Memo
on the Electron

tukasa

はじめに

Memo on the Electron（MoE）とは、私（つかさ）が開発したメモの方法論である。思いついたアイデアを手帳に書いた後、それをデータ化して、ファイラで管理するのが特徴である。基本的には、梅棹忠夫が京大式カードでやっていたことを PC でやっているだけだが、個々の箇所で重大な変更を施している。本論文は、その MoE の概要について解説したものである。

まずは、この手の技術に不慣れな人のため、知的生産の技術がこれまでたどってきた過程を歴史的に振り返ってみよう。

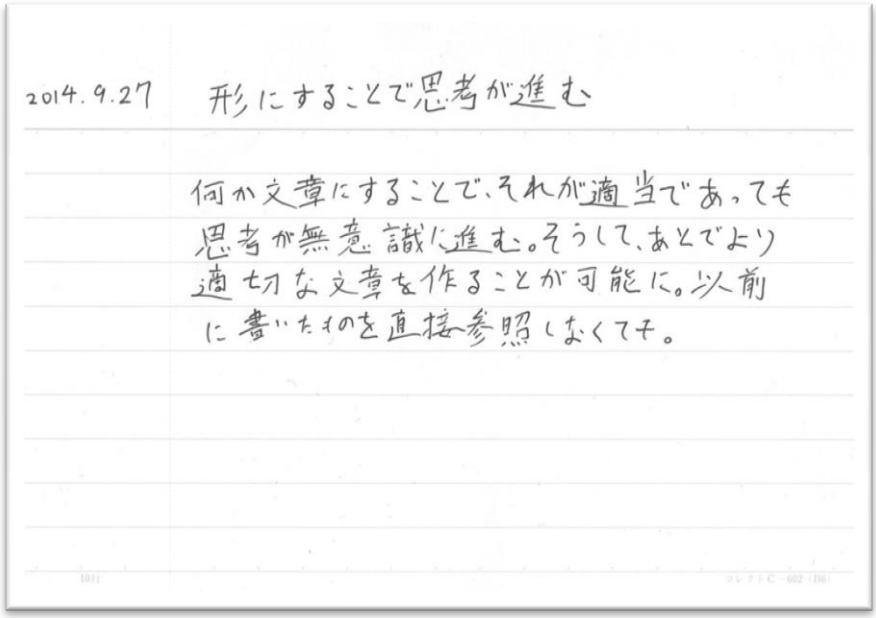
知的生産の技術の歴史

その１ー京大式カード

カードを書く

1969 初版の梅棹忠夫『知的生産の技術』が、この手の技術の走りである。そこでは、京大式カードを使って知的生産を行う方法が記されている。

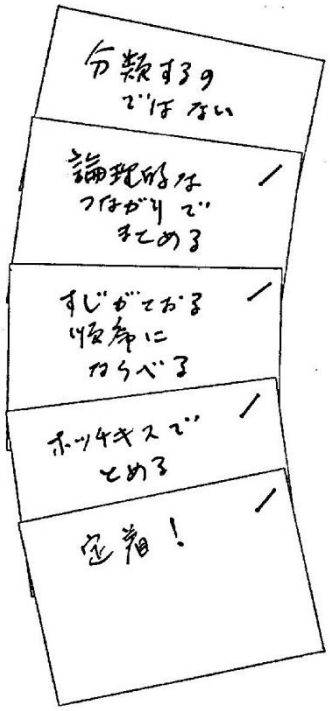
京大式カードの大きさは B6。紙は丈夫で、罫線が薄く引かれている。文房具店で、百枚入りのものが一つ 540 円程度で購入できる。記入の際は、日付、タイトルを必ずつける。一つのカードには一項目の内容のみを記述し、裏は使わない。そして、記入したカードはカードボックスに入れる。



カードを使った知的生産

カードが相当数たまったら、それを「くる」ことによって発想を促す。見返してみたり、いくつか取り出して、見比べてみたりするのだ。そうすれば、思いもよらないアイデアが組み合わされることで、新しい発見をすることができるだろう。つまり、カードを「発想支援装置」として使うわけだ。

また、これは具体的に何か論文を書こうという場合にも使えそうだ。論文を作る時には、とりあえず頭にあるものをすべてカードに書き出す。そうしてから、カードを見比べ、関係するもの同士でまとめる。そうすれば、それは論文の個々の章と対応したものになるだろう。それを参考にしながら、論文を書いていくのだ。つまり、カードは「論文を作る時の素材」としても活用できるわけである（梅棹自身は、カードと別の紙切れを使ってやる方法を提案している。思いつきを紙切れに書き記し、論理的なつながりがあるもの同士を右図のようにホチキスでくっつける）。



梅棹忠夫『知的生産の技術』P204

> 操作できるというところが、カードの特徴なのである。蓄積と貯蔵だけなら、ノートで十分だ。ノートにかかれた知識は、しばしば死蔵の状態におちいりやすいので、カードにしようというのではなかったか。カードの操作のなかで、いちばん重要なことは、くみかえ操作である。知識と知識とを、いろいろにくみかえてみる。あるいはならべかえてみる。そうするとしばしば、一見なんの関係もないようにみえるカードとカードのあいだに、おもいもかけぬ関連が存在することに気がつくのである。そのときには、すぐにその発見をもカード化しよう。そのうちにまた、おなじ材料からでも、くみかえによって、さらにあたらしい発見がもたらされる。これは、知識の単なる集積作業ではない。それは一種の知的創造作業なのである。カードは、蓄積の装置というよりはむしろ、創造の装置なのだ。(梅棹忠夫『知的生産の技術』P57)

アイデアはバラバラのほうが使いやすい

京大式カードの基礎にあるのは、「アイデアは個々バラバラになっているほうが使いやすい」という思想である。そしてこの思想は、一般に用いられている文房具とは潮流を異にするものである。この手の技術に疎い人は、論文を書く時にはノートを利用するわけだ。ノートに思いついたことを書き連ねて、あとでそれを材料にして論文を作ろうとするだろう。だが、そのような試みはたいてい失敗する。ノートは、書き込んだアイデアを組み替えることができない。関係するもの同士を並列したいのなら、新しくノートを購入してそこに書き写すしかない。が、それは非常に手間がかかり、破綻してしまう。ノートには柔軟性がなく、知的生産に向いていないのだ。そこで、アイデアを個々バラバラに書けるようにし、最初から組み替えが可能なようにしているのが、京大式カードを使ったシステムなのである。

ここにたどり着くには、思考作用は技術の問題であるという、醒めた視点が必要になる。自分が考えていることは有限であり、新しいアイデアも所詮は、過去に経験したことの組み合わせから生じるものでしかない。だから、アイデアをそれぞれ別個に書き出して、それを組み合わせれば、一つの思考ができあがる。精神作業とは、突き詰めれば、ただそれを効率化するものでしかない……という割り切った考え方が必要になるのだ。それは没個性的な技術であり、誰でも習得できるものであって、大思想家も

深い思索を積んだ者も、他と区別されないことになるだろう。思考行為に神秘性を認め、そこに独自の価値を見出そうとする人間にとっては、受け入れがたいことであり、普通では到達できない境地なのだ。たいていの人は、自分の知的生産がうまくいかない場合、その責任を自身に帰す。論文が掛けないのは自分の頭が悪く、精神的に未熟だからと反省し、さらに思索を深めるなり、偉大な先達の原典を読み込むなり、といった努力をすることになるだろう。だが、本当はそうでは無いのだ。論文を書けないのは、単に自分のやり方が悪いからである。問題点は技術的な未熟さにある、と総括し、それを突破しようとしているのである。カードを知的生産に使うことは、それゆえそれまでの方法論とは一線を画しており、飛躍なのである。

