

フロッピーディスクを用いた「ノート環境」の一例

——短い文章を互いに関連づけて書くために——

奥村 健一

1. はじめに

パーソナルコンピューターを用いて「ノート環境」を試作した。使用する機械は数年前のまま、机の上を不当に大きく占めており、外観はとてもノートとはいえない。それでも現在の私はこれをノートのようなものとしてとらえている。ただし、紙に書く習慣を否定する考えは全くない。これから提示するものは、紙のノートとはまた一味違った「考えを書きとめるための行為」のほんの一例に過ぎない。

最初からひとつのまとまった文章を書こうとするのであれば、今回のテーマである「ノート環境」はいっさい無用である。しかし、ときどき浮かんでくるアイデアや、対象を観察しているときに生まれる疑問などは、さまざまに枝が分かれるように生じてくる。また、ひとつの考えがいくつかの対象や領域に関連するばあいもある。部分的に生じる記述をひとつのまとまった流れの中に納める前に、さまざまな枝分かれの状態を保ったまま保存するのに適した文房具がないだろうかと考えてきた（図1を参照）。

最近では、パーソナルコンピューターでも、ハードディスクや光磁気ディスクなどの記憶媒体によって、1ギガバイト（1,000メガバイト）にも達する外部記憶を読み書きできるようになった。これは大型の辞典が納まってしまふ容量である。何もかも1か所に放り込んで集中させておくメリットは大きい。しかし、長期間にわたって同じスタイルで蓄積しながら使うデータもあれば、あ

る程度でひと区切りつけるほうが良い対象もある。これらはむしろ積極的に分けて扱いたいと考えた。「ノート環境」は主に後者のためのものであり、フロッピーディスクを用いることを中心としている。1メガバイトのフロッピーディスクであれば、日本語の文字が約50万字書ける。これは書き方にもよるが、大学ノート数冊分に相当する。このフロッピーディスク1枚の中で、自然にコンピュータの特質が利用できることを目標とした。以後「ノート」は紙のノートを意味し、パーソナルコンピュータを用いたノートのようなものは「ノート環境」と呼ぶことにする。

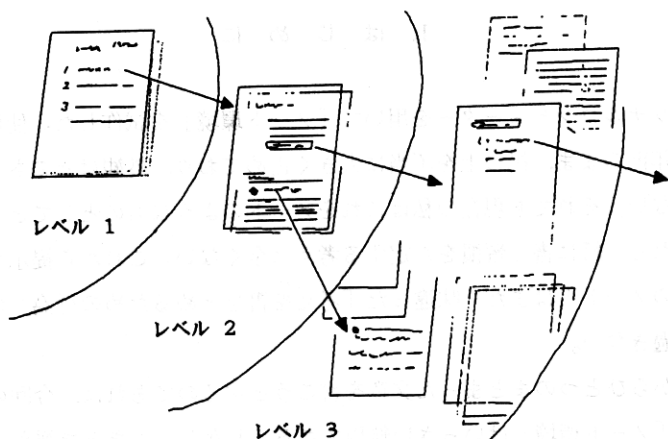


図1 記述の連なり

2. ノートは異質なコンピュータの利点

ノートは大別して2つある。ひとつは日記帳や大学ノートなどのようにページが固定的なもので、必ず書いた時間順に文章が集まる特徴がある。書いた結果が静的であることは書く行為の基本的な前提である。あとから変更することのできない記録こそが重要である場合も多い。もうひとつはルーズリーフなどのようにページの移動ができるもので、バインダーに綴じたあともページの差し替え、挿入、再編成ができる。

ノートにはその他に記入欄、大きさ、装丁、紙質などさまざまな機能、表情があり、その人の意識にきめ細かく対応させて使うことができる。また、紙に書く行為自体が楽しみや刺激となる面もあり、「考える」、「まとめる」などの目的を感覚面で援助する道具としても、ノートは非常に優れた性質をもっている。書いた文字はその人ごとの形であり、記号以上の意味を備えている。

