된 사용자라면, HTPU 프로그래밍 언어를 사용하여 원문 처리 어플리케이션을 작성할 수도 있습니다.

이러한 모든 것들이 HEVE 설정에 포함됩니다. HEVE의 설정은 다른 종류의 편집 속성을 설정하거나 또는 HEVE 명령을 실행하는 키를 정의하거나, 또는 HTPU 프로시저를 사용하여 편집기를 확장시켜서 할 수 있습니다.

일부 편집 속성들은 글로벌 설정 특성을 가지므로, 편집기 또는 모든 버퍼에 적용됩니다. 다른 속성들은 버퍼에만 적용되는 설정 특성을 가지게 되어, 특정 버퍼에만 적용됩니다. 예를 들어, 커서 이동 방식(고정 또는 자유) 유형과 탭 모드(삽입, 공백, 또는 이동) 유형은 편집하는 대상 버퍼 모두에 대하여 동일하게 적용되는 반면

에, 여백이나 단락 들여쓰기와 탭 멈춤을 각 버퍼에 대하여 다르게 설정할 수 있습니다.(예를 들어, 한 버퍼에는 오른쪽 여백을 75로 하고, 다른 버퍼에는 오른쪽 여백을 68로 설정할 수 있습니다)

키 정의, 런 순서, 글로벌 및 버퍼 특유의 설정, 또는 HTPU 프로시저는 기동 파일에 저장할 수 있습니다. 기동 파일에는 다음의 세 가지 유형이 있습니다.

- 섹션 파일
- 명령 파일
- 초기화 파일

섹션 파일은 키 정의와 다른 확장사항(컴파일된 HTPU 프로시저 등)을 저장하는 2진 파일입니다. 섹션 파일은 2진 형태의 파일이기 때문에, 편집 환경을 신속하게 설정할 수 있는 반면에, 사용자는 2진 파일을 화면표시하거나 편집할 수 없습니다. 섹션 파일의 내정 파일 유형은 .TPU$SECTION입니다. 사용자가 작성한 섹션 파일은, 사용자가 설정한 HEVE 버전인 것이 효과적입니다.

명령 파일은 HTPU 프로시저와 명령문을 포함하고 있는 파일로서, 일반적으로 사용자 자신의 명령을 HEVE에 추가하거나 또는 원문 처리 어플리케이션을 작성하는 것과 같이, 편집기를 설정하거나 확장하기 위한 것입니다. 명령 파일은 섹션 파일보다 늦게 실행됩니다. 명령 파일의 내정 파일 유형은 .TPU입니다.

초기화 파일은 HEVE 명령들을 포함하고 있는 파일로서, 일반적으로 각 편집 세션 마다 모든 명령들을 매번 입력하지 않고, 여백이나 탭 멈춤 또는 기타 필요한 사항들을 설정하기 위한 것입니다.

HEVE를 호출할 때 초기화 파일을 사용하거나 또는 편집 세션 중에 초기화 파일을 실행할 수 있습니다. 초기화 파일은 섹션 파일이나 명령 파일보다 늦게 편집 환경을 설정합니다. 초기화 파일의 내정 파일 유형은 .EVE입니다.

8.1 HEVE 내정 설정값

표 8-1은 HEVE 내정 설정값의 목록입니다. 이 설정값들은 사용자가 다른 값을 설정해 주기 전에는 HEVE가 그 값을 사용합니다. 이 표는 기동 파일을 작성할 때에 변경할 설정값을 점검하는데 활용하면 좋습니다.

표 **8-1** 모든 버퍼에 대한 **HEVE** 내정 글로벌 설정값

내정 설정값	효과
SET BOX NOSELECT	상자-형식의 선택, 삭제, 및 삽입을 가동중지시켜서, 표준 방식의 행 단위의 범위 선택을 할 수 있게 합니다.
SET BOX PAD	버퍼의 모드에 관계없이, 상자 편집을 위하여 채우기와 겹쳐쓰기를 가동시킵니다.
SET DISPLAY CURSOR	커서 화면 표시 모드를 가동시켜서, 위 아래로 이동시 커서를 문자 경계에 정렬되도록 합니다

표 **8-1** (계속) 모든 버퍼에 대한 **HEVE** 내정 글로벌 설정값

내정 설정값	효과
SET CURSOR FREE	버퍼의 미사용 부분으로 이동할 수 없는 커서 고정 모드와는 달리, 커서를 버퍼 내의 어느 위치로나 이동시킬 수 있고, 또 원문을 어느 곳으로나 입력할 수 있습니다. SET KEYPAD WPS 명령을 사용하면 자동적으로 커서 고정 모드가 가동됩니다.
SET EXIT ATTRIBUTE CHECK	속성을 변경한 후에 종료하면, HEVE가 그것을 저장할 것인지를 묻게 합니다.
SET FIND CASE NOEXACT	모두 소문자로 된 탐색 문자열을 입력하면, 대소문자를 구분하지 않고, HEVE가 원문 내의 모든 탐색 대상 문자열을 찾게 합니다.
SET FIND FORM NOEXACT	문자 크기를 구분하지 않고, HEVE가 모든 탐색 대상 원문 문자열을 찾게 합니다.
SET FIND NOWHITESPACE	FIND와 WILDCARD FIND 명령이, 입력한 것과 꼭같이, 원문 문자열 내의 공백과 탭을 대조합니다. 이때, 행 구분을 넘어가는 것은 탐색하지 않습니다.
SET FUNCTION KEYS NODECWINDOWS	DECwindows 기능을 위하여 일부 키를 정의하기보다는, 정상적인 키 정의(HEVE 내정, HEDT 키 패드, 또는 WPS 키패드)를 유지합니다.

표 **8-1** (계속) 모든 버퍼에 대한 **HEVE** 내정 글로벌 설정값

내정 설정값	효과
SET KEYPAD HEDT 또는 SET KEYPAD VT100	VT400-계열, VT300-계열, 및 VT200-계열 단말기에서는 숫자 키패드 상의 키들이 HEDT에 대하여 내정 키로 동작합니다. VT100-계열 단말기에서는, 숫자 키패드가 HEVE 내정 키 묶음으로 사용됩니다. 제어 키들은 어느 유형의 단말기에서나 동일하게 정의됩니다. 또한, 어느 유형의 단말기에서나 WPS 키패드를 설정할 수 있습니다.
SET NOCLIPBOARD	HEVE에서 복사, 삭제 및 삽입 동작시 INSERT HERE 버퍼를 사용합니다. DECwindows에서는, 클립보드를 가동시켜서 HEVE와 다른 DECwindows 어플리케이션 사이에서 원문을 전송할 수 있습니다. WPS 키패드 키는, 설정값에 관계 없이, 클립보드를 사용하지 않습니다.
SET NODEFAULT COMMAND FILE	HEVE가 다음 파일 중의 하나를 속성 저장을 위한 내정 명령 파일로 사용합니다. o HEVE 호출시 /COMMAND 한정자로 지정한 명령 파일 o 현 디렉토리에서 파일명이 HTPU$COMMAND.TPU인 명령 파일 o 논리명 HTPU$COMMAND로 정의되어 있는 명령 파일

표 **8-1** (계속) 모든 버퍼에 대한 **HEVE** 내정 글로벌 설정값

내정 설정값	효과
SET NODEFAULT SECTION FILE	파일 속성 저장시에 사용되는 파일로서, 내정되어 있지 않은 섹션 파일. 이 섹션 파일은 사용자가 지정해 주어야 합니다.
SET NOPENDING DELETE	선택한 것을 지우지 않고, 삭제하거나 또는 새로운 원문을 입력할 수 있게 합니다.
SET SECTION FILE PROMPTING	속성이나 또는 다른 설정 사항을 저장할 때에, HEVE가 섹션 파일을 물어보는 프롬프트를 표시하게 합니다.
SET SCROLL MARGINS 0 0	창의 상단(즉, 창의 상단에서 아래로 0번째 행) 또는 하단(즉, 창의 하단에서 위로 0번째 행)을 지날 때, 자동적으로 영상 이동이 시작되게 합니다.
SET TABS INSERT	TAB 명령을 사용하여 탭 문자를 삽입합니다. 탭 문자 대신에 공백을 삽입하거나 또는 아무것도 삽입하지 않고 커서만 이동하도록 탭 모드를 설정할 수 있습니다.
SET TABS INVISIBLE	화면에 표시되는 탭과는 달리, 편집 중에 탭 문자가 빈 공백으로 표시되게 합니다. 이 탭은 작은 H_T (수평 탭) 다음에 점(.)으로 표시됩니다.

표 **8-1** (계속) 모든 버퍼에 대한 **HEVE** 내정 글로벌 설정값

내정 설정값	효과
SET WIDTH 80	HEVE의 화면의 폭을 일반적인 80열짜리의 단말기와 같게 설정합니다.
SET WILDCARDS VMS	행 위의 임의의 원문을 별표(*)를 사용하여 대조하거나 또는 행 위의 단일 문자를 백분율 기호(%)를 사용하여 대조하는 것과 같이, WILDCARD FIND 명령이 VMS-스타일의 와일드카드를 사용하게 합니다. SET WILDCARDS ULTRIX 명령을 사용하여 ULTRIX-스타일의 와일드카드를 사용할 수도 있습니다

표 8-2는 버퍼 특유의 설정값에 대한 HEVE의 내정 설정값 목록입니다.

표 **8-2 HEVE** 내정 버퍼 특유의 설정값

내정 설정값	효과
FORWARD	FIND와 MOVE BY LINE과 같은 명령이 커서를 오른쪽/아래로 이동합니다. 이 방향을 역방향(왼쪽/위)으로 변경할 수도있습니다.
INSERT MODE	입력하는 문자가, 기존의 원문을 오른쪽/아래로 밀어내면서, 현재 커서 위치에 삽입되게 합니다. 이 모드를 겹쳐쓰기 모드로 변경할 수 있습니다.
SET JOURNALING	모든 원문 버퍼에 대하여 버퍼-변경 저널링을 가능하게 합니다.

표 **8-2**(계속) **HEVE** 내정 버퍼 특유의 설정값

내정 설정값	효과
SET BUFFER MODIFIABLE	작성한 버퍼를 수정(편집)할 수 있게 합니다. 이 버퍼를 수정할 수 없게 설정할 수도 있습니다.
SET BUFFER WRITE	이 설정값은 두 가지 효과를 가집니다. 하나는 현재 버퍼를 수정할수 있도록 설정하는 것이고, 다른 하나는 기존 버퍼 내용을 변경한 경우에 HEVE가 그 변경 내용을 버퍼에 기입(저장)하게 합니다. 버퍼를 판독-전용(기입금지)으로 설정할 수도 있습니다.
SET LEFT MARGIN 1	이 값이 최좌측 열입니다. Return 키를 누르거나, 또는 FILL 명령을 사용하거나, 또는 HEVE가 원문을 줄 바꿔쓰는 경우에, 새로운 행이 버퍼의 왼쪽 여백에서 시작합니다.
SET PARAGRAPH INDENT 0	사용자가 작성하거나 또는 FILL 명령으로 재형식화 한 단락이 들여쓰기되지 않고 버퍼의 현재 왼쪽 여백에서 시작되게 합니다
SET RIGHT MARGIN 79	내정 오른쪽 여백값은 단말기에서 설정한 79값보다 1열 만큼 작습니다. 이 폭이 80열이면, 내정 오른쪽 여백은 79입니다. FILL 명령을 사용하거나 또는 행의 끝에서 입력하면, HEVE가 버퍼의 오른쪽 여백에서 원문을 줄 바꿔쓰기합니다.
SET TABS EVERY 8	탭 멈춤을 제 9, 17, 25, 33, 41열 등으로 설정합니다. 이 탭 멈춤값을 다른 간격으로 설정할 수도 있습니다.
SET WRAP	행의 끝에서 원문을 입력하면, Return 키를 누르거나 또는 FILL 명령을 사용하지 않아도, HEVE가 버퍼의 오른쪽 여백에서 원문을 줄바꿔쓰기합니다.

일부의 HEVE 글로벌 설정값들은 섹션 파일이나, 또는 HEVE가 HTPU 명령 파일에 생성한 코드에 저장할 수 있습니다. 여백 및 탭 멈춤과 같은 다른 설정값들은 초기화 파일에 저장되도록 설정할 수 있습니다.

8.2 속성의 저장

속성들은 글로벌(전역적) 설정 특성을 갖습니다. 일부의 글로벌 설정값들만이 향후의 편집 세션을 위하여 섹션 파일이나 또는 HTPU 명령 파일에 저장될 수 있도록 정의됩니다. 표 8-3은 섹션 파일 또는 명령 파일에 저장할 수 있는 설정값들을 보여줍니다.

표 **8-3** 속성 설정을 위한 **HEVE** 명령

명령	내정 설정값
SET BOX [NO]SELECT	SET BOX NOSELECT
SET BOX [NO]PAD	SET BOX PAD
SET CURSOR { BOUND / FREE }	SET CURSOR FREE
SET [NO]DISPLAY CURSOR	SET DISPLAY CURSOR
SET [NO]EXIT ATTRIBUTE CHECK	SET EXIT ATTRIBUTE CHECK
SET FIND CASE [NO]EXACT	SET FIND CASE NOEXACT
SET FIND FORM [NO]EXACT	SET FIND FORM NOEXACT
SET [NO]CLIPBOARD	SET NOCLIPBOARD

표8-3 (계속) 속성 설정을 위한 **HEVE** 명령

명령	내정 설정값
SET [NO]DEFAULT COMMAND FILE	SET NODEFAULT COMMAND FILE
SET [NO]DEFAULT SECTION FILE	SET NODEFAULT SECTION FILE
SET [NO]PENDING DELETE	SET NOPENDING DELETE
SET [NO]SECTION FILE PROMPTING	SET SECTION FILE PROMPTING
SET TABS {INSERT|MOVEMENT |SPACES}	SET TABS INSERT
SET TABS [IN]VISIBLE	SET TABS INVISIBLE

이들 설정값을 위한 명령을 포함하고 있는 HEVE 초기화 파일이 있는 경우에는, 섹션 파일 또는 명령 파일에 설정값을 저장한 후에 그 명령 행들을 삭제할 수 있습니다.