> 技術というものは、原則として没個性的である。だれでもが、順序をふんで練習してゆけば、かならず一定の水準に到達できる、という性質をもっている。それに対して、研究だとか勉強とかの精神活動は、しばしばもっとも個性的・個人的ないとなみであって、普遍性がなく、公開不可能なものである、というかんがえかたがあるのである。それは、個性的な個人の精神の、奥ぶかい秘密の聖域でいとなまれる作業であって、他人にみせるべきものではない……。
>
> しかし、いろいろとしらべてみると、みんなひじょうに個性的とおもっているけれど、精神の奥の院でおこなわれている儀式は、あんがいおなじようなものがおおいのである。おなじようなくふうをして、おなじような失敗をしている。それなら、おもいきって、そういう話題を公開の場にひっぱりだして、おたがいに情報を交換するようにすれば、進歩もいちじるしいであろう。そういうようにしようではないか、というのが、このような本をかくことの目的なのである。(梅棹忠夫『知的生産の技術』P8)

論文作成とカードの死蔵

このシステムは、取り扱うカードが少ないうちは問題なく機能する。レジュメを作りたい時は、思いついたことなり調べたことなりを、片っ端からカードに書いていき、あとでそれを並び変え、それを材料にして文章を書けばいい。本を読みながら、気づいたこと、思いついたことをカードに書きとめておけば、あとでそれを元にして、その本のまとめを作成することができるだろう。白紙を前に考え込みながら、いきなり文章を漠然と書き始めるよりは、格段に質のいいものが短時間でできるはずだ。

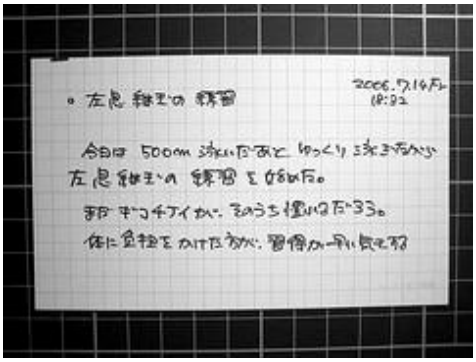
だが、複数の事柄について書かれたカードが、カードボックスの中に大量に蓄積されるようになると、このシステムは機能しなくなる。例えば、ある特定のテーマの論文を書く必要がある時、以前に書いたメモを参考にしようと思って、カードを見返してみるとしよう。すると出てくるのは、そのテーマとは無関係な、個々バラバラの内容について書かれているカードだ。読んだ本のメモ、経験したことの記録、ずっと以前に取り組んでいた課題に関する思いつき……。出会い、経験し、取り組んでいる課題の多様さにあわせて、カードボックスには様々な種類のカードが蓄積されている。そして、カードの多様性に比例して、現在取り組んでいるテーマに関係するカードを取り出すことが、難しくなるのである。さらにここに、時間経過によるメモの劣化という問題が加わる。アイデアは、それを思いついた当初は意味を持っていたとしても、時間の経過によって意味のわからないものになる。カードをめくり、古い方へ行けば行くほど、理解のしがたいカードに出会うことになるだろう。結果、知的生産をするときには、最近作成したカードだけをめくってみて、今の問題意識に関係がありそうなものを取り出すことになるだろう。そして、過去のカードは、過去に遡れば遡るほど、今の問題意識からはかけ離れ、かつ理解することもできないものとして、全く顧みられなくなるだろう。これは、それらを一つのデータベースとしてためていることが、無意味だということを意味する。過去に書いたカードが、死蔵してしまうのだ。

発想支援装置にもならない

さらに、アイデア発生装置としても、このデータベースは役に立たない。たとえば、私が過去に使っていたカードボックスには、当時ハマっていた FPS の特定マップの攻略アイデア、気になったレシピ、何かの思いつきの断片、何かの記録、といったものが混じっている。さらに、過去のデータになればなるだけ、意味の取れなくなったアイデアの欠片に出会うことになり、状況はさらにひどくなる。これらを見比べて、何か意義のあることを思いつけるわけがない。

複数の情報を並列することで発想する。これは、その対象が少なくとも相当練り上げられ、情報として洗練されているものでないと意味がない。アイデア段階でそのまま打ち捨てられた状態のものが並んでいたとしても、それはノイズにしかならないのである。新しい刺激が欲しいのであれば、誰かに話を聞きにいくなり、町に出て新しい経験をするなり、誰かの本を読むなりすればいい。多大な手間をかけて、自分でデータベースなど作る必要は無いのである。

その２ーPile of Index Cards（PoIC）



梅棹の京大式カードの次の段階に来るのが、PoIC である。PoIC は、京大式カードを用いた方法論を独自に体系化したものである。

PoIC では、５×３の方眼のカードを利用する。タイトルの横についているのは、カードのジャンルを分かりやすくするためのアイコンで、これは「発見」アイコンである（他に「記録」「GTD」「参照」アイコンがある）。左上隅の塗りつぶしの位置は、アイコンと対応しており、カードボックスから目的のカードを見つけやすくしている。他にも、カードを持ち運ぶ方法について考察していたり、アイデアを書き留める野帳の使い方についても書いていたりと、考案者の実際の試行錯誤に基づく修正が盛り込まれている。梅棹の著書は具体的な方法論について詳しく論じたものではないので、京大式カードを試してみたい人は、これを参考にするのがいいだろう。

タスクフォース


タスクフォース。左が選り抜かれたタスクフォース。右が残りのカード（拡大する）


グループ化（拡大する）


名付け（拡大する）


空間配置（拡大する）


コンパイル（拡大する）

その中で最も重要な過程の一つに、タスクフォースがある。それは、ためたカードを元に再生産をするプロセスである。まずは広い場所を用意し、そこにカードを並べていく。そして、カード同士を見比べ、似たことについて書かれているカード同士を重ね、グループを作る。そして、その束の上に、グループの内容を記した付箋を貼る。このカードの束を元にして、文書を作る。

文書に内容を反映した束は、お役御免として、カードボックスとは別の場所にしまう。そして、カードボックスには、使わなかったカードだけを戻す。こうして、カードボックスからは利用したカードが消え、カードボックスの中身は、タスクフォース前よりも、秩序だったものになる。

タスクフォースの画期性

その特徴は以下の二点である。

1. 再生産という過程を組み込んだこと
2. カードボックス内の秩序を一定に保つ方法論を提示したこと

これは、梅棹忠夫の京大式カードと似たような方法に見えるかもしれないが、実は思想的に逆を行くものである。梅棹の場合、築くのはデータベースである。カードがたまればたまるだけ、システムの有用度は高まるというのが根本思想なのだ。それゆえ、「利用したカードを取り除く」といった、データベースを破壊する操作は原理的にできない。たとえカードを活用したとしても、それは必ずあとで元の位置に戻すのである。梅棹の思想では、カードによる知的生産がうまく機能しないとしたら、それはカードの枚数が足りないからだ、というように総括されるわけだ。

PoIC は、京大式カードの限界点を、「アイデアから再生産する過程」を意識的

に作り出さなかったことに求めている。カードの死蔵が起こるのは、時間が経過しカードが増えることによって、カードボックスがカオスになるからだ。そこで、カードをまとめ、束ね、廃棄する、というタスクフォースの過程を組み込むことで、カードボックス内の秩序を一定にしようとしている。この点で梅棹の方法論とは別物であり、思想的な飛躍が存在するわけである。

時系列でカードを蓄積していくと、自然の法則に従って、システムの中のエントロピー（情報の乱雑さ）は一方的に増えていきます。分類しない時系列では、なおさらです。このままでは、PoICは破綻しそうにも思えます。私自身、カードが増えるにしたがって、このまま行ったらどうなるのだろうか、と心配になったことがありました。