しかし、ノートでは時としてめんどうが生じることもある。そこで、どのようなばあいにもノートではめんどうになるかを以下に列挙し、それぞれについてコンピューターにできることを併記してみる。

2.1 書くテーマをきめ細かく用意しておきたいとき

ノートやバインダーでは、たくさんのテーマを1冊にまとめておこうとすると、全体が大きくなり、軽快には扱えなくなる。逆に細かいテーマごとに専用のノートを用意してゆくと、ノートが非常に多くなってしまうばあいも出てくる。また、2つ以上のテーマに関連する考えが生じたときには、それぞれの場所に同じことを書くのが最も確実な記述であるが、繰り返し書くのはめんどうである。

コンピューターで扱う記憶媒体であれば、思いつくテーマごとに専用のノートを用意する感覚で使うことができる。仮にそのテーマが中断したり不要になっても、カセットテープのように磁性体なので再使用できる。また、2つ以上のテーマに関連する考えが生じたときには、同じ文章を必要な場所にコピーして追加することが簡単にできる。しかし、同じ文章があちこちにいくつもあったのでは、それだけで場所をとってしまい、無駄が生じる。文章をコピーするかわりに、その文章をすぐに参照できる参照項目を置くことにより、文章本体となりに置いたのと同様な効果が得られる（参照項目とその自動的な生成については後述）。

2.2 記述が枝分かれするとき、あるいは他の記述と関連するとき

途中になっていた記述を補うには、流れを続けるためにもとの文章に加える

ときと、もとの文章（の流れ）に含まれない枝として書くときがある。

枝として文章を書くときには方法が2つある。ひとつは、枝をもとの文章と図式的に（線などで）結びつけて表現する方法である。もうひとつは、離れた場所に書いて、もとの文章には脚注のように枝の所在を示す記号、あるいは「第〇〇章参照」のような参照項目を置く方法である。既存の他の文章と関連してくるときにも、参照項目によって関連づけができる。

ノートのばあい、本文の近くに余白があれば、そこに別の文章を書いて図式的に結びつけて表現できる。しかし、離れた場所の記述を関連づけるために参照項目を置けばあい、ノートでは「何が、どこにある」を必ず手で書き、参照には必ずページをめくる必要があるため、枝状に広がる記述や他の部分とつなげたりする記述は敬遠される。

コンピューターで文章を参照するには、その文章の入ったファイル名を指定するだけで良い。文章ファイルの場所はコンピューターが自動的に記録しており、コンピューターが見つけて画面に表示してくれる。また、文章の最初の部分を参照項目の見出しとして用い、必要な場所に参照項目を自動的に作ることが可能である（4.2を参照）。参照項目を選ぶことによって、見たいものは瞬時に表示される。

2.3 いくつかの集め方をしたいとき、順番以外の読み方をしたいとき

ノートでは、全ての内容を日付順に並べ、しかも同時にテーマごとに分けて並べることはできない。ひとつの属性にそって集めることはできても、他の属性についてはバラバラになってしまう。そのわけは、ノートでは集まった文章を直接たどって参照するからである。ノートの中の分散した要素をたどることは、ページを何度も往復することになり、面倒である。かといって、目次や索引を作るのも手間がかかる。

コンピューターで文章をある順番に集めることは、文章ごとの参照項目をその順番に並べ換えて集めることに等しい。日付順の参照項目は日付順のノート、テーマごとの参照項目はテーマごとのノートに等しくなる。文章本体はひとつ

でも、それを参照する項目をいくつでも置けるので、同時にいくつかの集め方が可能である。索引はコンピューターで自動生成が可能である。あるいは、索引がなくても、文字や単語の検索によって、該当する箇所とその周辺の内容を知ることができる。