표 8-1의 다른 글로벌 설정값(영상이동 여백 또는 와일드카드의 유형 등)과 모든 버퍼 설정값(여백 또는 탭 멈춤 등)들은 섹션 파일에 저장될 수 없습니다. 일반적으로, 이들 설정값을 저장하기 위해서는 초기화 파일을 사용합니다.

편집 세션 중에 SAVE ATTRIBUTES 또는 SAVE EXTENDED HEVE 명령을 사용하거나 또는 편집 세션을 종료하므로써, 속성을 저장할 수 있습니다. 내정에 의해, 속성을 변경한 후에 저장하지 않고 종료하면, HEVE가 다음의 예에서와 같이 프롬프트합니다.

```
Command: SET CURSOR BOUND
Command: SET FIND CASE EXACT
Command: SET TABS VISIBLE
             .
             .
             .
Command: EXIT
Attributes were changed. Save them? [YES]
```

변경 내용을 저장하려면, 단순히 Return 키만 누르면 됩니다. 그러면, HEVE가 종료하기 전에 SAVE ATTRIBUTES 명령을 수행합니다. 이때, 변경 내용을 저장하지 않으려면, N을 입력하면 됩니다. 그러면, HEVE가 바로 종료합니다.

이 프롬프트를 발신하지 않게 하려면, 즉 보다 빨리 또는 간단하게 종료시키려면 SET NOEXIT ATTRIBUTE CHECK 명령을 사용하십시오. 그러나, 이 명령은 현재 편집 세션에는 적용되지 않습니다. 왜냐하면, 종료 점검은 그 자체가 글로벌 설정 특성을 가지므로 섹션 파일이나 명령 파일에 저장되기 때문입니다. 일단 저장하면, 그 설정값들은 사용자가 사용하는 관련된 섹션 파일 또는 명령 파일의 향후 편집 세션에 적용됩니다.

저장할 때에 섹션 파일 대신에 명령 파일에 저장시키려면, 다음의 명령을 사용하여, 속성을 저장하거나 또는 종료할 때에(속성을 변경시킨 경우), HEVE가 명령 파일에 저장할 것인지를 묻는 프롬프트 행에 내정 명령 파일의 파일명(HTPU$COMMAND.TPU)을 보여줍니다.

```
Command: SET NODEFAULT SECTION FILE
Command: SET NOSECTION FILE PROMPTING
Command: SET DEFAULT COMMAND FILE HTPU$COMMAND.TPU
```

표 **8-4** 속성 저장 명령의 요약

명령	사용법 또는 효과
SAVE ATTRIBUTES	HEVE 프롬프트에 대한 응답 내용 또는 다른 HEVE 명령을 사용한 설정 내용에 따라, 속성을 섹션 파일 또는 명령 파일에 저장합니다. 섹션 파일에 저장하면, 그 효과는 SAVE EXTENDED HEVE와 같습니다. 명령 파일에 저장한 경우에는, HEVE가 속성 설정 및 메뉴 정의를 위하여 특별히 표시된 HTPU 명령문 블록을 생성하고, 명령 파일을 작성하거나 아니면 이 기존 명령 파일을 명령문 블록으로 갱신합니다.
SAVE SYSTEM ATTRIBUTES	HEVE 내정 속성을 섹션 파일 또는 명령 파일에 저장합니다. 이것은 섹션 파일이나 명령 파일을 내정 HEVE 설정값과 메뉴 정의에 복구할 경우에 유용합니다.
SAVE EXTENDED HEVE	섹션 파일을 작성하고, 속성, 키 정의, 메뉴 정의, 컴파일된 프로시저 및 HTPU 명령문을 사용하여 설정한 글로벌 변수 등과 같은 기타 확장 사항들을 저장합니다. 명령 행에서 섹션 파일을 지정해 주지 않으면, HEVE가 지정해 주도록 프롬프트하거나 또는 내정 섹션 파일을 사용합니다.(내정 섹션 파일을 설정해 놓은 경우)

8.3 HTPU 프로시저를 사용한 HEVE의 확장

HTPU 언어로 코드를 작성하여 HEVE를 확장하거나 수정할 수 있습니다. DECTPU 프로그래밍 언어에 근거한 HTPU 프로그래밍 언어는, HEVE 편집기와 같은 원문 처리 어플리케이션 작성을 위하여 설계된 구조적 언어입니다. HEVE에서의 한자 원문 처리 기능은 완전한 세트의 HTPU 프로시저나 또는 수정된 DECTPU 내장 프로시저를 통하여 수행됩니다. 새로운 명령을 HEVE에 추가하거나, 또는 편집기의 설정, 또는 사용자 자신의 어플리케이션 작성을 위한 프로시저를 작성할 수 있습니다.

DECTPU 프로그래밍 언어와 내장 프로시저에 관한 상세한 사항은 "Guide to the DEC Text Processing Utility" 및 "DEC Text Processing Utility Reference Manual"을 참조하십시오. HTPU 특유의 내장 프로시저에 관한 정보는 "HTPU와 HEVE 참조서"을 참조하십시오.

새로운 HEVE 명령 프로시저를 작성할 때에는, 다음 사항에 유의하십시오.

- 프로시저 명에 접두어 EVE_를 붙여서, HEVE가 해당 프로시저를 HEVE 명령으로 인식하게 하십시오. 그 프로시저를 컴파일한 후에는, Do 키를 누른 다음에 EVE_ 접두어 또는 밑줄 없이 프로시저명만 입력하여 실행시킬 수 있습니다. 예를 들어, EVE_SET_LEFT_MARGIN 프로시저의 경우에는 SET LEFT MARGIN과 같이 입력하면 됩니다. 새로운 명령을 실행하기 위한

키를 정의할 수도 있습니다.

- 단어 PROCEDURE와 ENDPROCEDURE는 제1열에서 시작하여야 합니다.

부록 A에 커서가 화면 경계에 도달하였을 때에 화면을 자동으로 이동시키는 HTPU 프로시저의 예를 수록해 놓았습니다.

8.4 편집 세션 중의 **HTPU** 프로시저의 컴파일링

편집 세션 중에 HTPU 프로시저를 컴파일하려면, 컴파일하려는 프로시저 상의 어느 위치에나 커서를 위치시킨 후, EXTEND THIS 명령을 사용하십시오. 현재 버퍼에 여러개의 HTPU 프로시저가 있고, 그것들 모두를 컴파일하려면, EXTEND ALL 명령을 사용하십시오.

EXTEND HEVE 또는 EXTEND HTPU 명령을 사용할 수도 있으며, 또한 현재 버퍼 내의 지정된 HTPU 프로시저를 컴파일하기 위한 프로시저 명을 지정할 수도 있습니다.

컴파일한 HTPU 프로시저나 또는 내장 프로시저를 실행하려면, HTPU 명령을 사용하십시오. 예를 들어, 다음 명령은 HTPU 내장 프로시저를 실행하여 반자 ASCII 문자로 된 입력 문자열("DIGITAL")을 전자로 변환하여(CHANGE_SIZE), 그 변환된 문자열을 현재 커서 위치로 집어 넣습니다(COPY_TEXT).

```
Command: HTPU COPY_TEXT (CHANGE_SIZE ("DIGITAL", FULL_FORM))
```

향후의 편집 세션을 위하여 컴파일된 프로시저를 저장하려면, SAVE EXTENDED HEVE 명령을 사용하여 섹션 파일을 작성하십시오.

8.5 HEVE 초기화 파일의 사용

초기화 파일은, 각 편집 세션마다 매번 반복해서 명령을 입력하지 않고도, 일반적인 여백이나 탭 멈춤, 및 기타 사항들을 설정하는데 필요한 HEVE 명령들을 포함하고 있는 파일입니다. 초기화 파일은 편집 세션 중에 HEVE를 호출할 때에 사용할 수 있습니다.

다음은 편집에 필요한 사항을 설정하고, 키들을 정의하는 명령을 포함하고 있는 예제 HEVE 초기화 파일입니다.

```
!   MYINIT.EVE initialization file
!
SET LEFT MARGIN 5
SET PARAGRAPH INDENT 4
SET RIGHT MARGIN 70
SET TABS EVERY 10
SET SCROLL MARGINS 9% 9%
SET FIND WHITESPACE
!   Key definitions
DEFINE KEY= F20      SHOW BUFFERS
DEFINE KEY= Ctrl/P   PAGINATE
DEFINE KEY= GOLD-L   ADJUST LEFT
DEFINE KEY= GOLD-R   ADJUST RIGHT
DEFINE KEY= KP7      WPS GOLD-R
```

편집 세션을 시작하면, HEVE는 먼저 현재(내정) 디렉토리를 탐색한 다음에, 파일 HEVE$INI.EXE에 대한 논리명 SYS$LOGIN을 탐색합니다. HEVE는 이 논리명을 탐색할 때에, 첫번째의 초기화 파일을 사용합니다(있는 경우). 다른 초기화 파일을 사용하려면, 논리명 HEVE$INIT를 정의하거나 또는 /INITIALIZATION 한정자를 사용하십시오. 예를 들어, 다음 명령은 MYEVE.EVE 초기화 파일을 사용하여 HEVE를 현재(내정) 디렉토리로 호출합니다.

```
$ EDIT/HTPU/INITIALIZATION=MYEVE
```

초기화 파일을 모든 세션 또는 대부분의 세션을 위하여 사용하려고 하거나, 특히 초기화 파일을 편집 파일이 있는 디렉토리와 다른 디렉토리에 넣어 놓으려면, 논리명 HEVE$INIT를 정의하십시오. 이러한 정의는 사용자의 LOGIN.COM 파일에 넣어 놓을 수 있습니다.

위의 명령 예에서, 상기의 두가지 이상의 방법으로 초기화 파일들을 지정한 경우에는, /INITIALIZATION 한정자 다음에 논리명 HEVE$INIT, 그리고 그 다음에 현재 또는 로그인 디렉토리내의 초기화 파일명을 지정해 주어야 합니다.

극소수의 키 만을 정의하는 경우에는, DEFINE KEY 명령을 초기화 파일에 넣어 놓을 수 있습니다. 그러나 다수의 키들을 정의하는 경우에는, 키 정의를 섹션 파일에 저장하여야 합니다. 섹션 파일은 2진 형태로 되어 있으므로, 기동 시간이 초기화 파일보다 빠릅니다

기동시에, HEVE는 명령 파일을 실행(있는 경우)하고, 섹션 파일을 로드한 후에 초기화 파일을 실행합니다. 따라서, 초기화 파일 내의 설정 사항과 키 정의들이 섹션 파일과 명령 파일 내의 것들을 무시

합니다. 여백, 탭 멈춤, 기타 버퍼-전용 설정값들을 위한 초기화 파일 내의 명령들이 주(또는 최초) 버퍼와 HEVE 시스템 버퍼 $DEFAULTS$에 적용됩니다. $DEFAULTS$ 버퍼는 템플릿 버퍼입니다. 사용자가 버퍼를 작성하면 -예를 들어, GET FILE 명령을 사용하여- HEVE는 $DEFAULTS$ 버퍼의 설정값들을 사용하므로, 각각의 새로운 버퍼들도 동일한 설정값을 갖습니다.

따라서, 초기화 파일이 SET RIGHT MARGIN 70 명령을 포함하고 있으면, 사용자가 작성하는 각 버퍼도 그와 동일한 오른쪽 여백값을 갖게 됩니다. 내정 설정값을 알아보려면, SHOW DEFAULTS BUFFER 명령을 사용하십시오. 편집 중인 버퍼의 설정값을 알아보려면, SHOW 명령을 사용하십시오.

편집 세션 중에 초기화 파일을 실행하려면, @ 명령 다음에 실행할 초기화 파일명을 지정하여 사용하십시오. 예를 들어, 다음 명령을 사용하여 MYEVE.EVE 초기화 파일을 실행할 수 있습니다.

```
Command: @MYEVE
```

세션 중에 초기화 파일을 실행시키면, 버퍼 설정 명령이 현재 버퍼에만 적용됩니다. 이것은 초기화 파일이 포함하고 있는 명령을 입력하는 것과 동일한 효과를 냅니다. 예를 들어, 초기화 파일을 작성하여 서식문이나 다른 종류의 문서를 편집하기 위하여 왼쪽 여백과 오른쪽 여백을 재설정할 수도 있습니다.

8.6 HTPU 명령 파일의 사용

명령 파일은 편집기를 확장하거나 또는 수정하기 위한 HTPU 프로시저와 명령문을 포함하고 있는 파일입니다. 예를 들어, 명령 파일을 사용하여 추가의 HEVE 명령을 작성하거나 또는 키 정의, 속성을 설정할 수 있습니다. 또한, 명령 파일을 사용하여 사용자 자신의 어플리케이션 작성이나 일괄처리 편집을 위한 특수한 원문-처리 환경을 설정할 수 있습니다.

편집 세션을 시작하면, 내정에 의해 HEVE가 먼저 사용자의 현재 디렉토리에서 HTPU$COMMAND.TPU 파일을 탐색합니다. 다른 명령 파일을 사용하려면, 논리명 HTPU$COMMAND를 정의하거나 또는 /COMMAND 한정자를 사용하십시오. 예를 들어, 다음 명령은 사용자의 최상위 레벨(로그인) 디렉토리에 있는 MYPROCS.TPU 명령 파일을 사용하여 HEVE를 호출합니다.

```
$ DEFINE HTPU$COMMAND SYS$LOGIN:MYPROCS
$ EDIT/HTPU
```

위의 명령에서, 두가지 이상의 방법으로 명령 파일들을 지정한 경우에는, /COMMAND 한정자 다음에 논리명 HTPU$COMMAND, 그리고 그 다음에 현재 디렉토리 내의 명령 파일명을 지정해 주어야 합니다.