この自然の法則に逆らってエントロピーを減らそうとする場合、人間の「努力」が必要になります。図書館や博物館では、「つねに分類する努力」によってこれを実現しています。そのために、これらの公共施設では高いコスト（人件費、時間）を払っています。しかし、前述のように、PoICでは積極的に（？）検索・分類しません。では、どのようにしてシステムの破綻を防ぐのでしょうか。

答えは簡単で、やはり検索・分類するのです。従来の方法と違うのは、これが一番最後に来ることです。PoIC において、カードを書くのは、個人の知識のデータベースを構築することです。しかし、これはまだ準備段階です。PoIC の本当の目標は、このシステムを使って、新しい知恵・知識・成果を再生産することです。そうして初めて "Get things Done!" となります。(PoIC - 時系列スタック法)

タスクフォースの限界

私は梅棹の京大式カードで挫折したあと、PoIC に影響されて、タスクフォースの過程を取り入れた。だが、それでもやはり、死蔵という問題は解消されなかった。カードという物理的特性がネックになるからだ。

タスクフォースを実行するには、まず、広い場所を用意しなければならない。気楽に机の上程度の大きさでやるということはできない。これを実現するには畳数枚程度の広さが必要になる。また、この作業は一度に行わなければならない。そうそう何度もカードをすべて調べて並び替えることはできないし、時間が経過すると、どのカード同士が同じ内容なのかの判別もできなくなってしまう。カードがたまるにつれて、負担が増大することになるのだ。

再生産が必要だ、という発想は正しいかもしれない。だがそれは、カードを用いるという前提自体が壁になることで、行き詰まるのである。ここに、それを乗り越えるものとして、PC を用いる方法が要請されることになる。

※画像はどちらも PoIC のホームページから引用

その３ーPC の検索機能を利用

PC を用いて知的生産を行おうという試み自体は、以前からある。アウトラインプロセッサのような、それ専用のソフトは数多く存在するし、howm や ChangeLog のように、エディタの拡張として実装したものもある。私は、その中で、テキスト形式でメモをためる方法を追求した。メモファイルが、特定のソフトでしか開けないものであった場合、後々そのメモが使えなくなることを恐れたからだ。

そこで期待したのは、PC の機能の有効活用である。例えば何か調べ物をしたい場合、Google で適当な検索語を入れれば、自分に必要な情報が乗っていそうなリンクがずらっと表示され、それをいくつか開けばたいていの場合、疑問は解決する。はてなや wiki を読んでいてよくわからない言葉があれば、その語のリンクをたどれば、その説明を見ることができる。ニコニコ動画で興味のある動画に出会ったら、それにつけられているタグをクリックすれば、関連する動画が羅列して表示される。このように、ネットにおいては、データが無数にあったとしても、そこから自分にとって有用な情報を、ＰＣ特有の機能でもって取得する方法が存在する。これを応用すれば、カードで試行錯誤していた「情報が増えることによるカオス化」など、容易に解決するのではないか、と思ったのだ。

移植

だが、その前に、カードでやっていたことを PC に移植することが必要になる。エクスプローラーとメモ帳にしか触れたことが無く、「メモの中身を書くにするために一々ファイルを開くのとか面倒じゃないの？」と思っている人がいるかもしれないので、ここで説明しておこう。

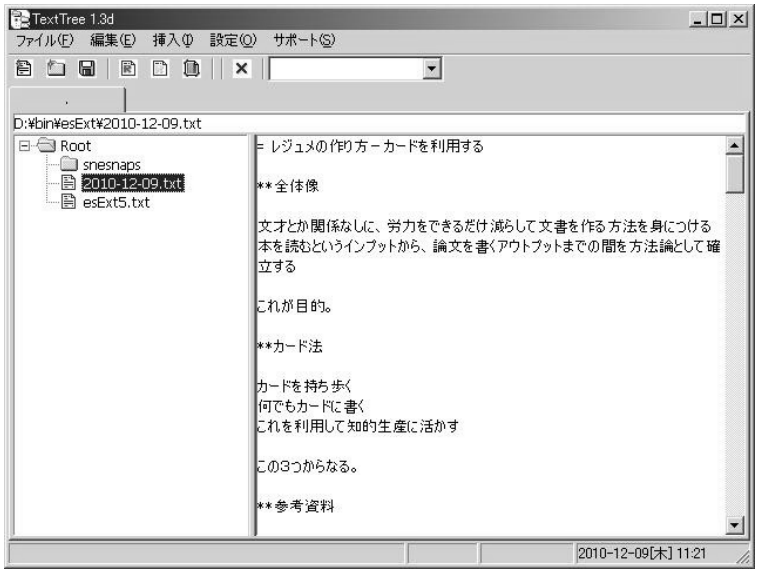
京大式カードにおいては、カードを「くる」という操作を行っていた。この操作は、カードボックスにたまったカードのタイトルだけを見て、関係するものを見つけ出す方法と、カードをめくっていってその本文をざっと見ることで、その内容を把握し関係するものを見つけ出す方法の二つからなっている。このことをＰＣで実現するために必要になるのは

- タイトル一覧表示
- 連動ビュー

の二つである。前者が、「カードのタイトルを見る」に対応する。あるフォルダにメモファイルが入っているとして、各メモのタイトルが一覧表示される仕組みがあればいい。後者が、「カードの中身をざっと見る」に対応する。カーソルをファイルに合わせれば、それと連動して、別ウィンドウでそのファイルの中身が表示されればいいわけだ。

これを実現する方法は、TextTree、howm 等いくつかある。これで、カードで実現していた「最近のメモをざっとみて関係するメモのみを抜き出す」ことは可能になる。

TextTree によるメモの管理



Grep による関連メモの並列

そこで本題の「PC の機能を利用して、データベースから関連するメモを抜き出す」に移る。これの追求で、私は立ち止まった。

それには

- Grep の利用
- 関係するファイルに共通のタグを入れる
- メモデータ本体とは別に情報管理

の三つの方法がある。

まずは Grep について話そう。Grep とは、複数のテキストファイルを対象にして、特定の単語を検索する仕組みである。これは、エディタの機能としてついていることが多く、また Grep のみに特化したソフトも存在する。だが、Grep を利用して自分が今知りたいテーマについての情報を並列するのは難しい。そもそもテーマに関係する単語が存在するのかという問題がある。また、たとえあったとしても、それによってアイデアメモの内から関連するものをすべて拾える保証がない。それに、この方法だと、どうしても関係ないメモがヒットしてしまう。

この方法は、Google で調べものをした際の類推から、有効なものであると思われている。例えば何かを調べものがあるとき、我々は Google で、それに関係していそうな単語を検索する。すると複数のサイトがヒットするので、それを上から順に開いていく。そうすると、たいていは知りたいと思った情報に行き着くわけだ。

だが、Grep で検索する対象は未完成なアイデアであり、その点が Google 検索の状況と、大きく異なっている。Google 検索をかける対象は、完成したデータである。だから、ヒットしたリンクには、知りたい情報が載っている可能性が高い。しかし、その対象がアイデアメモの場合、ヒットするのはすべて、未完成な情報である。それ単体では無意味なものでしかなく、それをいくつ閲覧したところで、知

りたいことが書かれていたりはしないのだ。さらにそれは、過去のものになればなるほど、それを書いた際の問題意識の忘却から、内容が理解できないものになる。それだけならまだしも、全く現在の問題と関係の無いメモまで同時にヒットしてしまうのだ。だから、Grep によって自分の求める情報が手に入る可能性は低い。