3. 市販アプリケーションと「ノート環境」の比較

パーソナルコンピューター用で文章記述のための市販アプリケーションにはワードプロセッサ、エディター、アイデアプロセッサ、「ハイパーカード」(米国アップル社パーソナルコンピューターに付属のソフトウェア)などがある。これらと比較することにより「ノート環境」の特徴を説明する。

3.1 ワードプロセッサとの比較

ワードプロセッサは文章をひとつの流れにまとめるための記述環境である。脚注が書きやすく、目次が自動的に生成でき、数字の組み合わせで指定して段落ごとの内容を自由に置き換えることのできるものもあるが、階層の深さは章、節、項など、全体の流れに必要な数段階以内に制限される。

「ノート環境」は記述が枝分かれしたままでも快適に文章が扱えるための環境である。文章や項目の追加訂正には市販エディターを使用している。エディターの中であれば2万字以上の文章を快適に扱える。階層の深さは「ノート環境」の中で10段階を超えても全く問題なく扱うことができる。

3.2 エディターとの比較

エディターはコンピューターのプログラムを編集するための主要な道具である。文章も勿論書けるし、ファイル名と行数を指定すれば瞬時にその場所を表示し加筆できる機能もある。しかし、必ず個別に異なるファイル名を書き、ファイル名で文章を呼び出すことが必要である。MS-DOS (パーソナルコンピューターに広く普及している、操作環境の底辺を形作るソフトウェアで、米国マイ

クロソフト社の登録商標)のばあい、ファイル名の長さは限られており(8字以内)、ファイル名を文章の参照項目として扱うことには難がある。

「ノート環境」で扱う文章も MS-DOS のもとで扱う個別のファイルであるから、ファイル名は不可欠である。しかし、ここでは DOS に必要なファイル名と、文章を選ぶための参照項目を自動的に作り、通常はファイル名を画面に表示しない。したがって、ファイル名を意識する必要がほとんどない。

3.3 アイデアプロセッサとの比較

アイデアプロセッサは文章やチャートなどを階層的にどこからでも書くことのできる環境である。

「ノート環境」はアイデアプロセッサのひとつである。チャートなどの図式を扱う機能は現在備えていない(シンプルな線画であれば描くことができる)。他と異なる特徴として、書いた文章に年月日、時刻を自動的に書き加えることができる。また、ひとつの文章を他のいくつかの文章に関連づけてあるとき、その文章から他との関連をすべて表示させることも容易である。さらに、文章や簡条書の中にプログラム実行文を置くこともできる(「ノート環境」は Lisp と BASIC のプログラミング言語を用いた試作、実行環境でもあり、プログラミングを伴うテーマの記述にも適している)。

3.4 「ハイパーカード」との比較

「ハイパーカード」は書体、文字の大きさ、図形との組み合わせ、画面からの選び方、画面の取り換え方などの表現方法が豊富にあり、データや構造をいくらかでも階層的に構築でき、プログラムも使えるため、非常に広範囲に用いられている。しかし、さまざまな表現の部品を系統的に作ることができるかわりに、それらに名称を与えたりつなぎ合わせる操作が個々に必要である。文章を書けば必ず文書名(ファイル名に相当する)を書くことになる。プログラムは部品のいろいろな振舞を設定するために用いられる。あくまでもシナリオが決まったあとのプレゼンテーションや、定型化した使い方を設定するのに適した

道具であると考えられる。

「ノート環境」では書体、文字の大きさなどに選択の余地がなく、内容の選び方も項目番号または文字入力と限られている。そのかわり、文章名を書く必要がない。相互の関連づけのため手続きもごく少なく、リターンキーだけで済むことがほとんどである。

4. 仕様のもとになる考え

2章に示したコンピューターの基本的な利点をそのまま生かすことを中心に「ノート環境」の操作を設定した。

4.1 選択項目の扱い

文章中の選択項目を選んで次に進むことが、「ノート環境」の最も基本的な操作である。選択項目は置き方によって目次、メニュー、脚注などの役割を果たすことができる（以後、文章を目次、メニューなどと呼ぶこともある）。