기동시, HTPU는 HEVE가 초기화 파일을 실행(있는 경우)하기 전에, 섹션 파일을 로드(있는 경우)한 다음에 명령 파일을 컴파일

하여 실행합니다. 따라서, 명령 파일 내의 프로시저, 설정값, 및 키 정의들이 섹션 파일내에 있는 것들을 무시합니다.

8.7 HTPU 섹션 파일의 사용

섹션 파일은 누적 특성을 갖습니다. 즉, 최종의 키 정의와 속성값 및 컴파일된 프로시저, 그리고 현재 사용중인 섹션 파일에 저장된 모든 것들을 저장합니다. 섹션 파일은 한번에 한개만 사용할 수 있습니다.

HEVE 호출시에 섹션 파일을 사용하려면, /SECTION 한정자를 사용하거나 또는 논리명 HTPU$SECTION을 정의하십시오. 이 중 한가지 방법으로 섹션 파일을 지정할 때에는, 섹션 파일명 외에 디스크와 디렉토리를 함께 지정해 주어야 합니다. 그렇지 않으면, HTPU가 섹션 파일이 SYS$SHARE에 있는 것으로 간주합니다. 예를 들어, 다음 명령은 사용자의 최상의 레벨(또는 로그인) 디렉토리 내의 MYEVE.TPU$SECTION 섹션 파일을 사용하여 HEVE를 호출합니다.

```
$ DEFINE HTPU$SECTION SYS$LOGIN:MYEVE
$ EDIT/HTPU
```

이 정의를 사용자의 LOGIN.COM 파일에 넣어 놓을 수도 있습니다. 섹션 파일을 위의 두가지 방법으로 지정할 때에는, 논리명 HTPU$SECTION 앞에 /SECTION 한정자를 지정해 주어야 합니다.

섹션 파일은 키 정의(사용자가 작성한 런 순서 등)와 컴파일한 임의의 HTPU 프로시저를 저장하는 파일입니다. 그러나, 섹션 파일은

여백, 탭, 및 일부의 글로벌 설정값과 같은 설정값들은 저장하지 않습니다. 일반적으로, 이러한 설정값들은 초기화 파일에 저장합니다.

8.8 기동 파일의 사용 비교

기동 파일에는 섹션 파일, 명령 파일 및 초기화 파일의 세가지 유형이 있습니다. 이들 각 파일은 각각의 특성이 있습니다. 이 절에서는 각 종류의 기동 파일의 장단점에 대하여 논의합니다. 이러한 내용은, HEVE 설정시에 어떤 종류의 기동 파일을 사용할 것인지를 결정하는데 참고할 수 있습니다.

섹션 파일은 누적 특성을 가지기 때문에, 최종의 내용만 유지하고 그 이전의 내용은 저장합니다. 이 때문에, 섹션 파일은 크기가 매우 큽니다(1000 블록 이상). 또한, 섹션 파일은 모든 글로벌 설정값과 어떠한 버퍼-특유의 설정값들도 저장하지 않습니다. 따라서, 일부의 설정값들은 명령 파일이나 또는 초기화 파일에 저장되어야 합니다. 그러나, 섹션 파일은 2진 형식의 파일이므로, 편집 환경을 매우 신속히 설정할 수 있습니다. 그리고, 이것이 HEVE를 설정하는데 있어서 가장 편리한 방법입니다. SAVE ATTRIBUTES 또는 EXTENDED HEVE 명령을 사용하여, 간단히 섹션 파일을 작성할 수 있습니다.

초기화 파일의 경우에는, 편집 환경을 설정하는 속도가 가장 늦습니다. 그리고, 사용자 자신의 초기화 파일을 작성할 수 있으나, 런 순서를 저장할 수는 없습니다. 그러나, 초기화 파일의 크기는 섹션

파일보다 작으며, 편집 속성을 저장하는데 제한사항은 없습니다.

명령 파일의 경우에는 편집 환경을 설정하는 속도가 섹션 파일보다 늦습니다. 새로운 HEVE 명령이나 또는 HTPU 프로시저를 사용하여 HEVE를 확장하기 위해서는 HTPU 프로그래밍에 대한 지식이 요구됩니다. 그러나, 명령 파일의 크기는 섹션 파일보다 작으며, 편집 속성 값들을 저장하는데 제한 사항이 없습니다. 또한, 기동시 명령 파일의 실행 속도는 초기화 파일의 실행 속도보다 빠릅니다.

8.9 논리명을 사용한 **HEVE** 키패드의 정의

HEVE 키패드를 선택하기 위하여 HEVE$KEYPAD 논리명을 정의할 수 있습니다.

이의 유효한 이름으로는 HEDT, EVE, NUMERIC, VT100 및 WPS가 있습니다. 논리명을 정의해 주지 않으면, HEVE의 내정 키패드가 VT100 단말기에는 VT100 키패드, 그리고 VT200 및 그 이후 계열 단말기에는 HEDT 키패드가 적용됩니다. 예를 들어, WPS 키패드를 사용하면서 HEVE를 호출하려면, 다음 명령을 사용하십시오.

```
$ DEFINE HEVE$KEYPAD WPS
$ EDIT/HTPU
```

이 논리명은 모든 편집 세션에서 동일한 키패드를 사용하고, 그 키패드 설정값들을 저장하기 위한 기동 파일을 작성하지 않을 경우에 유용합니다. 이 논리명은 섹션 파일에 저장되어 있는 모든 키패드 설정값을 무시합니다. 사용자의 초기화 파일이나 또는 명령 파일에

키패드를 설정하므로써, 이 논리명의 효과를 무시되게 할 수 있습니다.

사용자 자신의 키패드를 설정하여 HEVE를 확장시키는 경우에는, 사용자 키패드의 이름으로 사용될 논리명을 정의할 수도 있습니다. 예를 들어, 이름이 SIMPLE인 키패드를 작성하고, 그 키패드를 설정하는 프로시저의 이름이 EVE_SET_KEYPAD_SIMPLE인 경우에, 논리명을 SIMPLE로 정의하면 HEVE가 기동시에 사용자의 키패드를 호출합니다.

제 9 장

HEVE 외의 사항

HEVE 내에서 DCL 명령을 실행하거나, 또는 부프로세스를 사용하여 DCL 명령 레벨과 HEVE 편집 세션 사이를 신속하게 전환할 수 있습니다.

9.1 DCL 명령의 실행

HEVE 내에서 DCL 명령을 입력하려면, 명령 'DCL' 다음에, 실행할 DCL 명령을 지정하여 사용합니다. DCL 명령을 지정하지 않으면, HEVE가 그것을 입력하도록 프롬프트합니다. 예를 들어, HEVE 내의 현재 디렉토리에 있는 파일 목록을 보려면, 다음 명령과 같이 사용하십시오.

```
Command: DCL DIRECTORY
```

그러면, 메시지 창에 다음의 메시지가 표시됩니다.

```
Creating DCL subprocess ...
```

DCL 명령은 부프로세스를 생성합니다. HEVE는 부프로세스가 시스템에 의해 중지되지 않는 한, 편집 세션 중에 한번만 DCL 부프로세스를 생성하여 기동시킵니다.

DCL 명령이 실행되면, HEVE가 필요에 따라 다른 창을 작성한 후에

DCL 명령을 화면에 표시하고, 그 명령의 실행 결과를 DCL 버퍼에 출력합니다. 이때, 커서는 DCL 명령을 실행하기 전의 버퍼 내의 위치에 그대로 남아 있습니다. 이후에 커서를 DCL 버퍼로 이동하거나 또는 원문의 선택 및 제거, 또는 제6장에서 설명한 기술을 사용하여 그 원문을 편집 버퍼에 복사하는 작업 등을 수행할 수 있습니다. DCL 명령은 특히, HEVE 버퍼에 출력된 것을 복사(capture)할 경우에 유용합니다.

일부의 경우에는, DCL 부프로세스가, 명령이 완료된 후에까지 계속될 수 있습니다. 이 경우에는, 사용자가 해당 명령을 종료시키지 않는 한 다음의 후속 DCL 명령이 동작되지 않습니다. 예를 들어, DCL HELP 명령을 사용하여 시스템의 도움말 주제를 보다가 DCL DIRECTORY와 같은 다른 DCL 명령을 실행하려고 한 경우, 부프로세스가 아직 실행중이므로 원하는 결과를 얻을 수 없습니다. 이러한 경우에는 다음과 같이 하십시오.

1. 'DCL' 명령만을 사용하여, HEVE가 명령 문자열을 입력하도록 프롬프트하게 하십시오.

2. 이 프롬프트에서, CTRL/V 키(또는 QUOTE로 정의된 다른 키)를 누르십시오.

3. CTRL/Z 키를 눌러서, HELP 명령을 종료시키십시오.

그 다음에 다른 DCL 명령을 실행할 수 있습니다.

9.2 부프로세스의 작성

편집 세션을 중단하지 않고, HEVE 편집 세션과 DCL 명령 레벨 사이를 전환하도록 부프로세스를 생성할 수 있습니다. 부프로세스를 생성하려면, SPAWN 명령을 사용하십시오. 그러면, HEVE가 현재 편집 세션을 일시중단하고, 단말기를 새로운 부프로세스에 접속합니다. DECwindows 인터페이스에서는 SPAWN 명령을 사용할 수 없습니다. 그러나, SPAWN 명령을 사용할 필요 없이, 다른 DECwindows/Motif Hangul 어플리케이션을 실행하거나 또는 Hangul DECterm 창을 작성할 수 있습니다.

일반적으로 SPAWN 명령은 HMAIL 유틸리티와 같이 연속적인 출력을 생성하는 명령이나 유틸리티를 실행하거나 또는 자신의 화면 관리를 수행하기 위하여 사용합니다.

편집 세션으로 복귀하려면, DCL 명령 LOGOUT을 사용하여 부프로세스를 로그아웃하면 됩니다. 그러면, HEVE가 편집 세션을 재개하고, 커서는 부프로세스를 생성하기 전에 있던 위치에 나타납니다.

이와 달리, DCL 명령을 SPAWN 명령의 매개변수로 사용하여 특정 부프로세스를 생성할 수도 있습니다. 이 경우, HMAIL 유틸리티 명을 입력하도록 HMAIL> 프롬프트가 화면에 표시됩니다. HMAIL을 종료하면, 자동적으로 부프로세스가 로그아웃되고, HEVE가 편집 세션을 재개합니다.

DCL을 사용하여 프로세스를 작성하는 대신에, HEVE 편집 세션을

위한 프로세스를 생성한 다음에 DCL 명령과 유틸리티를 사용하기 위한 모 DCL 프로세스로 재접속할 수 있습니다.
이를 위해서는, 먼저 SPAWN 명령을 사용하여 편집 세션을 위하여 DCL 명령 레벨에서 부프로세스를 생성하십시오. 이 SPAWN 명령은 부프로세스(SHOW PROCESS 명령을 사용하면 "username_1"으로 화면에 표시됩니다)를 생성합니다. 이 부프로세스 레벨에서, HEVE를 호출하면 편집 세션으로 복귀합니다.

DCL 명령 레벨로 복귀하려면, HEVE 명령 ATTACH를 사용하여 모 프로세스(SHOW PROCESS 명령을 사용하면 "username"으로 화면에 표시되는 프로세스)로 복귀하십시오.

편집 세션을 재개하려면, DCL 명령 ATTACH 다음에 부프로세스 (username_1")의 프로세스 명을 지정하여 편집 부프로세스로 재접속하십시오.

SPAWN 명령과 ATTACH 명령을 사용하면, 전체 VMS 세션 동안에 편집 세션이 가동되게 유지할 수 있습니다. 이는 보다 많은 시스템 자원을 사용하면서 편집을 보다 신속히 재개할 수 있게 합니다.

제 10 장

HEVE DECwindows Motif/Hangul 인터페이스 사용

HEVE는 이전 장들에서 설명한 문자 셀 단말(HEVE/CCT) 사용자 인터페이스 외에도 DECwindows Motif/Hangul 사용자 인터페이스를 제공합니다. DECwindows Motif/Hangul 사용자 인터페이스(HEVE/Motif)에서 HEVE를 사용하면 DO 명령어와 사전 정의된 키로부터 뿐만 아니라 한글 풀다운 메뉴와 팝업 메뉴로부터 HEVE 명령을 실행할 수 있습니다

HEVE/Motif에서는 이전 장들에서 설명한 거의 모든 HEVE 명령어를 사용할 수 있습니다. HEVE/Motif에서 실행되지 않는 명령은 ATTACH, SPAWN 및 SPELL 뿐입니다.

이 장에서는 다음의 주제에 관하여 논의합니다.

- HEVE/Motif의 호출
- 커서 이동 및 화면이동
- 마우스 선택 및 신속 이동
- HEVE 명령 실행

10.1 시작하기 전에

DECwindows Motif/Hangul 어플리케이션을 사용하려면, 해당 어플리케이션 창이 들어갈 위치를 VMS 운영 체제에 알려주어야 합니다. 이것은 VMS 화면을 워크스테이션으로 설정하면 됩니다. 화면을

설정하려면 다음 DCL 명령어를 사용하십시오.

```
$ SET DISPLAY/CREATE/NODE=<YOUR NODE NAME>
```

화면을 설정하지 않으면, HEVE가 DECwindows Motif/Hangul 사용자 인터페이스를 사용하여 시작하지 않고, 문자 셀 단말기 인터페이스로 들어가게 합니다.

항상 동일한 워크스테이션을 사용하여 편집을 수행하면, 위의 DCL 명령을 LOGIN.COM에 넣는 노력을 줄일 수 있습니다.

10.2 DECwindows Motif/Hangul 인터페이스를 사용한 HEVE 호출

DECwindows Motif/Hangul 인터페이스를 사용하여 HEVE를 호출하려면 다음 명령을 사용하십시오.

```
$ EDIT/HTPU/DISPLAY=MOTIF
```

한정자 /DISPLAY는 DECwindows Motif/Hangul 인터페이스가 사용되도록 지정합니다. DECwindows Motif/Hangul 인터페이스를 가끔 사용하는 경우에는, 다음과 같이 외부 명령을 정의하여

```
$ HEVE :=EDIT/HTPU/DISPLAY=MOTIF
```

단순히 HEVE만 입력하면 DECwindows Motif/Hangul 인터페이스를 사용하여 HEVE가 기동되도록 하거나 또는 다음과 같이 논리명을

정의하여

```
$ DEFINE HTPU$DISPLAY_MANAGER MOTIF
```

HEVE가 기동될 때 내정으로 DECwindows Motif/Hangul 인터페이스가 사용되도록 할 수 있습니다.