VxGrep で「メモ」という語を検索

関係するファイルに共通のタグを入れる

次にタグ付けについて。この場合、ファイル検索でヒットするよう、次のようにタグをつけることになる。

【タグ】ファイル名.txt

重要度をスターなどで表す場合も原理的にはこれと同じである。例えば、ファイル名の末尾に、特別な記号をつける。

ファイル名☆☆☆.txt

このようにして、タグなり記号なり、共通の単語をファイル名に入れる。そうすれば、その単語でファイル名検索をした時、関連するファイルを一覧表示できるわけだ。

このときの問題点の一つ目は、持続性を持つタグをつけることができない、ということである。何かを思いついてアイデアメモを書いたとしても、それを思いついた当初は、それが何についてのアイデアか、というカテゴライズができていない場合が大半である。今浮かんだアイデアに、現時点の判断で、将来までずっと通用するようなタグを付けるということは、ほとんど不可能だろう。だからどうしても、おおざっぱな区分か、あるいはそのときにたまたま思いついたタグを付けることになる。

「おおざっぱなタグ」は、おおざっぱなゆえに役に立たない。そのときどきの問題関心は移り変わる故に、同じタグであっても、メモ同士の関連性は薄いものになる。メモを利用するには、自身の問題意識に則した細かな区分が必要になるのだ。

「そのときに思いついたタグ」は、持続性が無いゆえに役に立たない。ある日思いついたアイデア A

に、思いつきで「ほげほげ」とタグを付けたとしよう。しかし、次にそのアイデア A と関連するアイデア B を思いついた時、「前に似たようなアイデア思いついたよな」ということは覚えていたとしても、それにつけたタグを正確に思い出すことができない。そうして、そのときたまたま思いついた、また別のタグを付けることになってしまう。しかし、Grep を利用するからには、タグ名は正確に一致していなければならない。だから、このようにして付けたタグは無駄になる。

また、タグを付けても、重要度での区別をつけることができないという問題がある。メモの中には、他の複数のメモを参考にして書いた、完成度の高いメモと、ただ思いつきを書いただけの、完成度の低いメモが混在している。しかしそれらは、タグを利用して表示した時、すべて同じように表示されてしまう。有用なメモが、他の雑多で低価値のメモの中に埋もれてしまうのである。同じタグを持つファイルが増えれば増えるほど、この事態は顕著になる。結局、カードボックスの中で、有用なカードが大量の他のカードに埋もれて死んだのと同じように、同じタグを持つファイルの中で、有用なメモファイルが死んでしまうことになる。

データ本体とは別に情報管理

次に「メモデータ本体とは別に情報管理」について。カード型データベースソフトを使い、一つのデータに対し一つの蔵書カードを作成。検索その他はこの蔵書カードを利用して行う。気になって調べはしたが、適当なカード型データベースソフトを見つけられなかったという理由であきらめる。

そして破綻へ

結局、メモをデータベース化して PC 特有の機能で操作する、という試みは破綻した。結果、最近のメモのみを一覧表示し、関係するメモをいくつか見比べる、という操作のみをすることになった。これは、最近作成したメモ以外はあってもなくてもいいということ、すなわち過去のメモ作成にかけた労力が無駄になったということを意味する。カードと同じ問題が、ここでも生じたわけだ。

その４ーデータベースからバッファーへ

野口悠紀雄のバッファー理論

PC に特有の機能を使っても、問題は解決しなかった。いくらメモをためたとしても、それを有効に使う方法はない。データはやはり死蔵してしまったのだ。既存のツールを使うだけではだめだ。知的生産を実現するためには、この状況を突破する必要がどうしてもある。

その突破のきっかけになったのが、野口のバッファー理論だった。これは、野口悠紀雄『「超」整理法〈3〉』において出てくる理論である。

知的生産活動で扱うデータは、決まりきった仕方では処理できない、という特性を持っている。それは、最初はどのカテゴリーに入るかも不明で、重要度も分からない。それを扱う方法に先例もモデルも無い場合が大半だ。このような性質を持つデータに対しては、処理が確立されている、定型的な仕事でとられてきた方法は使えない。今までとは異なる発想に基づく処理システムを用意し、そこで処理をすべきである。野口はこのような考え方をする。

> まず、対象がフロー（流れ）であることを明確に意識する必要がある。必要とされるのは、膨大な量のフローを制御するダイナミック（動的）な方法である。一定量のストック（蓄積）を管理するスタティック（静的）な対処法ではない。
>
> これは自明のことである。しかし、従来の収納システムは、「内容がほぼ変わらないストックのための管理」のためのものだ。「大量のものが流入し、ストックの内容が短期間のうちに入れ替わってしまう」という認識は、殆どないのである。(野口悠紀雄『「超」整理法〈3〉』P21)

ではどうするか。それには、「とりあえずおいておく場所（バッファー)」を用意し、そこに分類せずにデータを入れればよい。ある程度の時間がたてば、そこに放り込まれたデータは自然に醸成し、区分も判明になるだろう。そのときになってはじめてそのデータを取り出し、分類して、処理をすればいい。

> マゼラン的な仕事を扱うには、マニュアル遵守的な仕事とは異なる発想にもとづいて、処理システムを構築する必要がある。
>
> とくに重要なのは、「バッファー」(buffer)だ。これは、「緩衝器」、つまり二つのもの、あるいは二つのプロセスの中間にあって、衝撃を受け止めるための装置である。
>
> まず、外から入ってきたものや新しく作ったものを、簡単な手続きによって（できれば、殆ど手間をかけずに)、システムの中に受け入れる必要がある。整然とした収納でなくともよい。しかし、書類が紛失したり迷子になったりすることはないようにする。つまり、「とりあえず受け入れる」のである。これが、「受け入れバッファー」だ。
>
> そして、不要と思われるものを必要なものから区分し、所要の措置や加工などを行い、次の段階に送る。
>
> この際、確立された処理法はないのだから、本当に正しい処理をしたかどうかは分からない。やり直す必要があるかもしれない。そのため、完全でなくともよいから、一応の措置をする。一〇〇％処理を目論んで何もしないのではなく、とにかく一歩進める。「ゼロか完璧か」でなく、八割の処理をするのだ。これは、単なる先送りとは違う。「もっとも重要と思われること」は、行っておくのである。
>
> そして、「多分必要ないだろう」と思われるものを、日常の仕事のジャマにならないようなところに置く。つまり「とりあえず捨てておく」のである。これが、「廃棄バッファー」だ。(野口悠紀雄『「超」整理法〈3〉』p41)

バッファーによる解決

アイデアメモが使えなくなったのは、メモが処理しきれなくなるほどたまり、古いデータがノイズになって、全体が機能しなくなったからである。そこで、発想を転換する。最初にためるのはデータベースではなく、バッファーだ。そこに入れたデータは、一時的に保存されているだけである。時間が経過して、それらのアイデアメモがどのような問題意識によって書かれたものかが判然としたなら、それらをまとめて別の場所に移動し、その上で知的生産作業を行う。分類は、後でするのだ。

- 最初は区分せず、とりあえず入れる場所を用意してなんでもそこに放り込んでいく
- 半自動的にそれが保てるシステムを構築する

この二点を実現すれば、不定型なデータの処理が可能になる。

今までアイデアメモをうまく活用できなかったのは、データベースの規模の問題でもなければ、データベースの運用方法に通じていないからでもない。前提としていた、データベースを作る、という発想からして既に間違っていたのだ。個々のものの重要性がはじめからわかっており、カテゴライズがすでになされているようなものであれば、データベースを築くことは有効である。それは、規模を増せばそれだけ、有用度を増すだろう。だが、アイデアのような不完全なデータを大量にためても、無意味なのである。それをもとにしてデータベースを築いたとしても、それは個々バラバラのスピードで陳腐化していく。そしてそれにつれ、データベースはカオス化し、使いものにならなくなる。この問題は、データベースから有用なデータを抜き取る機能をいくら追求したとしても、決して解消されないだろう。それよりは、バッファーが半自動的に洗練され、不要なものが削ぎ落とされ、常に自身にとって有益な情報がある状態を保ち続けることのほうが、重要なのである。