- ・選択項目とはそれを選んだときに何らかの作用をする項目であり、文章を参照するための参照項目（4.2を参照）、階層的に連なった内容を持つ簡条書の項目（4.3を参照）、プログラム実行文の3種類がある。
- ・選択項目は前半が見出しで、後半に実効部分を置いた一つのセンテンス（リターンキーを押すまでの文字列で、最大500字以内）である。文章の参照項目のばあい、文章内容を知るための見出しとファイル名であり、簡条書の項目のばあい、補足説明としての見出しと簡条書の項目であり、プログラム実行文のばあい、実行内容の説明とプログラム文である。もちろん、実効部分が見てわかりやすいものであれば、わざわざ見出しをつける必要はない。
- ・文章中に選択項目を自由に置くことができる。
- ・画面表示の際、選択項目には項目番号がつく。見出しつきの項目では、通常は見出しだけが表示される。

- ・選択項目を選ぶと文章や箇条書の参照、プログラムの実行が行われる。

4.2 文章を書くとき参照項目が作られる

参照項目は文章を呼び出して参照するための選択項目である。文章を書くとき「ノート環境」は次のように参照項目を作る（参照項目の例は図2を参照）。

- ・文章の最初のセンテンスを見出しとして扱う。新しい文章を書き終わった直後に年月日、曜日、時刻を文章の見出しにつけ加えることもできる。
- ・新しい文章を書くとき、文章の見出しの写しに日付をそえて参照項目の見出しとなり、8桁の数字（西暦年数の下2桁と月日、それに2桁の通し番号）がファイル名となる。見出しとファイル名を加えて参照項目が作られる。
- ・参照項目は参照もとの文章の末尾、参照項目の一覧表の末尾、階層索引に追加される。

4.3 箇条書の扱い

参照項目の見出しをあとで変更するのは、いくつかの索引の変更も必要になり、ややめんどうである。これに対して、箇条書はあとからどの部分でも追加、変更が容易である（箇条書の例は図4、5を参照）。

- ・箇条書はグループ名とそこに置かれるいくつかの項目から成る。
- ・項目には他の箇条書のグループ名を置くことができる。つまり他の箇条書を階層的に関連づけることができる。階層の深さ、項目の総数は本体メモリー中のユーザーエリアの大きさで制限される。
- ・項目には文章の参照項目、プログラムの実行文を置くことができる。
- ・箇条書はエディターでテキストとして書くことも、「ノート環境」の中で書くこともできる。テキスト上でいくつかの箇条書を正確に関連づけて書くときには細心の注意が要るが、「ノート環境」の中では簡単に追加訂正ができる。
- ・テキストの箇条書を本体メモリーに読み込むと、今までの階層的な設定に追加できる。本体メモリーの設定をテキストに書き出すことも可能。テキスト

では近接した要素だけを断片的に書いておき、あとで本体メモリーに読み込んでから「ノート環境」で整理し、最後に正確なテキストに落とすこともできる。

4.4 本体メモリーの狭さから生じたこと

現在使用しているパーソナルコンピュータでは、本体メモリー中のユーザーエリアが数百キロバイト以内に限られており、書いたものの全部を置くことはできない。このため「ノート環境」では、基本的な初期設定とテーマの根幹に近い部分の簡条書だけを本体メモリーに常駐させて応答性を良くし、それ以外の文章はそれぞれ個別のファイルとして記憶媒体に置き、必要なときに本体メモリーのバッファに読み込むことにした。もしユーザーエリアが広がれば、一度に全部の内容を本体メモリーに置くことができ、全部をひとつのファイルにまとめることもできる。それでも「ノート環境」の操作方法は変わらないはずである。