문자 셀 단말 인터페이스를 사용하도록 복귀하려면, 다음 명령을 사용하여 HEVE를 기동시키십시오.

```
$ EDIT/HTPU/DISPLAY=CHARACTER_CELL
```

10.2.1 메뉴 언어 선택

DECwindow Motif/Hangul에서 HEVE는 영문과 한글 둘 다 풀다운 메뉴와 팝업 메뉴를 가지고 있습니다. 사용자는 DECwindow Motif/Hangul의 세션 관리자(Session Manager)에서 언어를 전환하는 것과 같은 방법으로 HEVE의 메뉴 원문 언어를 전환할 수 있습니다.

사용자는 세션 관리자의 "옵션" 풀다운 메뉴중 "언어..." 항목을 선택하기만 하면 됩니다. "언어 옵션" 팝업 메뉴에서 한글을 선택하면 한글 메뉴 원문으로 전환할 수 있으며, 영어를 선택하면 영어 메뉴 원문으로 전환할 수 있습니다.

10.2.2 한글 문자의 입력

한글을 입력하려면, 먼저 COMPOSE/Space[1] 키를 눌러서 한글 입력 방법이 시작되게 해야 합니다. 이는 COMPOSE 키와 스페이스 바 키를 동시에 누르면 됩니다.

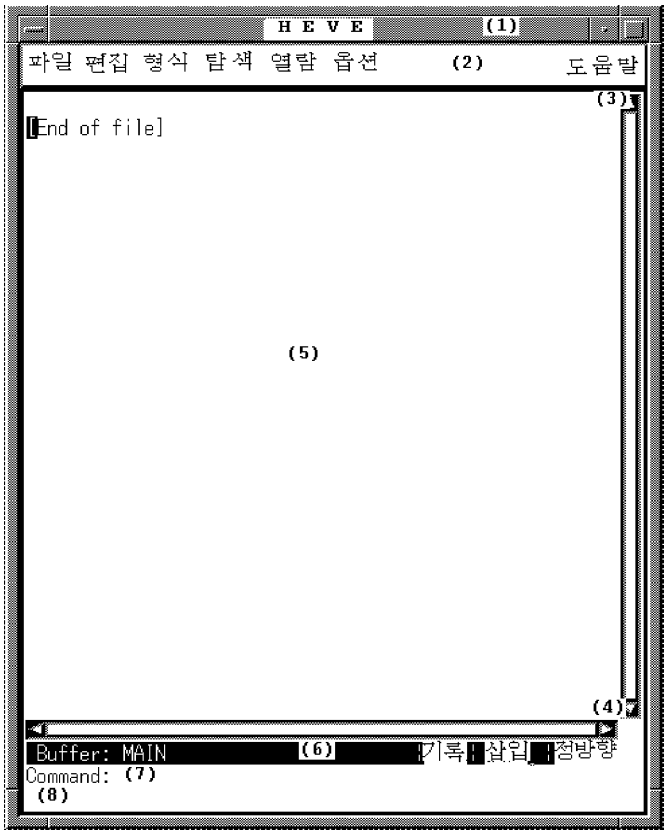
DECwindows Motif/Hangul은 클라이언트 프로세스를 사용하여 한글의 입력 순서를 처리하기 때문에, 원격 입력 클라이어언트가 종단된 경우에는 HEVE/Motif 입력 방법 프로세싱을 재설정하고 원격 입력 클라이언트를 재기동시켜야 합니다. HEVE/Motif 입력 방법을 재설정하려면, "옵션" 풀다운 메뉴의 "입력 방법 재설정" 메뉴 항목을 선택해야 합니다.

10.3 DECwindows Motif/Hangul 인터페이스 화면 구성

일단, 위에서 설명한 대로 DECwindows Motif/Hangul 인터페이스를 사용하여 HEVE를 호출하면, 편집 세션을 시작할 수 있으며, 이때 HEVE/Motif는 그림 10-1과 같은 창을 생성합니다.

[1] LK401 키보드에서는, COMPOSE 키만 사용합니다.

그림 10-1 HEVE DECwindows Motif/Hangul 편집 세션에서의 화면 구성



HEVE/Motif의 화면은 HEVE/CCT의 화면과 유사합니다. 이 화면의 각 구성 요소는 다음과 같습니다.

1. 제목 막대 - 어플리케이션의 이름을 보여 줍니다. 내정값은 HEVE입니다.

2. 메뉴 막대 - HEVE/Motif에서 사용할 수 있는 풀다운 메뉴를 보여 줍니다. 풀다운 메뉴 외에도 MB3[1] 단추를 사용하여 팝업 메뉴를 액세스할 수 있습니다.

3. 수직 화면이동 막대 - 주 창에 표시되는 현재 버퍼 부분을 보여 줍니다. 화면이동 막대 위에서 클릭하여 화면을 위 아래로 이동할 수도 있습니다.

4. 수평 화면이동 막대 - 주 창을 이동시키면서 보여 줍니다. 화면이동 막대 위에서 클릭하여 화면을 좌우로 이동할 수 있습니다.

5. 주 창 - 편집 중인 버퍼를 보여 줍니다. 이것은 HEVE/CCT 인터페이스의 주 창과 동일합니다.

6. 상태 행 - HEVE의 현재 상태를 보여 줍니다. 이것은 HEVE/CCT 인터페이스의 상태 행과 동일합니다.

7. 명령 창 - 실행할 HEVE 명령어를 지정할 위치를 제공합니다. 이것은 HEVE/CCT 인터페이스의 명령 창과 동일합니다.

8. 메시지 창 - HEVE 명령에 의해 표시된 최종 메시지를 보여

[1]마우스의 세번째 단추입니다.

줍니다. 이것은 HEVE/CCT 인터페이스의 메시지 창과 동일합니다.

10.4 커서이동 및 화면이동

HEVE/Motif를 사용하여 편집할 경우, MB1[1]을 클릭하여 커서를 창의 어느 위치로든지 이동할 수 있습니다. 뿐만 아니라, 화면이동 막대 위에서 클릭하거나 끌기 기능을 사용하여 창을 위 아래로 또는 좌우로 이동할 수 있습니다. 또한, 마우스를 사용하여 일부 HEVE 기능을 설정할 수 있습니다.

마우스를 오프(off)로 설정하여, 일부 마우스 기능을 가동 중지시킬 수 있습니다.

10.4.1 커서 위치 지정

주 창의 임의 위치로 커서를 이동하려면, 먼저 커서를 위치시키려는 셀에 가리키개를 위치시킨 후 MB1을 클릭하십시오.

주 창에 2개 이상의 창이 있는 경우, 원하는 창에 가리키개를 위치시킨 후 MB1을 클릭하여 한 창에서 다른 창으로 커서를 이동할 수 있습니다. 이 방법은 HEVE 명령어인 Next Window를 입력하는 것보다 시간이 절약됩니다.

[1]마우스의 첫번째 단추입니다.

10.4.2 명령 창에서의 클릭

명령 창에 가리키개를 위치시킨 후 클릭하여 HEVE 명령 모드를 기동할 수 있습니다. 이 방법은 DO 키를 누르는 것과 동일한 효과를 가집니다.

10.4.3 상태 행에서의 클릭

HEVE/Motif에서는 상태 행에서 클릭하여 일부 기능을 설정할 수 있습니다. 이를 요약하면 다음 표와 같습니다.

표 10-1 상태 행의 서로 다른 구성 요소 위에서 MB1을 클릭한 효과

상태 행 구성 요소	효과
버퍼명	명령 Next Buffer를 실행합니다.
쓰기/읽기 전용	버퍼를 수정/수정불가능 모드로 전환시킵니다.
삽입/겹쳐쓰기	삽입/겹쳐쓰기 모드로 전환합니다.
정방향/역방향	내정 동작 방향을 정방향/역방향으로 전환시킵니다.

10.4.4 화면 이동 막대 위에서의 클릭

DECterm 또는 Hangul DECterm에서 사용할 경우, HEVE/CCT 인터페이스에서도 지원되는 커서 이동 방식 외에, 마우스를 사용하여서도 각각 수평 및 수직 화면이동 막대 위에서 클릭하여 좌우 및 위 아래로 화면을 이동할 수 있습니다.

화면이동 막대의 각 영역에서 MB1을 클릭하거나 끌었을 때 나타나는 효과는 다음과 같습니다.

- 수직 화면이동 막대의 스테핑 화살표 위에서 클릭할 경우
  => 한 행 화면이동합니다.
- 수직 화면이동 막대의 화면이동 영역 위에서 클릭할 경우
  => 한 화면만큼 화면이동합니다.
- 수직 화면이동 막대의 슬라이더를 끌 경우
  => 원하는 버퍼 위치로 화면이동합니다.
- 수평 화면이동 막대의 스테핑 화살표 위에서 클릭할 경우
  => 한 열만큼 이동합니다.
- 수평 화면이동 막대의 화면이동 영역 위에서 클릭할 경우
  => 화면 폭 만큼 이동합니다.
- 수평 화면이동 막대의 슬라이더를 끌 경우
  => 원하는 버퍼 위치로 이동합니다.

10.4.5 마우스를 오프(off)로 설정하기

주 창, 상태 행 및 명령 버퍼에서의 마우스 클릭 효과는 오프시킬 수 있지만, 화면이동 막대 위에서의 마우스 효과는 오프시킬 수 없습니다.

마우스 효과를 오프하려면, 다음 HEVE 명령을 사용하십시오.

```
Command: HTPU SET (MOUSE,OFF)
```

이 명령은 DECterm을 사용할 때 HEVE/CCT 인터페이스에서도 사용될 수 있습니다.

10.5 선택 및 클라이언트간 통신

HEVE/Motif는 1차 및 2차 선택을 통해 클라이언트간 통신을 제공합니다. 글로벌 선택을 사용하는 외에, DECwindows Motif/Hangul 클립보드를 사용하여서도 다른 DECwindows Motif/Hangul 어플리케이션과 통신할 수 있습니다.

다음의 명령을 사용하여 HEVE/Motif에서의 마우스 기능을 off로 설정하여, 마우스를 사용한 선택을 중지시킬 수 있습니다.

```
Command: HTPU SET (MOUSE,OFF)
```

10.5.1 1차 선택

HEVE/Motif에서, 선택 범위는 DECwindows Motif/Hangul의 1차 선택으로 변환됩니다. HEVE/Motif에서는 다음과 같은 방법으로 선택 범위를 생성할 수 있습니다.

1. SELECT 명령이나 또는 SELECT 명령으로 사전 정의된 키를 사용하십시오.

2. 주 창에서 MB1을 누른 상태로 끄십시오.

3. 선택하려는 원문의 양에 따라 다음과 같이 MB1을 클릭하십시오.

- 현재 단어를 선택하려면 두번 클릭하십시오.
- 현재 행을 선택하려면 세번 클릭하십시오.
- 현재 단락를 선택하려면 네번 클릭하십시오.
- 현재 버퍼의 모든 내용을 선택하려면 다섯번 클릭하십시오

선택된 영역은 HEVE/CCT의 경우와 같이 역상으로 나타납니다. DECwindows Motif/Hangul에 1차 선택이 있는 경우, HEVE/Motif를 포함한 모든 어플리케이션에서 MB2[1] 를 눌러서 그 선택 값을 조회할 수 있으며, 특히 HEVE/Motif에서는 1차로 선택한 것이 MB2를 눌렀을 때 가리키개가 놓인 버퍼 위치에 삽입됩니다 이때, 그 위치의 오른쪽 원문은 밀려 납니다.

10.5.2 2차 선택

DECwindows Motif/Hangul의 2차 선택을 사용하면 한 창의 원문을 다른 창으로 신속하게 복사할 수 있습니다. 이 방법을 신속 이동(Quick Transfer)이라 합니다.

HEVE/Motif에서 2차 선택은 주 창의 원문 위에서 MB2를 누른 상태로 끌면 됩니다. 선택 내용은 HEVE/Motif에서는 밑줄 표시된 화상으로 표시되며, 다른 DECwindows Motif/Hangul 어플리케이션에서는 임의의 화상으로 표시됩니다. MB2에서 손가락을 떼면 2차로 선택한 것이 즉시 복사되어 HEVE/Motif의 현재 커서 위치에 삽입됩니다. 이때, 커서의 오른쪽에 있는 원문은 밀려납니다.

MB2를 눌러도 HEVE/Motif의 커서 위치에는 영향을 주지 않습니다

[1]마우스의 두번째 단추입니다.

10.5.3 1차 및 2차 이동

글로벌로 선택한 것은 HEVE/Motif로 또는 HEVE/Motif부터만 복사할 수 있으며, 또한 글로벌로 선택한 것은 이동할 수도 있습니다 1차 선택을 이동하려면 MB2를 누르고 있는 동안 COMPOSE 키를 누르고 있어야 합니다. 2차 선택을 이동하려면, 2차 선택을 위해 MB2를 끄는 동안 COMPOSE 키를 누르고 있어야 합니다.

삭제된 원문은 HEVE 시스템 버퍼에 저장되며, Restore Selection 명령을 사용하여 복구할 수 있습니다. Restore Selection 명령은 최종적으로 이동된 원문을 현재 버퍼의 편집 위치에 삽입합니다.

10.5.4 클립보드

내정에 의해, HEVE/Motif는 다른 DECwindows Motif/Hangul 어플리케이션과 통신하기 위해 DECwindows Motif/Hangul Clipboard를 사용하지 않습니다. 다음 HEVE 명령을 사용하여 Clipboard를 가동할 수 있습니다.

```
Command: SET CLIPBOARD
```

Clipboard를 가동 중지하려면, 다음 명령을 사용하십시오.

```
Command: SET NOCLIPBOARD
```

Clipboard가 가동되면, HEVE/Motif는 선택 내용을 DECwindows Motif/Hangul로 복사하여 그것을 다른 어플리케이션이 읽을 수 있게 합니다.