このことに気づいていなかったから、PC 上でアイデアメモを管理する試みは失敗していたのである。そこでは、どうしてもデータベースを否定するような発想、例えば「過去のデータを編集する」「不要なデータを消す」「不可逆的な仕方で特定のグループに分割する」といったことができなかった。そして、カードを有効活用するための試みも、タグ付けといった表層的な操作に限定されていた。我々は、既存の思考に縛られすぎていたのだ。ノートを捨ててカードを使い出した時のように、データベースを捨て、バッファーを中心とした方法を構築する必要がある。

MoE の原型

データベースを作る必要はない。だから、メモが不要になればそれを捨ててしまってもかまわない。自由にデータを組み替え、まとまりを作り、別の場所に移動させ、原本を改変しても全くかまわない。とにかくずっと、全体が有用であり続ける状態をつくれ……。この発想を元にして、MoE の原型はできあがった。あとは、これを PC で実現すればいいわけだ。

1. メモの受け入れバッファーをはじめから用意する。これは、分類だとか重要度だとかそういったことをいっさい気にせず、放り込める場所にする

→これまではデータベースとして扱っていたフォルダを、バッファーとして捉え直せばいい。だから、「思いついたメモをすべてそのフォルダに入れる」ということ自体は変える必要はない。ただ、データベースでは無くなったのだから、別フォルダへの移動も、メモの編集も、自由に行っていいようになる。

2.　それが秩序を保った状態を維持する

→フォルダから関連するメモを取り出し、別フォルダに移動することによって、実現する。このプロセスは、関連するメモ同士のまとまりを作るという意味だけではなく、バッファーの秩序を保つという意味も持つだろう。

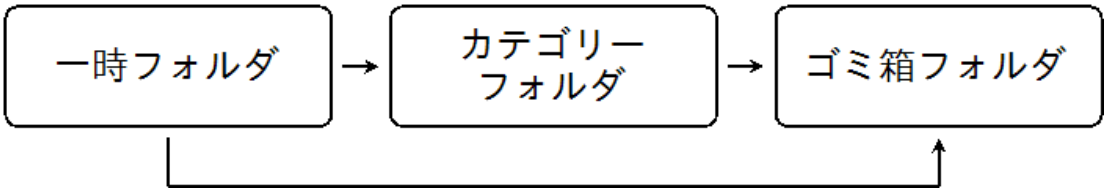
また、メモを廃棄するためのフォルダをあらかじめ用意しておく。不要だと判断したメモを容易に捨てられる状況を作ることは、全体の秩序を維持することにつながるだろう。

フォルダ構造とバッファー

では次に、これをフォルダ構造と、フォルダ移動の観点から見てみよう。必要なのは、

- 一時フォルダ……最初にアイデアメモを保存するフォルダ
- カテゴリーフォルダ……カテゴリー分けのためのフォルダ
- ゴミ箱フォルダ……不要なメモを捨てるためのフォルダ

の３つのフォルダである。メモを順番に移動させることにより、常に有用なメモのみを利用できるようにする。



メモの移動だが、次のようになる。

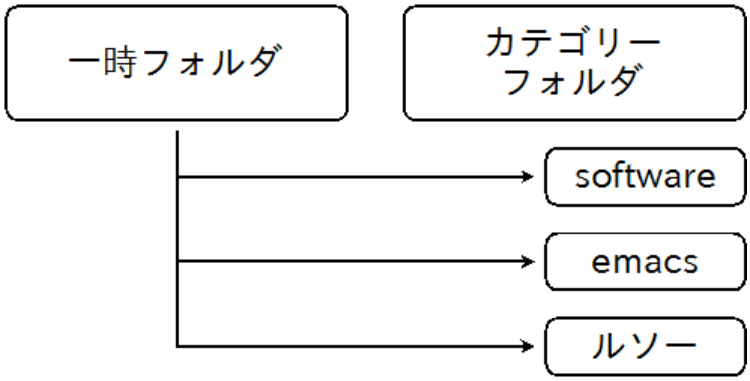
1.　一時フォルダにためる

思いついたメモは、どれでもまずは一時フォルダに保存。思いついた順にここに無差別に入れていく。メモの種類も、カテゴリーも、以前に書いたメモとの関係性も一切気にせず、どんどん思いつくままに作っていこう。

２．　一時フォルダからカテゴリーフォルダに移動

時間が経過すれば、各々のメモが、どのような問題意識で書かれたものか明確になってくる。そうしたら、そのとき初めてカテゴリー分けを行う。カテゴリーフォルダ※に、新たにフォルダを作成。一時フォルダからそのフォルダへ、関係するファイルをまとめて移動する。個々のメモの内容は、タイトル表示とビューアで確認すればいい。

※後述するが、実際の運用では、一時フォルダの下に個々のカテゴリーフォルダを作るようにしている。



こうして、カテゴリーフォルダの中に、個々のカテゴリーフォルダを作っていく。後は、各々のフォルダへ行ってメモを閲覧しながら、知的生産作業をすればいいわけだ。

3．ゴミ箱フォルダへの移動

一時フォルダ内やカテゴリーフォルダ内のメモの内容を、ビューアで閲覧している時に、不要になったメモが見つかることがあるだろう。例えば、すでに作成した文書にその内容が反映されているメモや、状況の変化によって、すでに問題意識自体が古くなっているメモだ。それを見つけたら、その都度、ゴミ箱フォルダにそのメモを移動する。フォルダ単位で不要だと判断したものがあれば、それも同様に、ゴミ箱フォルダに移動する。

時間軸という観点から

これが利用しているのは、「時間がたてば書き出したメモも醸成され意味がわかってくる」という原理である。人間はそのときどきでいろいろな問題に取り組んでいる。それは突発的なものであったり、あるいはずっとかかえているものであったりするだろう。それに対しての対処法、方針、総括、感想といったものが、そのときどきにたまたま経験することをきっかけにして、無秩序にひらめく。ただ、それがどのような問題に関してのものなのか、といった全体的な見通しは意識されないことも多いわけだ。そうして、そのときにたまたま思いついたことを元にして、何かの行為をするわけである。それは、偶然によって失敗するか成功するか、するだろう。だが、すぐにそのことについては忘却するわけだ。そうして、何度も同じ思いつきを繰り返し、偶然的に行為をし、また忘却して、というように、同じところをくるくると回ることになるのである。

そこで、思いついたことをメモとして蓄積する。それ自体は、はじめは単体としてみればたいして価値のないものだろう。だが、時間の経過に伴い、それがどのような問題意識からきたものなのか、そしてそれをどのように解決しようとしているのか、といった判断がだんだんとつくようになる。関係するメモを取り出し、カテゴライズし、見比べ、それを実際の行動に反映させれば、思いつきでやっていた時よりも適切な行為ができるようになるだろう。

我々は現実において様々な問題に出会い、そのときどきでなんらかの対処をする。そのような実際の流れとは別に、それらを一旦、メモ群という別の場所にとりあえずおいておき、寝かせて、それを再び現実に反映させる。リアルの流れとはまた別の時間の流れというのを、別の場所に平行して作り出すのである。