4.5 最初のファイル名、最初のグループ名

- ・各フロッピーディスクごとに、必ずホームメニュー（B: HOME. TXT というファイル名のメニュー）をひとつ置き、そこから分岐していく。新しいフロッピーに最初にすることは、ホームメニューを書くことである。ただし、ユーティリティ（A: HOME. TXT）でホームメニューの雛型をコピーして使い始めることができる。
- ・簡条書は、元の方で必ず ROOT というグループ名につながる。途中から追加する簡条書のグループ名が ROOT から階層的につながった項目のいずれかであれば良い。

4.6 線画の扱い

キー入力による文字表現を補う意味で、最少限の機能のグラフィックエディター（BASIC プログラム）を呼び出せるようにした（図3を参照）。ただし、画像のデータ圧縮は行っていないため、ほんのわずかの線を描いても、32～96

キロバイトを費やしてしまう。使用は必要最小限にとどめておくべきだろう。特殊な使い方として、数秒間のアニメーションであれば、このグラフィックエディターで手書きし、フロッピー1枚に保存し、再生することができる。

5. ノート環境の使用状況

使用中の主な場面を紹介する。具体的な操作は6章を参照のこと。

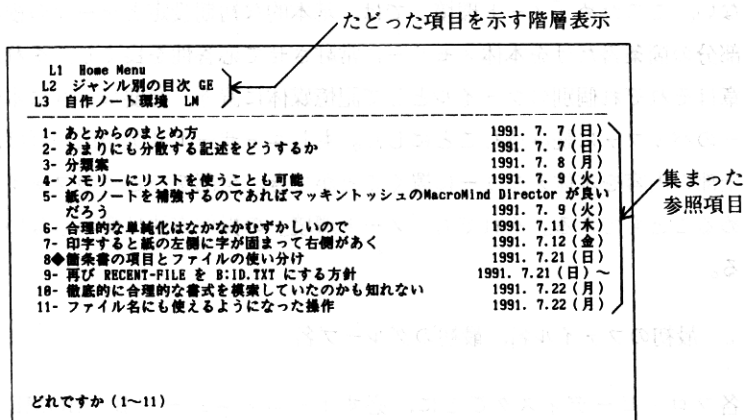


図 2 参照項目が集まると、目次やメニューのようになる

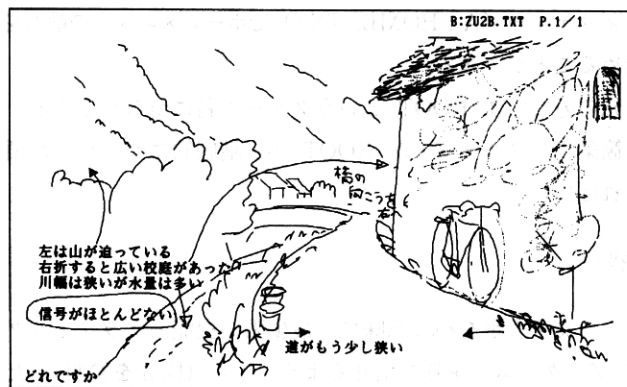


図 3 フリーハンドの線(線画)を加えたメモも書ける

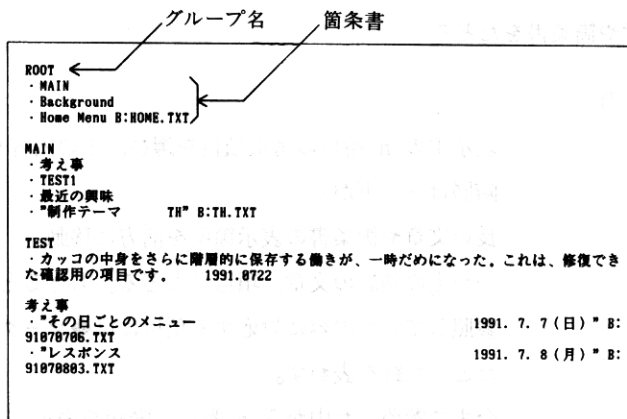


図 4 テキスト上では箇条書が個々に分かれている

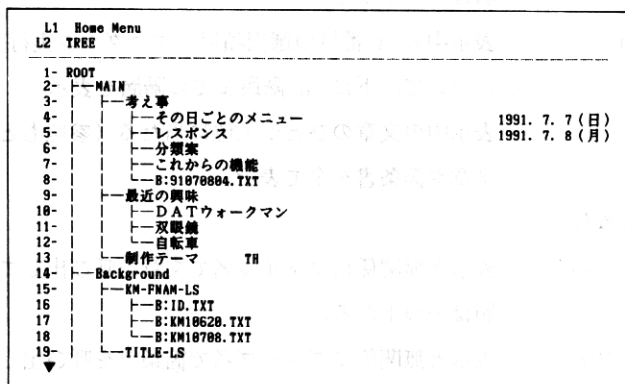


図 5 本体メモリーに読み込んだ箇条書は階層的に扱うことができる

6. 主な操作と表示

初歩的なプログラミングのため、入力操作はキーボード（アルファベット 1～2 文字とリターンキーを打つ操作など）が中心である。なお、以下に出てくる小文字アルファベット n, m は実際には実数、大文字アルファベットは 1 バイトの文字である。

6.1 文章や箇条書をたどる

・基本操作