10.6 메뉴를 사용한 편집

위에서 언급한 대로, HEVE/Motif에서의 편집에는 명령 버퍼에 HEVE 명령을 입력하는 것 외에도 메뉴를 사용할 수 있습니다.

HEVE 명령은 HEVE/CCT 인터페이스에서와 동일한 방법으로 사용할 수 있습니다. 즉, Do 키를 눌러 명령 버퍼로 간 다음 모호하지 않은 약어 형태로 명령을 입력한 후 Return 키를 누르면 입력된 명령이 실행됩니다.

사전 정의된 키를 사용하여 HEVE/Motif에서 HEVE 명령을 실행할 수도 있습니다. HEVE 명령어의 목록과 그에 대응되는 사전 정의된 키에 대해서는 부록 B를 참조하십시오.

아래 표는 메뉴 항목과 그 항목에 해당하는 HEVE 명령을 수록하였습니다. HEVE 명령의 효과에 대해 알아보려면, 이전 장들을 참조하십시오.

영어 메뉴를 선택하면, 메뉴 항목의 레이블은 그 항목이 호출하는 명령어와 동일하게 됩니다.

표 10-2 파일 풀다운 메뉴

한글 레이블	영어 레이블	기능
신규	New	새로운 버퍼 생성
선택된것 열기	Open Selected	선택 또는 탐색된 파일 열기
열기...	Open...	지정 파일을 현 HEVE창에 삽입 또는 필요한 경우 새 버퍼 생성
포함...	Include...	현재 행위치에 지정한 파일의 내용 삽입
파일 보관	Save File	편집 세션을 종료하지 않고 파일로 보관
파일 지정 보관...	Save File As...	편집 세션을 종료하지않고 지정한 파일에 보관
종료	Exit	모든 버퍼를 저장한 후 편집 세션을 종료
중지	Quit	버퍼를 저장하지 않고 편집 세션을 종료

표 10-3 편집 풀다운 메뉴

한글 레이블	영어 레이블	기능
복원	Restore	마지막으로 지운 내용을 복구
잘라냄	Cut	선택한 범위를 INSERT HERE 버퍼로 이동
복사	Copy	선택한 범위를 INSERT HERE 버퍼로 복사
붙임	Paste	INSERT HERE 버퍼를 현 커서 위치로 붙임
대체...	Replace..	지정한 "구 문자열"을 탐색하여 지정한 "새 문자열"로 대체
모두 선택	Select All	현재 버퍼 모두 선택

표 10-4 형식화 풀다운 메뉴

한글 레이블	영어 레이블	기능
단락 재구성	Fill Paragraph	현재 단락을 재구성
영역 재구성	Fill Range	선택한 범위를 재구성
행 중앙화	Center Line	현재 행을 좌우 여백이 같게 중앙 정렬
대소문자 변경	Change Case	대/소문자로 전환
페이지 매김	Paginate	매 54행 마다 "소프트" 페이지 구분을 삽입
여백 설정	Set Margins...	왼쪽 및 오른쪽 여백을 설정
단락 내려쓰기 설정...	Set Paragraph indent...	단락의 첫번째 행의 들여쓰기를 지정한 값으로 설정

표 10-5 탐색 풀다운 메뉴

한글 레이블	영어 레이블	기능
다음 찾기	Find Next	FIND를 사용하여, 입력된 문자열탐색 다음에 나타나는 것을 탐색
찾기...	Find...	지정한 문자열을 탐색
와일드 카드 사용 찾기	Wildcard Find...	와일드카드 문자를 사용하여 탐색
선택된 것 찾기	Find Selected	현재 선택한 원문을 탐색
대체...	Replace...	지정한 "구 문자열"을 탐색하여, 그것 모두를 지정한 "새 문자열"로 대체
와일드 카드 표시	Show Wildcards	현재 인식 가능한 와일드카드 문자를 표시
빈 칸 설정	Set Whitespace	Find Whitespace 또 는 Find NoWhitespace 설정
와일드 카드 설정	Set Wildcard	Wildcard VMS 또는 Wildcard Ultrix를 설정

표 10-6 열람 풀다운 메뉴

한글 레이블	영어 레이블	기능
두 창	Two Windows	현재의 창을 크기가 같은 두 개의 창으로 분리
한 창	One Window	현재의 창을 제외한 모든 창을 삭제
창 분할	Split Window...	현재의 창을 두 개 이상의 창으로 분리
창 삭제	Delete Window	HEVE에서 두 개 이상의 창을 사용하고 있는 경우, 현재 창을 삭제
버퍼 표시	Show Buffers	버퍼 명을 입력하지 않고도 버퍼를 열람하거나 삭제할 수 있도록, 생성한 버퍼 목록을 표시하고 그 목록 안에 커서를 위치시킴
폭 설정...	Set Width...	화면 폭을 지정한 값으로 설정

표 10-7 옵션 풀다운 메뉴

한글 레이블	영어 레이블	기능
런	Learn	키에 연계시킬 일련의 키조작을 시작함
확장	Extend	Extend 명령을 사용하여 HEVE를 확장
메뉴확장...	Extend Menu...	메뉴 항목을 정의
글로벌 속성	Global Attributes...	글로벌 속성을 설정
버퍼 속성...	Buffer Attributes...	버퍼 속성을 설정
탐색 속성	Search Attributes	빈공백 및 와일드카드 탐색을 설정
폭 설정	Set Width...	화면 폭을 지정한 값으로 설정
속성 보관...	Save Attributes...	섹션 파일을 생성하거나 명령 파일을 생성 또는 갱신하여, 글로벌 설정과 기타 설정 사항을 저장
시스템 속성 보관...	Save System Attributes...	섹션 파일을 생성하거나 명령 파일을 생성 또는 갱신하여, HEVE 내정 속성과 메뉴 정의를 저장함

표 1-7 (계속) 옵션 풀 다운 메뉴

한글 레이블	영어 레이블	기능
글자꼴 크기	Font Size	주 창의 글자 크기를 "크게"/"작게"로 설정
입력 방법 재설정	Reset Input Method	원격 입력 방법 프로세스가 재시작될 경우, 한글 입력 처리를 재가동시킴

표 10-8 선택 팝업 메뉴

한글 레이블	영어 레이블	기능
재설정	Reset	선택 또는 기타 동작을 취소하고 정방향으로 설정
첫 문자 대문자화	Capitalize Word	현재 단어의 첫번째 문자는 대문자로, 그 단어의 나머지 문자는 소문자로 변경
단어 전체 대문자화	Uppercase Word	현재 단어의 문자를 모두 대문자로 바꿈
단어 전체 소문자화	Lowercase Word	현재 단어의 문자를 모두 소문자로 바꿈
다음 찾기	Find Next	Find를 사용하여, 이미 입력한 원문 문자열 다음에 나타나는 것을 탐색

표 1-8 (계속) 선택 팝업 메뉴

한글 레이블	영어 레이블	기능
잘라냄	Cut	선택한 영역을 INSERT HERE 버퍼로 이동
복사	Copy	선택한 영역을 INSERT HERE 버퍼로 복사
선택된 것 열기	Open Selected	선택하거나 탐색한 이름의 파일 열기
선택된 것 찾기	Find Selected	현재 선택한 원문을 탐색

표 10-9 무선택(No-Selection) 팝업 메뉴

한글 레이블	영어 레이블	기능
선택	Select	선택 항목 위에서 편집할 원문을 선택함
모두 선택	Select All	현재 버퍼의 모든 내용을 선택함
단어 복원	Restore Word	이전에 Erase Word 명령을 사용하여 지운 단어를 복구
선택 영역 복원	Restore Selection	삭제 대기 또는 DECwindows Quick Copy를 사용하여 마지막으로 지운 내용을 복구

표 1-9 (계속) 무선택(No-Selection) 팝업 메뉴

한글 레이블	영어 레이블	기능
행 복원	Restore Line	Erase Line 또는 Erase Start of line을 사용하여 마지막으로 지운 행 복원
문자 복원	Restore Character	Erase Character를 사용하여 마지막으로 지운 문자 복원
문장 복원	Restore Sentence	WPS Delete Beginning Sentence 키를 사용하여 마지막으로 지운 문장을 복원
다음 찾기	Find Next	Find를 사용하여, 이미 입력한 원문 문자열이 다음에 나타나는 것을 탐색
단락 재구성	Fill Paragraph	현재 단락을 재구성
행 중앙화	Center Line	현재 행을 좌우 여백이 같게 중앙 정렬함
붙임	Paste	INSERT HERE 버퍼 내용을 현재 커서 위치에 붙임
대체...	Replace...	지정한 "구 문자열"을 탐색하여, 모두 지정한 "새 문자열"로 대체

표 10-9 (계속) 무선택(No-Selection) 팝업 메뉴

한글 레이블	영어 레이블	기능
표시	Mark...	현재 커서 위치에 표시를 하여, 나중에 그 위치로 이동할 수 있게 함
이동...	Go To...	지정한 표시기 위치로 이동
포함...	Include...	현재 행을 시작하기 전에, 지정한 파일의 내용을 현재 버퍼에 삽입

제 11 장

저널링 및 복구

저널 파일은 편집 내용을 기록하여, 시스템 장애로 인하여 편집 세션이 중단되는 경우에, 작업 내용을 복구할 수 있게 합니다. HEVE는 다음 두 가지 유형의 저널링 및 복구 방법을 제공합니다.

- 버퍼 변경 저널링(Buffer-change journaling)
  버퍼 변경 저널링은 사용자가 생성하는 각각의 원문 버퍼에 대해 별도의 저널 파일을 생성합니다. 이것은 HEVE의 내정으로 설정된 것입니다. 버퍼 변경 저널링은 DECwindows Motif/Hangul 인터페이스와 문자 셀 단말기 인터페이스 모두에서 작동합니다. 사용자는 다른 편집 세션으로부터 버퍼를 복구할 수 있습니다. 일반적으로, HEVE에서 RECOVER BUFFER 명령을 사용하여 한 번에 하나의 버퍼를 복구할 수 있습니다. 복구 명령으로 복구되는 것은 원문 뿐이며 시스템 장애 이전의 설정값, 키 정의 또는 시스템 버퍼(예를 들면, INSERT HERE 버퍼)의 내용은 복구되지 않습니다.

- 키조작 저널링(Keystroke journaling)
  키조작 저널링은 편집 세션을 위하여 하나의 저널 파일을 생성합니다. 키조작 저널링은 문자 셀 단말기 인터페이스에서만 작동합니다. DECwindows Motif/Hangul 인터페이스에서는 작동하지 않으며, 다른 제한 사항을 가집니다. 복구는 키조작을 하나씩 하나씩 재현하여 편집 세션을 재생성합니다.

일반적으로, 그렇게 할 이유는 없지만, 사용자는 편집 세션에 두

가지 유형의 저널링을 모두 가질 수 있습니다. 대개는, 버퍼 변경 저널링이 제한 사항이 적고 복구 속도가 빠르기 때문에 더 좋은 방법입니다.

명령 행의 /NOJOURNAL 한정자를 사용하여 HEVE를 호출할 경우에는, 두 가지 저널링을 모두 가동중지시킬 수 있습니다. 이 경우, 기동은 보다 빠르지만 편집 세션동안 시스템 장애가 일어나는 경우 작업 내용을 상실할 위험성이 있습니다. 이 방법은 편집을 하지 않고 파일을 검사하거나 데모 작업에서 HEVE를 사용할 경우에 유용한 방법입니다.

11.1 버퍼 변경 저널링

버퍼 변경 저널링은 각 원문 버퍼를 위하여 하나의 저널 파일을 생성합니다(HEVE는 INSERT HERE 버퍼, DCL 버퍼, 또는 $RESTORE$ 버퍼와 같은 시스템 버퍼에 대해서는 버퍼 변경 저널 파일을 생성하지 않습니다). 버퍼를 편집할 때에, 저널 파일은 지우기, 삽입 또는 원문의 재형식화와 같이 사용자가 가한 변경 내용을 기록합니다. HEVE를 종료하거나 버퍼를 삭제하는 경우, 저널 파일도 삭제됩니다. 시스템 장애로 인하여 편집 세션이 중단되면, 저널 파일이 저장됩니다. 시스템 장애가 일어나기 이전의 약간의 키조작 내용은 상실될 수 있습니다.

버퍼 변경 저널 파일은 논리명 HTPU$JOURNAL로 정의된 디렉토리에 기입됩니다. HTPU$JOURNAL이 정의되지 않은 경우, 일반적으로 사용자의 최상위 레벨(로그인) 디렉토리인 SYS$SCRATCH가 사용됩니다. 저널 파일을 다른 디렉토리에 기입하기 위해 HTPU$JOURNAL

논리명을 재정의할 수도 있습니다(이 논리명은 키조작 저널 파일에 적용되지 않습니다). 예를 들어, 아래의 명령은 [USER.JOURNAL]이라는 부디렉토리를 생성한 후, 이 부디렉토리에 HTPU$JOURNAL을 정의합니다.

```
$ CREATE/DIRECTORY [USER.JOURNAL]
$ DEFINE HTPU$JOURNAL [USER.JOURNAL]
```

이 정의를 사용자의 LOGIN.COM 파일에 넣을 수 있습니다.
버퍼 변경 저널 파일은 상당히 클 수 있으며, 심지어는 편집하는 원문 파일보다 더 클 수도 있습니다. 버퍼 변경 저널 파일의 잠재적인 크기와 각 원문 버퍼마다 저널 파일이 있기 때문에 HTPU$JOURNAL을 SYS$SCRATCH보다 용량이 큰 디스크에 하나의 디렉토리나 부디렉토리로 정의하는 것이 좋습니다.