まとめ

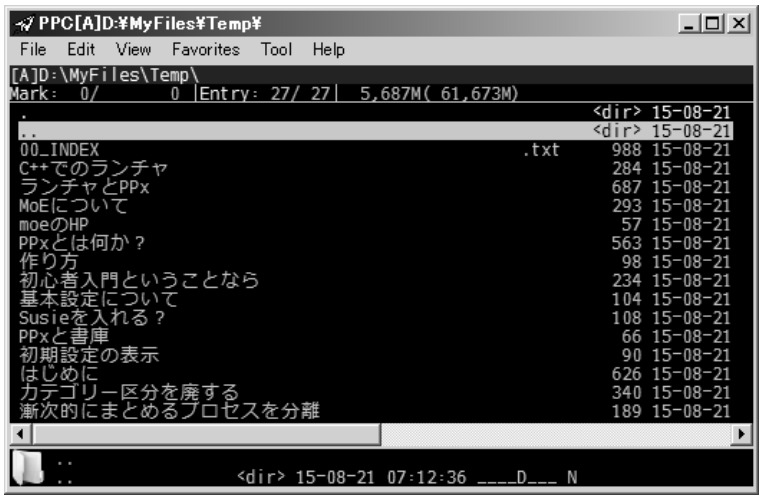
バッファー理論に基いて、死蔵しないシステムを構築するには、メモファイルの柔軟な移動が必要になる。そしてそれを実現するには、任意の場所へのファイル／フォルダ作成、フォルダ削除、リネーム、ファイル移動といった操作が必須になるわけだ。「その３―PC の検索機能を利用」までは、まだ単一のソフトウェアを作るだけで用は足りた。しかし、この段階になると、ファイラと同等の機能が要求されることになる。こうして、PC を用いたメモシステムは、その要件をさらに限定されるわけだ。それは、「連動ビューとタイトル表示を可能にするもの」であり、かつ「ファイラをベースにしたもの」でなければならないのである。そして、この条件のもとに私が築いたシステムが、MoE である。

MoE のベースにあるのは、京大式カードの思想（アイデアは分離して組み換えられたほうが便利）とバッファー理論（データベースの否定）である。それを、私が元々持っていた、ファイラについての多少の知識によって実現したわけだ。死蔵という問題を乗り越え、知的生産を実現するためには、ファイラの利用が必須になる。単一のソフトを開発すれば解決できるというわけでもなく、運用するためには、ある程度の思想が必要である。さらには、ファイラの操作にある程度まで親しんでいる者でなければ実装できない。以上のことが、これまで、PC で知的生産を実現することのネックになっていたのではないかと私は思っている。

MoE 概観

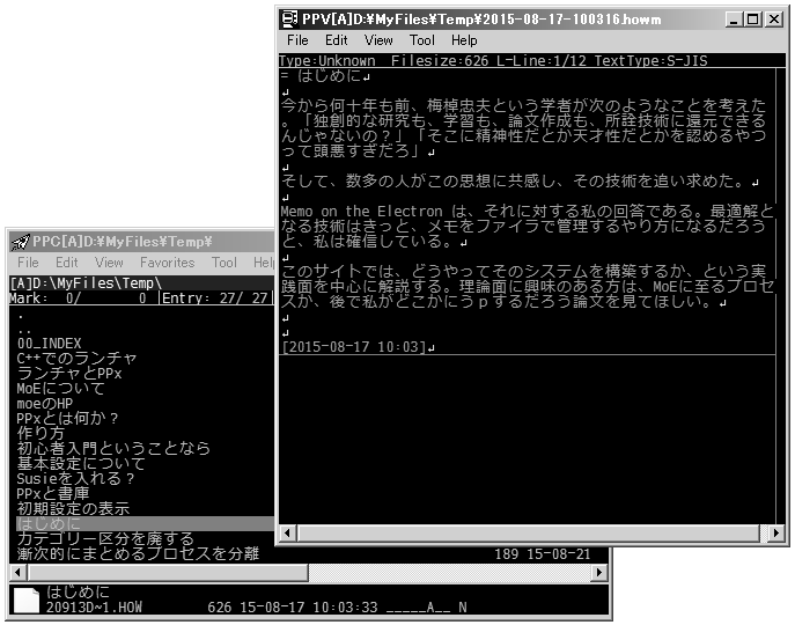
実際に行う操作は、次のようになる。

テキストファイル形式でメモを書き、それを特定のフォルダにためていく。手帳に思いついたことをメモっておいて、あとでそれを元にエディタで文章を書き、保存する、ということになるだろう。



それがある程度までたまったら、カテゴリー分けをする。ためたデータをビューアで閲覧しながら、関係するもの同士を一つのフォルダにまとめていく。

そののちに、個々のカテゴリーフォルダ内で、メモをビューアで閲覧し、何らかの知的生産を行う。また、その過程で不要なメモを見つけたなら、適宜ゴミ箱フォルダへと移動する。



MoE の実装

では、実装する方法について具体的に説明していこう。ちなみに、MoE のホームページ（ https://sites.google.com/site/moesystem/ ）にもほぼ同じ内容のことが書かれている。コードのコピーやスクリプトのダウンロードは、そちらを参考にして行ってほしい。

事前準備

まずは、ファイラとエディタの設定をする。

ファイラ（Paper Plane xUI）

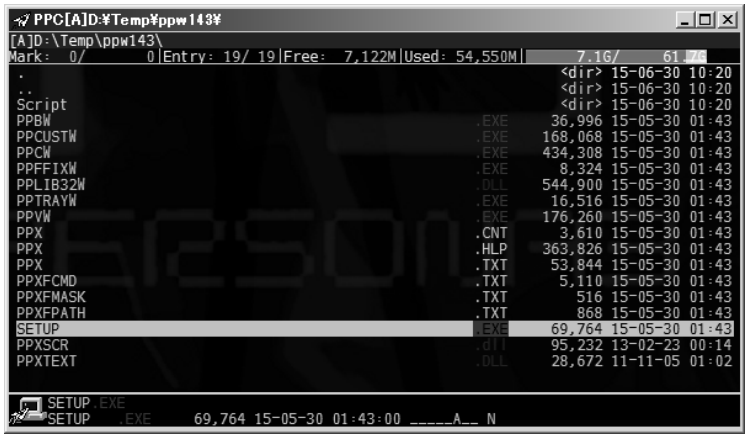
TORO's Library (http://homepage1.nifty.com/toro/index.html) に行く。そこで

- PPx 本体
- PPx Script Module
- PPx Text Module

をダウンロード。続いて

1. PPx 本体を解凍
2. PPx Script Module を解凍し、中にある PPXSCR.dll（64bit 版なら PPXSCR64.DLL）を PPx フォルダにコピー
3. PPx Text Module を解凍し、中にある PPXTEXT.dll（64bit 版なら PPXTEXT64.DLL）を PPx フォルダにコピー
4. PPx フォルダ内に、Script という名前で新しいフォルダを作成