n	表示中の n 番目の選択項目を選び、その内容を表示。 階層は一つ下がる。
B	長い文章や箇条書の表示箇所を前方に移動。
Q	一つ上の階層の文章、項目にもどる。もどると、今まで参照していた内容に対応する項目には◆マークがつき、たどった跡を表わす。
Ln	今まで参照した中から n 番目の階層を表示。
T	◆マークのついた選択項目を選んでその内容を表示。階層は一つ下がる。
__n m	表示中の n 番目の選択項目（プログラム実行文を除く）について、下に m 階層までの階層を表示。
R	表示中の文章のひとつ上位に当たる（参照もとになる）文章や箇条書を全て表示。

・補足的な操作

ファイル名	表示と無関係にファイル名で文章を呼び出して表示。階層は一つ下がる。
グループ名	表示と無関係にグループ名で箇条書を呼び出して表示。階層は一つ下がる。
QQQ...	Q の数だけ階層をもとにもどって表示。
TTT...	T の数だけ階層を下って表示。
X	もとの階層にある次の選択項目を選んで表示。
BB	もとの階層にあるひとつ前の選択項目を選んで表示。
__項目名 n	指定する項目、ファイル名から下に n 段階までの階層を表示する。
Rn	表示中の n 番目の選択項目のひとつ上位に当たる（参

照もとなる) 文章や簡条書を全て表示。

F 今までに参照した文章の参照項目を一覧

6. 2 文章の枝を作る

・基本操作

LL 新しい文章をエディターで書き始める(新しい枝を作る)。

・補足的な操作

U 既にある選択項目を他の文章の末尾に追加する。

6. 3 文章や簡条書に加筆する

・基本操作

W 表示中の文章をエディターで加筆する。文章中の簡条書の変更は、本体メモリーと食い違いが生じることがあるため、要注意。項目の追加は可。

+ 表示中の簡条書の編集を始める。

・補足的な操作

WW 表示中の文章を最後の部分からエディターで追加訂正。

>n n 番目の参照項目の文章内容をエディターで追加訂正。簡条書は追加のみ可。

>ファイル名 ファイル名を指定してエディターで新規作成。追加訂正。参照項目の一覧表に追加を指示できる。

> 一度ファイル名を指定すると、以後はファイル名を省略しても前項に同じ。

6. 4 簡条書の保存、読み込み

・基本操作

S 本体メモリーの状況を保存するメニューを呼び出し、メニュー選択。

(68)