버퍼 변경 저널 파일명은 편집할 파일이나 버퍼명으로부터 도출되며, 내정 파일 유형은 .TPU$JOURNAL입니다. 예를 들면, 다음과 같습니다.

원문 버퍼	버퍼 변경 저널 파일
MAIN	MAIN.TPU$JOURNAL
JABBER.TXT	JABBER_TXT.TPU$JOURNAL
GUMBO_RECIPE.RNO	GUMBO_RECIPE_RNO.TPU$JOURNAL
NEW TEST DATA	NEW_TEST_DATA.TPU$JOURNAL
* TEMP *	_ _TEMP_ _.TPU$JOURNAL

현재 버퍼의 저널 파일명을 알아보려면, SHOW 명령을 사용하십시오

11.1.1 버퍼 변경 저널링을 사용한 복구

버퍼 변경 저널 파일을 사용하여 편집 내용을 복구하는 데는 다음의 두 가지 방법이 있습니다.

- HEVE를 호출할 때 명령 행에 /RECOVER 한정자를 사용하는 방법
- HEVE에서 RECOVER BUFFER 명령을 사용하여 복구할 버퍼명이나 파일명 또는 해당 버퍼에 대한 저널 파일명을 지정하는 방법

아래의 예에서는 사용자가 JABBER.TXT라는 파일을 편집하고 있을 때 시스템 장애로 인해 편집 세션이 중단되고, 이후 편집 내용을 복구하기 위해 시스템 복구를 사용하는 것을 보여줍니다.

```
$ EDIT/HTPU JABBER.TXT
  .
  .
  .
*** system failure ***
  .
  .
  .
$ EDIT/HTPU JABBER.TXT /RECOVER
```

이와 달리, HEVE를 호출하여, 아래 명령을 사용하여 원문을 복구할 수도 있습니다.

```
Command: RECOVER BUFFER JABBER.TXT
```

버퍼 변경 저널 파일이 사용 가능한 경우에는, HEVE가 다음 정보를 보여주며 해당 버퍼를 복구할 것인지 묻습니다.

버퍼명
해당 버퍼에 대한 원래 입력 파일(있는 경우)
해당 버퍼에 대한 출력 파일(있는 경우)
복구용 소스 파일(있는 경우)
편집 세션의 시작 일-시
저널 파일 생성 일-시

버퍼를 복구하려면 Return 키를 누르고, 그렇지 않으면 N을 입력하십시오. 복구용 원시 파일은 삭제 또는 재명명할 수 없습니다. 그렇지 않으면 그 복구는 실패하게 됩니다. 원시 파일은 초기에 버퍼(있는 경우)로 읽어들인 파일이거나 시스템 장애 이전에 기입한 최종 파일입니다.

복구하려는 버퍼가 이미 있는 경우(대개는 주 버퍼), HEVE는 먼저 그 버퍼를 삭제한 다음 복구합니다. 복구하려는 버퍼가 수정된 경우, HEVE는 복구 전에 그 버퍼를 삭제할 것인지의 여부를 묻습니다.

버퍼 변경 저널 파일의 버전이 두 개 이상인 경우, RECOVER BUFFER는 버전 번호가 가장 높은 파일을 사용합니다. 그러나, 복구하려는 버퍼가 이미 있는 경우에는, 가장 최근에 갱신된 버퍼 변경 파일이 해당 버퍼에 대해 잠기며(locked), HEVE는 바로 한 단계 낮은 버전의 버퍼 변경 저널 파일을 사용할 것인지를 묻습니다.

이 버퍼를 복구하려면 Return 키를 누르고, 그렇지 않으면 N을 입력하십시오.

다른 편집 세션으로부터 버퍼를 복구할 수 있습니다. 버퍼명이나 저널 파일명이 확실치 않은 경우에는 아래와 같이 별표(*) 와일드 카드를 지정하십시오.

```
Command: RECOVER BUFFER *
```

그러면, HEVE가 원하는 것을 선택할 수 있도록 시스템 버퍼 $CHOICES$에 사용 가능한 저널 파일 목록을 모두 보여 줍니다. 원문 버퍼 모두를 한번에 복구하려면, RECOVER BUFFER ALL 명령을 사용하십시오. 그러면, HEVE가 버퍼 변경 저널 파일이 사용가능한 각 원문 버퍼의 복구를 시도합니다. 이때의 효과는 버퍼명이나 저널 파일명을 지정하지 않고 RECOVER BUFFER 명령을 반복해서 사용하는 것과 같습니다. 각각의 원문 버퍼에 대해 HEVE는 다음 중 하나를 프롬프트합니다.

응답	효과
YES	버퍼를 복구하고 있는 경우, 다음 버퍼의 복구 여부를 묻습니다. 이 응답이 내정 응답입니다. 단순히 Return 키만 눌러도 됩니다.
NO	복구를 생략합니다. 복구할 버퍼가 또 있는 경우에는, HEVE가 그 버퍼의 복구 여부를 묻습니다.
QUIT	복구 동작을 취소합니다. 해당 버퍼는 복구되지 않으며, 복구 동작은 계속되지 않습니다.

SET NOJOURNALING 명령을 사용하여 특정 버퍼에 대한 버퍼 변경 저널링을 가동 중지하고, 버퍼명을 지정할 수 있습니다. 지정된 버퍼명이 없으면, HEVE가 버퍼명을 입력하도록 프롬프트합니다. 모든 버퍼에 대해 버퍼 변경 저널링을 가동 중지하려면 SET NOJOURNALING ALL 명령을 사용하십시오. 일반적으로, 키조작 저널링을 사용하거나, 해당 버퍼가 단순히 "스크래치 패드"인 경우 또는 파일을 읽기 위한 임시 저장 영역인 경우와 같이, 편집 내용을 저널할 필요가 없는 경우에 버퍼 변경 저널링을 가동 중지합니다.

SET NOJOURNALING은 버퍼에 대한 저널 파일을 삭제하지 않습니다. 예를 들어, 버퍼 변경 저널 파일을 모두 삭제하려면, 다음 DCL 명령을 사용하십시오.

```
$ DELETE HTPU$JOURNAL:*.TPU$JOURNAL;*
```

버퍼 변경 저널링을 가동 중지한 경우, SET JOURNALING 명령 다음에 원하는 버퍼명을 지정하여 다시 저널링을 가동할 수 있습니다. 지정된 버퍼명이 없으면, HEVE가 버퍼명을 입력하도록 프롬프트합니다. 예를 들어, 다음 명령은 버퍼명이 JABBER.TXT인 버퍼를 저널링하게 합니다.

```
Command: SET JOURNALING JABBER.TXT
```

저널링 없이 HEVE를 호출한 경우, 편집 세션 동안에 버퍼 변경 저널링을 가동하려면 SET JOURNALING ALL 명령을 사용하십시오(다른 경우라면 이것이 HEVE 내정값입니다).

버퍼가 수정된 경우에는 버퍼 변경 저널링을 가동할 수 없습니다. 그 경우 HEVE는 다음과 같은 메시지를 화면에 표시합니다.

```
Command: SET JOURNALING MEMO.TXT
Buffer MEMO.TXT is not safe to journal (plase write it out).
```

이때는, 먼저 WRITE FILE 또는 SAVE FILE 명령을 사용하여 버퍼를 기입한 후 저널링을 가동해야 합니다.

11.2 키조작 저널링 및 복구

키조작 저널링은 사용자가 생성하는 버퍼의 수에 관계없이 편집 세션을 위하여 한개의 저널 파일을 생성합니다. 저널 파일은, 단순히 원문을 변경하기보다는, 명령을 포함하여 편집 세션에서의 키조작을 기록합니다. 키조작 저널링은 문자 셀 단말기 인터페이스에서만 작동하며 DECwindows Motif/Hangul 인터페이스에서는 작동하지 않습니다.

편집 세션을 종료하면, 키조작 저널 파일이 삭제됩니다. 시스템 장애로 인하여 편집 세션이 중단되면 저널 파일이 저장됩니다. 시스템 장애가 일어나기 이전의 약간의 키조작 내용은 상실될 수 있습니다.

키조작 저널링은 디버깅을 목적으로 한 문제를 기록(및 재생성)하는데 특히 유용합니다. HEVE나 HTPU에 문제가 있는 경우, 소프트웨어 성능 보고(SPR)를 실행하려면 키조작 파일(있는 경우), SHOW SUMMARY 명령의 실행 결과, 문제에 대한 설명 및 기타 관련

파일을 제출해야 합니다.

11.2.1 키조작 저널링의 가동

키조작 저널링을 가동하려면, HEVE를 호출할 때 /JOURNAL 한정자를 사용하여 생성할 키조작 저널 파일을 지정하십시오. 키조작 저널 파일의 내정 파일 유형은 .TJL입니다. 이 저널 파일은 현재의 디렉토리나 또는 명령 행에 지정한 디렉토리에 기입됩니다.

11.2.2 키조작 저널링을 사용한 복구

키조작 저널 파일을 사용하여 편집 내용을 복구할 경우, 명령 행에 저널 파일명을 지정해야 합니다. /JOURNAL 한정자를 사용하지 않고 /RECOVER 한정자를 사용하면, HEVE가 버퍼 변경 저널 파일로 복구를 시도합니다. 아래의 예는 키조작 저널 파일을 사용한 복구의 예입니다.

```
$ EDIT/HTPU/JOURNAL=MYJOU.TJL JABBER.TXT
  .
  .
  .
*** system failure ***
  .
  .
  .
$ EDIT/HTPU/JOURNAL=MYJOU JABBER.TXT/RECOVER
```

키조작 저널링을 사용하여 복구하면, 키조작이 하나씩 하나씩 재현되어 편집 세션이 재생됩니다. 그 다음에, 종료하면 복구된 원문이

저장됩니다.

키조작 저널 파일을 사용하여 편집 내용을 복구하기 전에, 관련된 모든 파일과 단말기 설정값이 편집 세션을 처음 시작했을 때와 동일한지 확인하십시오. 그렇지 않으면, 복구는 실패합니다.

HTPU 명령 행에 지정한 다중 입력 파일에 편집세션을 복구하려면, 원래의 명령 행과 동일한 순서로 입력 파일을 지정해야 합니다. 이것은 원래의 편집 내용이 버퍼 생성 순서에 의존하는 경우에 중요합니다.

원래 편집 세션 동안에 어떤 동작을 중지 또는 취소하기 위해 CTRL/C 키를 사용한 경우, 키조작 저널 파일을 사용한 복구가 작동하지 않을 수도 있습니다. 키조작 저널링은 CTRL/C 를 기록하지 않습니다. 따라서, 키조작을 재현하면 중단된 그 동작이 계속됩니다. 이것은 나머지 키조작이 재현되는 방법에 영향을 주기 쉬우며, 그 경우 복구는 실패할 수 있습니다.

11.3 요약

저널링과 복구가 상당히 신뢰성이 있기는 하지만, 시스템 장애로부터 사용자의 작업 내용을 보호하기 위한 가장 안전한 방법은 편집 내용을 자주, 특히 편집 세션 중간 중간에 기입하는 것입니다.

WRITE FILE 또는 SAVE FILE 명령을 사용하여 편집 세션을 종료하지 않고도 편집 내용을 기입할 수 있습니다.

아래 표에 서로 다른 /[NO]JOURNAL 한정자 조합과 저널 설정상의 SET NOJOURNALING ALL 명령을 사용했을 경우의 효과를 요약해 놓았습니다.

동작	효과
내정	버퍼 변경 저널링만 가동되고 키조작 저널링은 가동 중지됩니다.
HEVE 호출시 /JOURNAL= 한정자 사용	버퍼 변경 저널링과 키조작 저널링 모두 가동됩니다.
HEVE 호출시 /NOJOURNAL 한정자 사용	버퍼 변경 저널링과 키조작 저널링 모두 가동 중지시킵니다. 편집 세션 동안 버퍼 변경 저널링을 재가동하려면, SET JOURNALING ALL 명령을 사용하십시오.
HEVE 호출시 /JOURNAL= 한정자 사용. 그리고 HEVE 내에서 SET NOJOURNALING ALL 사용	키조작 저널링만 가동되고 버퍼 변경 저널링은 가동 중지됩니다. 편집 세션동안 버퍼 변경 저널링을 재가동하려면, SET JOURNALING ALL 명령을 사용하십시오.