以上で、PPx フォルダは次のようになる。

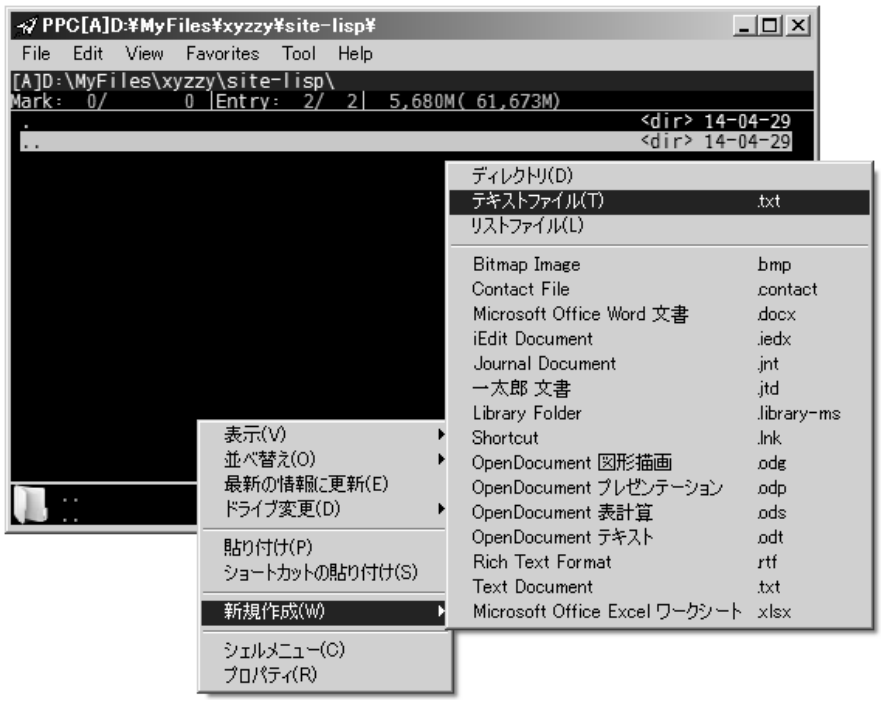


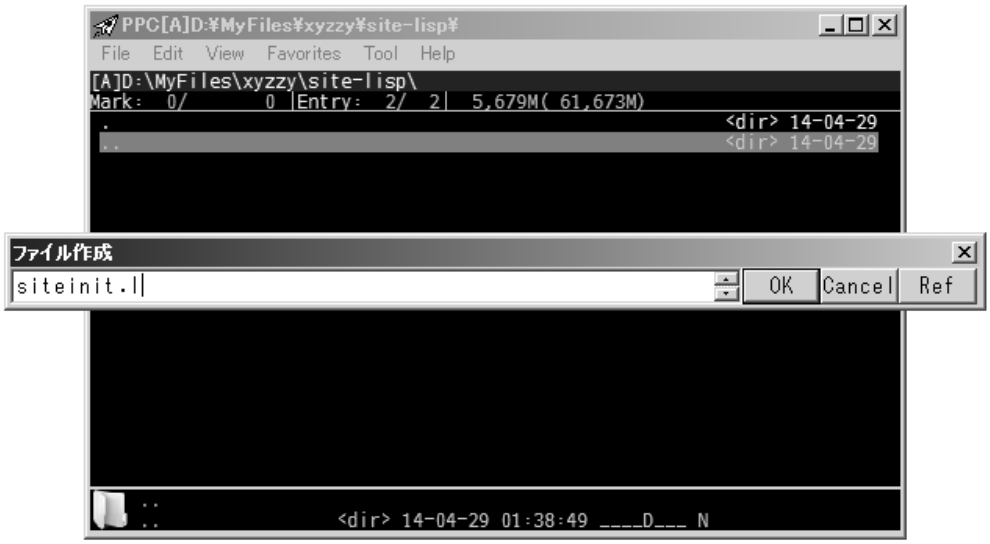
次に PPx フォルダにある SETUP.EXE を実行。セットアップ方法は「おまかせインストール」を、インストール先の選択は「コピーなしでインストール」を選択。後は適当で大丈夫だ。

エディタ（xyzzy）

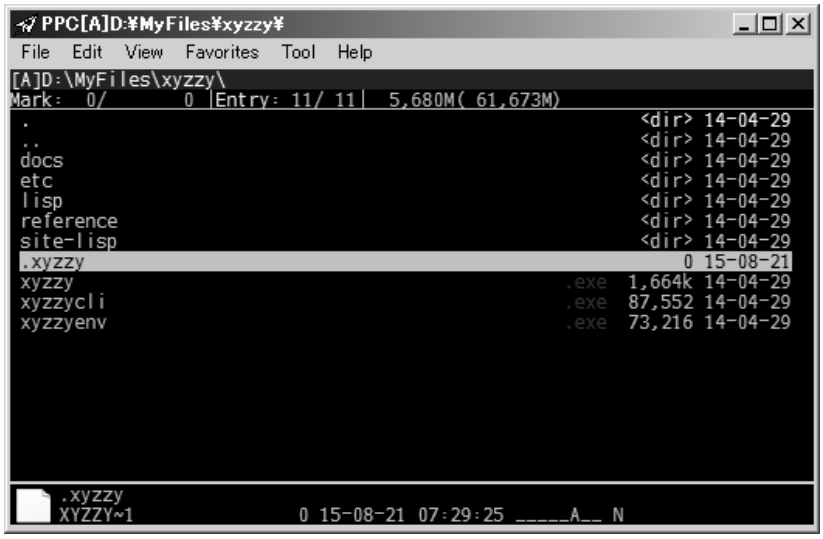
github 版 xyzzy 配布サイト（ http://xyzzy-022.github.io/ ）に行く。そこで xyzzy をダウンロードし、解凍する。

次に、xyzzy フォルダ内の site-lisp フォルダに siteinit.l を作成する。何も無いところを右クリックし、「新規作成 - テキストファイル」を選択。siteinit.l と入力し、OK を押せばいい。





同じ要領で、xyzzy フォルダに「.xyzzy」を作成する。



ついで、環境変数の設定を行う。コントロールパネル→システムとセキュリティ→システム→システムの詳細設定→環境変数→ユーザー環境変数の新規をクリック。変数名に XYZZYHOME を。変数値に xyzzy フォルダのパスを入力して OK を押す。

MoE 用設定

ここからは、MoE を行うために必要な独自の設定である。以下の５つを順番にやっていけばいい。

１． howm-create2dir.l（xyzzy）

howm ファイルを作成するスクリプト

howm-create2dir.l

を xyzzy の site-lisp フォルダに入れる。

２． xyzzy の設定ファイル

xyzzy フォルダの.xyzzy をエディタで編集し、以下を追加する。PPx では、エントリにカーソルをあわせて E を押せばデフォルトのエディタで、Shift+E を押せば PPe で編集することができる。

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;howm モード

(pushnew '("\\.howm$" . howm-create2dir-mode) *auto-mode-alist* :test #'equal)
(require "howm-create2dir")
```

３．MakeComment.js（PPx）

PPx の Script フォルダに、次のファイルを保存する。

MakeComment.js

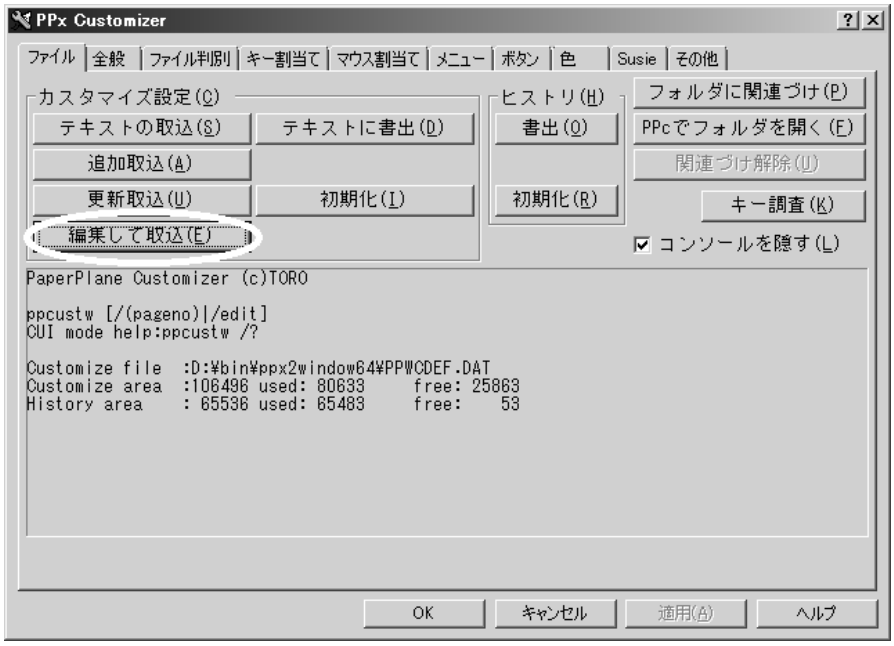
MakeComment.js は、タイトルとファイル名とを対応させたコメントファイル(00_INDEX.txt)を、フォルダ内に自動で作成するスクリプトである。

４．カスタマイザーで編集して取込（PPx）

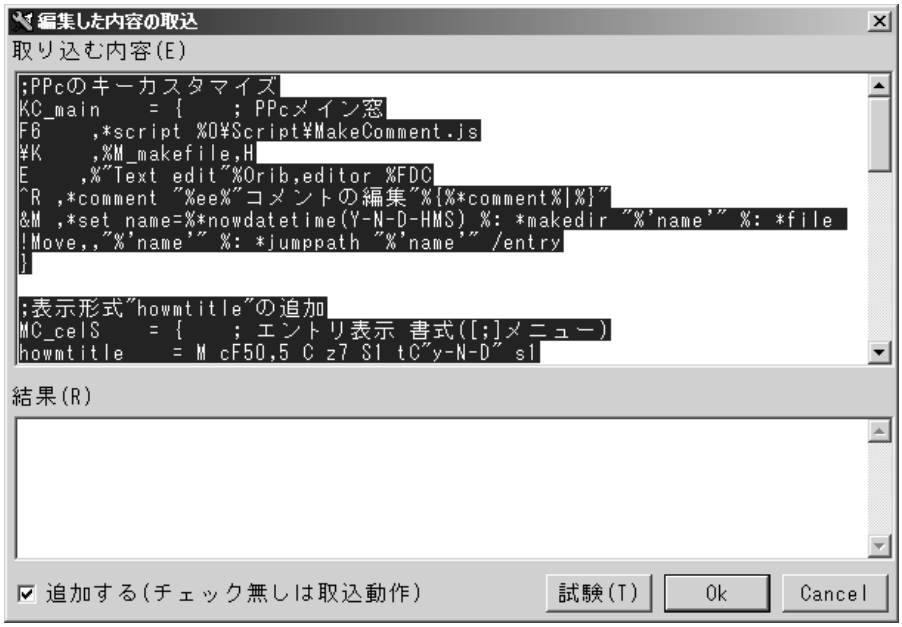
以下のコードを、クリップボードにコピーする。