簡条書の読み込みはホームメニューの1を選択。

6.5 検索（単語や文字列を含む場所を見つけ出す）

・基本操作

- ??文字列 参照項目や簡条書にある文字列を捜す。
- ???文字列 表示中の文章から下層に n段階までの範囲で、文字列をさがす。
- ??? n 表示中の文章から下層に n段階までの範囲で、関連する文章ファイル名、簡条書の項目を階層的に一覧。

・補足的な操作

- ?文字列 Lisp の関数名など本体メモリー中の文字列を捜す。

6.6 線画（文章ごとに1画面）を描く、参照する。

・基本操作

- GR 線画だけを参照。または線画だけを描き始める。
(DSP ファイル名)
線画の加わった文字画面を参照。または文字画面に線画を加える。

6.7 画面表示の選択

・階層表示

文章の表示に先立って、画面の上から順に、最初のホームメニューから現在の文章にいたるまでにたどった項目を表示する。階層表示をさせるかどうかは、文章ごとに指示できる。

・項目番号

通常の文章表示は、選択項目にだけ項目番号をつけるようになっているが、全てのセンテンスに番号をつけることもできる。

・補足的な階層表示

L 階層表示がないとき、ホームメニューから現在までの階層を確かめる。

• 文章中の記号

#文字列 階層的に設定してある言葉（箇条書の項目など）や Lisp プログラム文を#に続けて書けば、その語や式を評価、実行して表示する。

----- マイナス記号だけのセンテンス（左右に空白を含まず3つ以上続ける）を書いておくと、長い文章などの表示をそこで一旦止め、スクロールさせないで次の内容に切り換える。

////// センテンスの中で、表示が不要になった部分の直前（左側）に/を3つ以上続けて書いておくと、そこから後（右側）を表示しない。

• ファイル名を見たいとき

% 文章のファイル名など、表示を省略させた部分を全て表示する。

6. 8 命令語による操作

• MS-DOS と同様の命令語

COPY	ファイル名	ファイル名	ファイルをコピー
DEL	ファイル名		ファイルの削除
DIR	ドライブ名またはファイル名		ディレクトリを参照
REN	ファイル名	ファイル名	ファイル名の変更

• プリンターで印字

TYPE ハードコピーが必要なとき、表示中の文章内容と参照も
との 文章名を印字

• 使うことの多いメニューを略語で参照

A ドライブAにあるユーティリティーのメニュー

(70)

BK	図書メモ
DE	デザイン
GE	ジャンル別の目次一覧
M	メモ
N	ノートの目次
NI	日記
LL-IDX	参照項目の一覧表
など	

6.9 バックグラウンドに用意してあるもの

- ・階層索引 (B : ID. TXT)

選択項目の上下関係を簡条書で保存する。操作 S で消書される。

- ・参照項目の一覧表 (B : LL-IDX. TXT)

操作 LL で書いた文章の参照項目は全てこのファイルに追加される。

- ・ジャンル別の目次 (GE-LS)

GE-LS を書き換えて、各人に合ったジャンルを再設定できる。

7. 考 察

7.1 キー入力 of 文字表示ではむずかしいこと

「ノート環境」はノートの苦手なことを行う目的で制作した環境プログラムであるが、反対に紙に書くときには当然のことがコンピューターではむずかしかったり、ごく初歩的なプログラミングの範囲で仕様を設定したために使いにくくなった部分も少なくない。

- ・書き手のコンディションがかたに残るか

あとから考え事の続きをしたいときに、状況を頭の中に再現するためには、文字の形や1行の曲がり具合など、文章内容とは別の要素も有力な手がかり

になる。しかし、キー入力の手書きと異なり、書いているときのコンディションが文字の形に全く反映しない。清書に限れば、文字が均質化の方が好都合だが、考え中の文章にとって、文字の均質化は必ずしもメリットではない。コンピューターによっては書体や文字の大きさを選べるが、これは清書のための書体であり、書き手の状況表現には至らない。また、「ふさわしい書体を選びながら考えを練る」などというやっかいなことは、誰もしたくないだろう。