부록 A

HTPU 프로시저의 예

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! This example of HTPU procedures is to autoshift the screen when   !!!
!!! the cursor reaches the screen boundary. There are 3 new commands: !!!
!!! SET [NO]AUTOSHIFT and RESTORE SHIFTING. The MOVE LEFT and MOVE    !!!
!!! RIGHT commands are modified and superseded the standard HEVE      !!!
!!! procedures to support autoshifting. By default, the screen         !!!
!!! is shifted 15 columns at a time.                                  !!!
!!!                                                                   !!!
!!! The SET [NO]AUTOSHIFT commands are to disable/enable              !!!
!!! autoshifting.    The RESTORE SHIFTING command is to undo          !!!
!!! the shifting, i.e. shifting the screen left with the columns      !!!
!!! that you have shifted right.                                      !!!
!!! Please note that it is only an example. Some functionalities      !!!
!!! do not complete. For example, suppose you enable autoshifting     !!!
!!! and have shifted the screen, when you use the START OF LINE       !!!
!!! command, theediting point is moved to the start of line            !!!
!!! but cursor may not as the screen is shifted.                      !!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

procedure eve_set_autoshift                      ! enable autoshifting

set (RIGHT_MARGIN_ACTION, current_buffer);   ! disable the default action
                                              ! (word wrapping)
eve_set_nowrap;                               ! of cursor when it attempts
                                             ! to pass beyond the right
                                             !margin
set (DETACHED_ACTION, SCREEN, "heve$$autoshift");

endprocedure;     ! eve_set_qutoshift

procedure eve_set_noautoshift       ! disable autoshifting
                                    ! reenable word wrap
eve_restore_shifting;
set (RIGHT_MARGIN_ACTION, current_buffer, eve$kt_word_wrap_routine);
set (DETACHED_ACTION, SCREEN, eve$kt_detached_action_routine);

endprocedure;           ! eve_set_noautoshift

procedure eve_restore_shifting
! Undo the shifting, i.e. shifting the screen to the left with the
! columns that you have shifted right.
```

```
eve_shift_left (shift (current_window, 0));

endprocedure;           ! eve_restore_shifting

procedure heve$$autoshift

! The cursor is said to be detached when it is loose from the editing
! point.  This procedure is the core procedure of autoshifting.

local the_reason;

the_reason := get_info (SCREEN, "detached_reason");

if the_reason = tpu$k_off_right     ! the cursor is detached because
                                    ! it is moved off the right
                                    ! boundary of the screen
then
  eve_shift_right (shift_amount);     ! shift 15 columns to the right
else
  if the_reason = tpu$k_off_left    ! the cursor is detached because
                                    ! it is moved off the left screen
                                    ! boundary
  then
    eve_shift_left (shift_amount);    ! shift 15 columns to the left
  endif;
endif;

endprocedure;      ! heve$$autoshift

procedure eve_move_left  ! Move left one column (free and bound cursor)
                         ! supersede the standard HEVE eve_move_left
                         ! procedure
              ! Move left one column.  Do not wrap at edge of the screen.

on_error
   [OTHERWISE]:
endon_error;

if not eve$declare_intention (eve$k_action_up_left)
then
  return (FALSE);
endif;

if eve$x_bound_cursor
then
   move_horizontal (-1);
else
   align_cursor;
   if cursor_horizontal (-1, CHARACTERS) =0
   then
```

```
       eve_shift_left (shift_amount);
eve$learn_abort;
return FALSE;
   endif;
endif;

return (TRUE);

endprocedure; ! eve_move_left

procedure eve_move_right ! Move right one column (free and bound cursor)
                         ! supersedes the standard HEVE eve_move_right
                         ! procedure
           ! Move right one character. Do not wrap at edge of the screen.

on_error
   [OTHERWISE]:
endon_error;

if eve$in_prompting_window
then
   if current_character =""
   then
  return (TRUE);
   endif;
endif;

if not eve$declare_intention (eve$k_action_down_right)
then
   return (FALSE);
endif;

if eve$x_bound_cursor
then
   move_horizontal (1);
else
   align_cursor;
   if (cursor_horizontal (get_info (current_buffer, "character_length")) = 0)

   then
      eve_shift_right (shift_amount);
      eve$learn_abort;
      return (FALSE);
   endif;
endif;
return [TRUE];