- 文章を手直した跡が残るか

キー入力による文章の手直しは、直した跡が残らない。考えの経過記録として訂正箇所を見たいときがあるが、記録できるのは訂正したあとの文章である。訂正前に記録媒体に保存しておき、訂正後の文章と比較することによって訂正箇所を表示させることは可能であろう。

7.2 「ノート環境」はどのように使えるか

記述の枝分かれはどこでも可能であるが、気紛れに書き込むと、あとでたどりにくくなる。記述のつなげ方、集め方にも方針がいくつかあるので、例として紹介する。

- 汎用のノートとしてジャンルごとに書く

フロッピーディスクを汎用のノートにすると、個々の文章をジャンル別の目次を集めてゆくことが中心になる。ちょうど、ルーズリーフのページをジャンル別に集めるような感覚である。フロッピーディスクが一杯になったら次の1枚に移るので、コピーをしない限り、ディスク間で日付の重複は生じない。参照項目の一覧表をたどることで、時系列のノートにもなる。

- 専用のノートとして書く

フロッピーディスクを専用のノートにすると、ジャンル別の目次は、要素別の一覧の働きをする。要素別に書きながら、時系列のノートにもなる。

- 特定のテーマを構成する

テーマの要素を構成するために、箇条書で階層的な項目を作ることが中心になる。文章は項目の下位に置くか、離れて補足的に書くことになる。文章や項目を単に集めるだけではなく、全体の構成の中で削除や再編成を繰り返すときに向いている。

- その日ごとの考えから各部に広げる

事前に蓄積しているものにとらわれることなく、その日の考えを中心にするには、最初に文章を書き、それを一定の目次のもとに必ず加えるようにすれば良い（それぞれの日ごとの入口がひとつの目次に並ぶことになる）。あとは、その文章から分岐したり、他と合流してゆく。

事前に蓄積しているものを扱いながら、その日ごとの微妙な違いを保存するためには、前項の階層的な項目全体を、その日ごとに別個に保存すれば良い。

7.3 今後の課題

実際に使用した感触で不満足な点について、今後の課題としておく。

- 枝分かれした記述の軽重がわかる表現を

ひとつのテーマがあまりに多く分岐したり、ひとつの文章があまりに多方面に関連してくると、最初の考えが補強されるのではなく、かえって見えなくなってしまい、部分のもつ可能性ばかりが表れる傾向になる。その理由は、現在の「ノート環境」ではどの参照項目も同じ扱いであり、幹と枝の太さの違いがないためである。関連性の迷路に迷い込まずに的確にたどれるためには、単につながっているだけでなく、軽重の表現が必要である。

- 考えやすくするために自動設定する項目の工夫を

考えやすくするための工夫として、あらかじめ用意しておいた考慮点を具体的な対象に当てはめ、メニューにするルーチンを作った。対象をいくつか

の要素に分けて考えるための項目が並ぶのだが、このメニューを見ても必ずしも考えを促すことにはならなかった。もっと効果的な考慮点のセット、あるいは別の方法を見つきたい。

8. お わ り に

コンピューターを用いて書くことは、単に文章や項目のままにとどまるのではなく、適切な実行プログラムの生成、呼び出しに近づいてゆくであろう。「ノート環境」を作った目的は、日用のノートをつけながら、考えやすくする環境のありかたと考えを試す道具のありかたを、プログラミングを含めて模索するためである。現在のパーソナルコンピューターはハードウェアも市販アプリケーションも内部を高速化、高密度化させながら、快適さと機能の高度化を洗練させ続けている。これらの中にあってつい忘れがちな、道具の基本的なところからくる可能性について、私はこれからも追求してゆきたい。