endprocedure; ! eve_move_right
constant shift_amount :=15;        ! default number of shifted column
```

부록 B

HEVE 명령

아래 표에는 모든 HEVE 명령을 분류하고, 그 각각에 대한 간단한 설명이 되어 있습니다.

HEVE 명령은 10개의 범주로 분류되며, 다음과 같습니다.

- 원문 편집, 표 B-1은 제2장에서의 원문 편집 명령을 설명합니다.
- 상자 편집, 표 B-2는 제2장에서의 상자 편집 명령을 설명합니다.
- 커서 이동 및 화면이동, 표 B-3은 제2장에서의 명령을 설명합니다.
- 형식화, 표 B-4는 제3장에서의 원문 형식화 명령을 설명합니다.
- 탐색, 표 B-5는 제4장에서의 탐색 동작을 설명합니다.
- 창 및 화면, 표 B-6은 제6장에서의 창 조작을 위한 명령을 설명합니다.
- 파일 및 버퍼, 표 B-7은 제6장에서의 버퍼 조작 명령과 제11장에서의 파일 관련 명령을 설명합니다.

- 키 정의, 표 B-8은 제7장에서의 키 정의 명령을 설명합니다.
- 범용 명령, 표 B-9는 제1장 및 제9장에서의 여러가지 명령과 제5장에서의 선/상자 그리기 명령을 설명합니다.
- 설정, 표 B-10은 제8장에서의 설정을 위한 명령을 설명합니다.

표 B-1 원문편집

명령	정의된 키	설명
Change Mode	CTRL/A,F14	삽입모드/겹쳐쓰기 모드로 전환합니다.
Copy		선택한 영역을 INSERT HERE 버퍼로 복사합니다.
Cut	REMOVE,KP6	선택한 영역을 INSERT HERE 버퍼로 이동합니다.
Delete		커서 왼쪽 문자를 삭제합니다.
Erase Character	COMMA	커서 위치의 문자를 지웁니다.
Erase Line	PF4	현재 문자로 시작하는 한 행을 지웁니다. 다음 행(있는 경우)이 한행 위로 이동됩니다.

표 B-1 (계속) 원문 편집

명령	정의된 키	설명
Erase Previous Word	F13	커서가 단어 사이에 있는 경우 앞 단어를 지웁니다. 아니면 현재 단어의 커서 왼쪽 문자를 지웁니다.
Erase Start of Line	CTRL/U	행의 처음까지 지웁니다.
Erase Word	MINUS	커서가 단어 사이에 있는 경우 다음 단어를 지웁니다. 아니면 현재 단어의 커서 오른쪽 문자를 지웁니다.
Insert Here	INSERT	INSERT HERE 버퍼를 현재 버퍼의 커서 위치로 복사합니다.
Insert Mode	CTRL/A,F14	삽입모드로 변경합니다
Overstrike Mode	CTRL/A,F14	겹쳐쓰기 모드로 변경합니다.
Paste	INSERT	INSERT HERE 버퍼를 현재 버퍼의 커서 위치로 복사합니다.
Remove	REMOVE,KP6	선택한 영역을 INSERT HERE 버퍼로 이동합니다.
Restore	GOLD-INSERT	마지막으로 지운 것을 복구합니다.

표 B-1 (계속) 원문 편집

명령	정의된 키	설명
Restore Character	GOLD-COMMA	Erase Character를 사용하여 마지막으로 지운 문자를 복구합니다.
Restore Line	GOLD-PF4	Erase Line 또는 Erase Start of Line을 사용하여 마지막으로 지운 행을 복구합니다.
Restore Selection		삭제 대기 또는 DECwindows/Motif move를 사용하여 마지막으로 지운 것을 복구합니다.
Restore Sentence		WPS Delete Beginning Sentence 키를 사용하여 마지막으로 지운 문장을 복구합니다.
Restore Word	GOLD-MINUS	Erase Word로 지운 단어를 복구합니다.
Select	SELECT	선택 항목에서 편집할 원문을 선택합니다.
Select All		현 버퍼 모두 선택합니다.
Store Text	GOLD-REMOVE	선택한 영역을 INSERT HERE 버퍼로 복사합니다.

표 B-2 상자 편집

명령	설명
Box Copy	상자를 INSERT HERE 버퍼로 복사합니다.
Box Cut	상자를 INSERT HERE 버퍼로 이동시킵니다.
Box Cut insert	상자를 잘라내어, 상자 오른쪽의 원문을 왼쪽으로 당겨서 간격을 없앱니다.
Box Cut Overstrike	상자를 잘라내어, 상자의 오른쪽 원문의 열 정렬을 그대로 유지하도록 잘라낸 자리를 빈 공백으로 메꿉니다.
Box Paste	마지막으로 자르거나 복사한 것을 삽입합니다. 이때 대개는 기존 원문에 겹쳐쓰기됩니다.
Box Paste Insert	마지막으로 자르거나 복사한 것을 삽입합니다. 이때 기존 원문은 상자 오른쪽으로 밀려납니다
Box Paste Overstrike	마지막으로 자르거나 복사한 것을 삽입합니다. 이때 기존 원문은 상자로 겹쳐쓰기됩니다.
Box Select	표준 방식의 행 선택이 아닌 상자 방식으로 선택합니다.
Restore Box Selection	삭제 대기 또는 DECwindows/Motif moves를 사용하여 마지막으로 지운 것을 복구합니다.
Set Box Nopad	버퍼의 모드가 겹쳐쓰기 모드가 아닌 경우, 채우기 및 겹쳐쓰기를 가동 중지합니다.
Set Box Noselect	상자 스타일 선택, 오려붙이기를 중지합니다
Set Box Pad	버퍼의 모드에 관계없이 상자 편집시 채우기 및 겹쳐쓰기를 가동합니다.
Set Box Select	상자 스타일 선택, 오려붙이기를 가동합니다.

표 B-3 커서 이동 및 화면이동

명령	정의된 키	설명
Bottom	GOLD-DOWN	커서를 현재 버퍼의 마지막 행에 위치시킵니다
Change Direction	F11	탐색 및 이동 동작을 정방향/역방향으로 전환합니다.
End of Line	CTRL/E, GOLD-RIGHT	커서를 현재 행의 끝에 위치시킵니다.
Forward		탐색 방향을 정방향으로 변경합니다.
Go To		표시기를 지정한 위치로 이동시킵니다.
Line		커서를 지정한 행에 위치시킵니다.
Mark		커서 위치에 표시를 하여, 나중에 그 위치로 이동할 수 있게 합니다.
Move By Line		커서를 현재 방향으로 한 번에 한 행씩 이동시킵니다.
Move By Page	KP7	커서를 현재 방향으로 한 번에 한 페이지씩 이동시킵니다.
Move By Word	KP1	커서를 현재 방향으로 한 번에 한 단어씩 이동시킵니다.
Move Down	DOWN	커서를 한 번에 한 행씩 아래로 이동시킵니다.
Move Left	LEFT	커서를 한 번에 한 문자씩 왼쪽으로 이동시킵니다.
Move Right	RIGHT	커서를 한 번에 한 문자씩 오른쪽으로 이동시킵니다.

표 B-3 (계속) 커서 이동 및 화면이동

명령	정의된 키	설명
Move Up	UP	커서를 한 번에 한 행씩 위로 이동시킵니다.
Next Screen	NEXT SCREEN	다음 화면의 원문을 볼 수 있도록 수직으로 화면이동시킵니다.
Previous Screen	PREV SCREEN	이전 화면의 원문을 볼 수 있도록 수직으로 화면이동시킵니다.
Reverse		탐색 방향을 역방향으로 변경시킵니다.
Set Cursor Bound		HEDT에서와 유사하게, 커서 고정 모드를 가동시킵니다.
Set Cursor Free		커서를 화면의 어느 위치로든 이동할 수 있도록, 커서 자유 모드를 가동시킵니다.
Set Display Cursor		커서가 항상 다중 바이트 문자의 첫번째 바이트에 위치하도록 커서 화면표시 모드를 가동시킵니다.
Set Nodisplay Cursor		커서 화면표시 모드를 가동 중지시킵니다.
Set Scroll Margin		커서를 위 아래로 이동할 때에 자동으로 화면 이동이 시작되는, 상단 및 맨 하단 간격을 설정합니다.
Start of Line	CTRL/H, GOLD-LEFT	커서를 현재 행의 끝에 위치시킵니다.
Top	GOLD-UP	커서를 현재 버퍼의 첫 행에 위치시킵니다.
What Line		현재 커서가 위치해 있는 행의 행 번호를 화면에 표시합니다.

표 B-4 형식화

명령	정의된 키	설명
Capitalize Word		현재 단어의 첫번째 문자는 대문자로, 나머지 문자는 소문자로 변경합니다.
Center Line		현재 행을 좌우 여백이 같게 중앙 정렬시킵니다.
Convert Tabs		탭 문자를 공백으로 변환합니다.
Fill	GOLD-KP8	현재 단락을 재형식화합니다.
Fill Paragraph		현재 단락을 재형식화합니다.
Fill Range		선택한 영역을 재형식화합니다.
Fullform Word		현재 단어의 ASCII 문자를 전자 문자로 만듭니다.
Halfform Word		현재 단어의 ASCII 문자를 반자 문자로 만듭니다.
Insert Page Break	CTRL/L	커서 위치에 페이지 구분점을 삽입합니다.
Left Adjust		선택한 원문을 왼쪽 여백에 맞춰 정렬시킵니다
Left Indent		첫번째 선택한 열과 오른쪽 여백 사이에 선택한 원문을 정렬시킵니다.
Lowercase Word		현재 단어의 문자를 소문자로 만듭니다.
Paginate		매 54행마다 "소프트" 페이지 구분점을 삽입합니다.

표 B-4 (계속) 형식화

명령	정의된 키	설명
Right Adjust		선택한 원문을 오른쪽 여백에 맞춰 정렬시킵니다.
Right Indent		첫번째 선택한 열과 왼쪽 여백 사이에 선택한 원문을 정렬시킵니다.
Set Fill [No]tags		Fill이 DOCUMENT와 DSR 단락 태그를 이해하도록 또는 이해하지 못하게 합니다.
Set Left Margin		현재 버퍼의 왼쪽 여백을 지정한 값으로 설정합니다. 첫번째 열은 1입니다.
Set [No]wrap		오른쪽 여백을 넘어가는 경우에 자동적으로 줄바꿔쓰기를 가동/가동중지합니다.
Set Paragraph Indent		단락의 첫번째 행의 들여쓰기를 지정한 값으로 설정합니다.
Set Right Margin		현재 버퍼의 오른쪽 여백을 지정한 값으로 설정합니다. 이에는 마지막 열이 포함됩니다.
Set Tabs		현재 버퍼에 대한 탭 멈춤, TAB의 모드나 효과 또는 편집하는 동안 탭 문자 표시 등을 설정합니다.
Uppercase Word		현재 단어의 문자를 대문자로 만듭니다.

표 B-5 탐색

명령	정의된 키	설명
Find	FIND, GOLD-PF3	지정한 원문의 문자열을 탐색합니다.
Find Next	PF3	Find를 사용하여, 이미 입력한 원문 문자열이 다음에 나타나는 것을 탐색합니다.
Find Selected		현재 선택한 원문을 탐색합니다.
Global Replace		지정한 "구 문자열"을 탐색하여 모두 지정한 "새 문자열"로 대체합니다.
Replace		지정한 "구 문자열"을 탐색하여, 필요한 경우, 지정한 "새 문자열"로 대체합니다.
Set Find Case [No]Exact		탐색시에 대소문자 구분을 가동/가동중지합니다
Set Find Exact		탐색시에 대소문자 구분 및 문자크기 구분을 가동합니다.
Set Find Form [No]Exact		탐색시에 문자크기 구분을 가동/가동중지합니다
Set Find General		탐색시에 대소문자 구분 및 문자크기 구분을 가동중지시킵니다.
Set Find Nowhite space		공백 및 탭을, 탐색 문자열에 있는 것과 정확히 일치시키도록 FIND 및 WILDCARD FIND를 설정합니다.
Set Find Whitespace		공백, 탭 및 행 구분점을 "빈 공백"으로 처리하도록 FIND와 WILDCARD FIND를 설정합니다

표 B-5 (계속) 탐색

명령	정의된 키	설명
Set Wildcard ULTRIX		WILDCARD FIND가 Ultrix 와일드카드 문자를 인식하게 합니다.
Set Wildcard VMS		WILDCARD FIND가 VMS 와일드카드 문자를 인식하게 합니다.
Show Wildcard		현재 인식가능한 와일드카드 문자를 보여줍니다
Spell		DECspell을 호출하여 선택 범위 또는 버퍼 전체의 철자를 점검합니다.
Wildcard Find		와일드카드 문자를 사용하여 탐색합니다.

표 B-6 창 및 화면

명령	정의된 키	설명
Delete Window		HEVE에서 두 개 이상의 창을 사용하고 있는 경우, 현재 창을 삭제합니다.
Enlarge Window		현재 창의 높이를 지정한 행 수 만큼 확대합니다.
Next Window	GOLD-NEXT	다음 창을 표시합니다.
One Window		현재의 창을 제외한 모든 창을 삭제합니다.
Previous Window	GOLD-PREV	이전 창을 표시합니다.

표 B-6(계속) 창 및 화면

명령	정의된 키	설명
Refresh	CTRL/W	화면을 재생합니다.
Set Width		화면 폭을 지정한 값으로 설정합니다.
Shift Left		현재 창을 지정한 열 수 만큼 버퍼의 왼쪽으로 이동합니다.
Shift Right		현재 창을 지정한 열 수 만큼 버퍼의 오른쪽으로 이동합니다.
Shrink Window		현재 창의 높이를 지정한 행 수 만큼 축소시킵니다.
Split Window		현재 창을 두 개 이상의 창으로 분리시킵니다.
Two Windows		현재 창을 크기가 같은 두 개의 창으로 분리시킵니다.

표 B-7 파일 및 버퍼

명령	설명
Buffer	지정한 버퍼를 HEVE의 현재 창으로 집어 넣습니다.
Delete Buffer	지정한 버퍼를 삭제합니다.
Get File	지정한 파일을 HEVE의 현재 창으로 집어 넣고, 필요한 경우 새로운 버퍼를 생성합니다.
Get Wild File	다중 파일이 동시에 열린다는 것을 제외하면, Get File과 유사합니다.

표 B-7 (계속) 파일 및 버퍼

명령	설명
Include File	현재 행이 시작하기 전에, 지정한 파일의 내용을 현재 버퍼에 집어 넣습니다.
New	새 버퍼를 생성합니다.
Next Buffer	다음 버퍼를 HEVE의 현 창으로 넣습니다.
Open	지정한 파일을 HEVE의 현재 창으로 넣습니다. 필요한 경우, 새로운 버퍼를 생성합니다.
Open Selected	선택하거나 탐색한 이름의 파일을 엽니다.
Previous Buffer	이전 버퍼를 HEVE의 현재 창으로 넣습니다.
Recover Buffer	버퍼 변경 저널 버퍼를 복구합니다.
Recover Buffer All	찾은 버퍼 변경 저널링 파일을 모두 복구합니다.
Save File	편집 세션을 종료하지 않고 현재 버퍼를 파일에 기입합니다.
Save File As	편집 세션을 종료하지 않고 현재 버퍼를 지정한 파일에 기입합니다.
Set Buffer	버퍼의 편집 속성을 설정합니다: READ_ONLY, WRITE, MODIFIABLE, UNMODIFIABLE
Set [No]Journaling	지정한 버퍼에 대해 버퍼 변경 저널링을 가동/가동중지합니다.
Set [No]Journaling All	모든 버퍼에 대해 버퍼 변경 저널링을 가동/가동중지합니다.

표 B-7 (계속) 파일및 버퍼

명령	설명
Show Buffers	생성한 버퍼 목록을 표시하고 커서를 그 목록에 위치시켜서, 버퍼명을 입력하지 않고 버퍼를 열람하거나 삭제할 수 있게 합니다.
Show System Buffers	HEVE 시스템 버퍼의 목록을 표시합니다.
Write File	편집 세션을 종료하지 않고 현재 버퍼를 파일에 기입합니다.

표 B-8 키 정의

명령	정의된 키	설명
Define Key		키에 HEVE 명령을 연계시킵니다.
Learn	CTRL/K	키에 연계될 일련의 키조작을 시작하게 합니다
Remember	CTRL/R	런(Learn)에 의해 수행되는 일련의 키조작을 키에 연계시킵니다.
Set Func Key [No]Motif		일부 기능 키를 다른 DECwindows/Motif 어플리케이션에서와 같이 정의 또는 재정의합니다
Set [No]Gold key		키를 GOLD 키로 정의/정의해제합니다.
Set Keypad [No]HEDT		숫자 키패드와 다른 키들을 정의하는 HEDT 스타일 키패드를 가동/가동중지합니다.
Set Keypad [No]WPS		WPS 스타일 키패드를 가동/가동중지시킵니다.
Set Keypad Numeric		숫자 키패드를 내정 상태로 설정합니다.
Set Keypad VT100		VT100 스타일 키패드를 설정합니다.
Show Key		지정한 키의 현재 정의를 보여줍니다.
Undefine Key		Define Key를 사용하여 정의한 키를 정의 해제합니다.

표 B-9 범용 명령

명령	정의된 키	설명
Attach		지정한 VMS 프로세스에 접속시킵니다.
DCL		편집 세션으로부터 지정한 DCL을 실행하고, 그 결과를 HEVE의 두번째 창의 DCL 버퍼로 집어 넣습니다.
Do	DO	HEVE 명령을 실행합니다.
Draw Box		HEVE에서 상자 그리기 모드를 시작합니다.
Draw Line		HEVE에서 선 그리기 모드를 시작합니다.
Exit	CTRL/Z, F10	모든 버퍼를 기입하고 편집 세션을 종료합니다
Help	HELP	지정한 주제에 대한 도움말 원문을 검색합니다
Quit		버퍼를 보관 않고 편집 세션을 종료합니다.
Recall	CTRL/B	이전 HEVE 명령을 재호출 합니다.
Repeat	GOLD-%	다음 명령을 % 번, 또는 사전 정의된 키를 사용하지 않는 경우 임의번 반복합니다.
Reset	GOLD-SELECT	선택 또는 다른 동작을 취소하고 방향을 정방향으로 설정합니다.
Show		현재 버퍼에 대한 정보를 보여줍니다.
Spawn		현재 편집 세션을 종료하지 않고 일시 중지한 부프로세스를 생성합니다.
Symbol		2-바이트 기호 문자를 사용하여 그립니다.

표 B-10 설정

명령	설명
@	파일로부터 HEVE 명령을 읽고 실행합니다.
Define Menu Entry	지정한 풀다운 메뉴에 항목을 추가하고, 그 항목을 HEVE 명령에 연계시킵니다.
Extend All	현재 버퍼 내의 모든 HTPU 프로시저를 컴파일합니다.
Extend HEVE	지정한 현재 버퍼 내의 HTPU 프로시저를 컴파일합니다.
Extend This	커서 위치의 HTPU 프로시저를 컴파일합니다.
HTPU	HTPU 명령문을 실행합니다.
Save Attributes	섹션 파일을 생성하거나 또는 명령 파일을 생성 또는 갱신하여 글로벌 설정 및 기타 설정값들을 저장합니다.
Save Extended HEVE	섹션 파일을 생성하여, 글로벌 설정 및 기타 설정값들을 저장합니다.
Save System Attributes	섹션 파일을 생성하거나 또는 명령 파일을 생성 또는 갱신하여 HEVE 내정 속성과 메뉴 정의를 저장합니다.
Set Clipboard	복사, 오려붙이기를 위해 DECwindows 클립보드를 가동시킵니다.
Set [No]Default Command File	HEVE가 속성 저장을 위해 생성 또는 갱신하려는 명령 파일을 결정/취소시킵니다.
Set [No]Default Section File	HEVE가 속성 저장을 위해 생성 또는 갱신하려는 섹션 파일을 결정/취소시킵니다.

표 B-10 (계속) 설정

명령	설명
Set [No]Exit Attribute Check	종료시, HEVE가 변경한 내용의 보관 여부를 묻도록 또는 묻지 않도록 속성 점검을 가동/가동중지시킵니다.
Set [No]Pending Delete	Delete를 사용하거나 새 원문을 입력할 때, 선택한 것을 지우는 기능을 가동/가동중지시킵니다.
Set [No]Section Prompting	Save Attributes 또는 Save System File Attributes 명령 사용시, 섹션 파일을 프롬프트하는 기능을 가동/가동중지시킵니다.
Set Selection Grab Focus	HEVE 입력 포커스를 제공하는 경우에만 Grab DECwindows 1차 선택을 그래빙(grabbing)하는 기능을 가동시킵니다.
Set Selection Grap Selection	HEVE내의 원문을 선택하는 경우에만 Grab DECwindows 1차 선택을 그래빙하는 기능을 가동시킵니다.
Show Defaults Buffer	새 버퍼를 생성할 때 템플리트로 사용되는 HEVE 시스템 버퍼 $DEFAULTS$에 관한 정보를 보여줍니다.
Show Summary	SPR을 실행할 때 유용한 통계 및 기타 HEVE에 관한 정보를 보여줍니다.
Undefine Menu Entry	지정한 풀다운 메뉴에서 항목을 제거합니다.

부록 C

지원되는 HEDT 기능

표 C-1에서는 HEDT 특유의 기능과, 동일하거나 유사한 기능을 수행하지만 구문이 다른 HEVE가 지원하는 등가 기능을 비교합니다.

EDT가 제공하는 기능과 그에 해당하는 EVE 등가 기능에 대한 비교는 "Extensible Versatile Editor Reference Manual"를 참조하십시오.

표 C-1 HEDT 명령과 HEVE 명령간의 대응 관계

HEDT 명령	HEVE 명령	HEVE 등가 명령에 대한 주석
Set Default key	Set keypad	
Set [No]DispCol	HTPU Get_Info (SCREEN,"current _column")	HEVE 명령은 모든 키 조작에 대해 계속적으로 상태를 갱신하기에는 비효과적입니다.
Set [No]DispLine	What Line	버퍼 상단에 행 번호를 표시합니다.
Set [No]DispMode		상태 행에 삽입/겹쳐쓰기 모드를 표시합니다.
Set [No]DispStatus		상태 행에는 항상 삽입/겹쳐쓰기, 정방향/역방향, 및 기입/판독이 표시됩니다.

표 C-1 (계속) **HEDT** 명령과 **HEVE** 명령간의 대응 관계

HEDT 명령	HEVE 명령	HEVE 등가 명령에 대한 주석
Set [No]DispWrap	Show	왼쪽 및 오른쪽 여백을 볼 수 있으며, 자동 줄바꿔쓰기를 가동합니다.
Set Entity NoFirst	HTPU Set(Fill_ Not_Begin)	
Set Entity NoLast	HTPU Set(Fill_ Not_End)	
Set Wrap	Set Right Margin, Set Wrap	줄바꿔쓰기 원하는 열로 오른쪽 여백을 먼저 설정한 후, 자동 줄바꿔쓰기를 가동시킵니다
Set MGNL,MGNL	Set Left margin	
Set MGNR,MGNR	Set Right Margin	
Set EXECMD	@	HEVE는 HEVE 명령어만 이해합니다.
Set Search	Set Find Case [No]Exact, Set Find Form [No]Exact	
Set Shift		부록 A의 예제를 참조하십시오.
Set Terminal		지원되지 않습니다.
Show Command		지원되지 않습니다.

표 C-1 (계속) **HEDT** 명령과 **HEVE** 명령간의 대응 관계

HEDT 명령	HEVE 명령	HEVE 등가 명령에 대한 주석
Show Entity NoFirst	HTPU Get_info (SYSTEM,"Fill_ Not_Begin")	
Show Entity NoLast	HTPU Get_Info (SYSTEM,"Fill_ Not_End")	
Show Key	Help Keys	
Show Search		지원되지 않습니다.
Show Shift		지원되지 않습니다.
Show Terminal		지원되지 않습니다.
ADJC	Center Line	
ADJL	Left Adjust	
ADJR	Right Adjust	
AI	Insert Mode	
AO	Overstrike Mode	
CHGC	HTPU Change_ Case(Select_ range,INVERT)	먼저 대소문자를 변경할 원문을 선택한 후, HTPU의 이 내장 기능을 사용합니다. HEVE 명령에는 원문을 대문자화,소문자화,첫 문자만 대문자화하는 명령이 있습니다.

표 C-1 (계속) **HEDT** 명령과 **HEVE** 명령간의 대응 관계

HEDT 명령	HEVE 명령	HEVE 등가 명령에 대한 주석
CHGF	HTPU Change_Size (Select_range, Full_FORM)	먼저 전자로 변경할 원문을 선택합니다.
CHGH	HTPU Change_Size (Select_range, ASCII_CHAR)	먼저 반자로 변경할 원문을 선택합니다.
CHGL	Lowercase Word	
CHGU	Uppercase Word	
COLSEL	Box Select	
Fill	Fill	현재 Fill 명령은 정렬 기능이 없으므로 채움 이후에 Justify 내장 기능을 사용할 수 있습니다.
HDate		지원되지 않습니다.
NCode		지원되지 않습니다.
NDTL	Left Indent	
NDTR	Right Indent	
PLOT	DRAW LINE	
RI		선 그리기 문자를 보통 문자로 취급합니다.
RO		선 그리기 문자를 보통 문자로 취급합니다.