```
;PPc のキーカスタマイズ
KC_main      = {      ; PPc メイン窓
F6      ,*script %0\Script\MakeComment.js
\K      ,%M_makefile,H
E       ,%"Text edit"%Orib,editor %FDC
^R ,*comment "%ee%"コメントの編集"%{%*comment%|%}"
&M      ,*set      name=%*nowdatetime(Y-N-D-HMS)      %:      *makedir      "%'name'"      %:
*file !Move,,"%'name'" %: *jumppath "%'name'" /entry
}

;表示形式"howmtitle"の追加
MC_celS      = {      ; エントリ表示 書式([;]メニュー)
howmtitle      = M cF50,5 C z7 S1 tC"y-N-D" s1
}

;拡張子が howm の時に Enter を押すとビューアが起動
E_cr      = {
HOWM      ,%v "%1\\%R"
}

;PPv のキーカスタマイズ
KV_main      = {
UP      ,%KC"@UP@N"
DOWN      ,%KC"@DOWN@N"
LEFT ,*execute C,*cursor 6, -1 %: %KC"@N"
RIGHT ,*execute C,*cursor 6, 1 %: %KC"@N"
SPACE      ,%KC"@SPACE@N"
\SPACE      ,%KC"@\SPACE@N"
ENTER      = @Q
}

;howm 作成用メニュー

-|M_makefile =

M_makefile      = {      ** comment **
&howm-create2dir = %Ob,xyzzy -e (howm-create2dir) %1
--      =
ディレクトリ(&D)      = *makedir %1\%E
}

XC_cwrt = 2 ; コメントファイル作成は 0:手動 1:確認有 2:自動
```

PPx の上部にあるメニューバーから「ツール ― カスタマイズ」を選択するか、PPx フォルダの PPCUSTW.EXE を実行するかして、カスタマイザーを起動。「編集して取込」をクリックする。



すると、「編集した内容の取込」ウィンドウが、クリップボードの内容をペーストした状態で表示されるので、OK ボタンを押す。これで、このコードを取り込むことができる。



5．エディタの設定（PPx）

カスタマイザーを開き、その他タブのエイリアスから、editor のパスを xyzzy のパスに変更し、追加ボタンを押す。

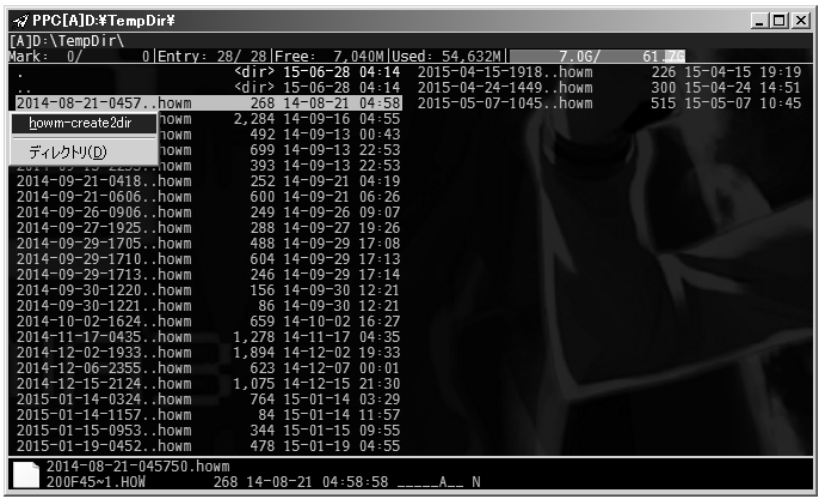
次に、エイリアス名に xyzzy、内容に xyzzy のパスを入力して追加を押す。上の欄に追加されたのを確認したら、適用ボタンを押す。

基本操作

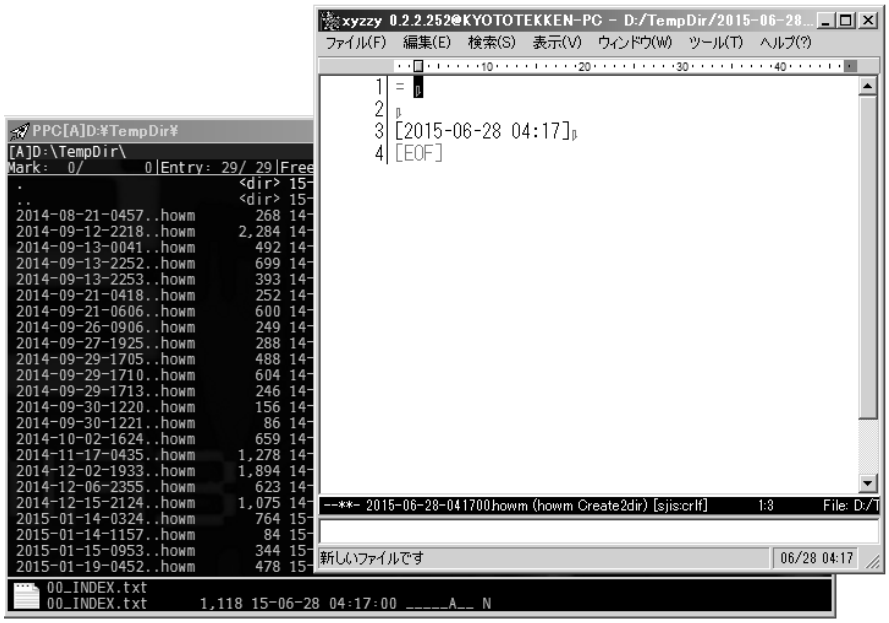
まずは、MoE の基本操作を確かめてみよう。

1．メモの作成

適当なフォルダで Shift+K を押すと、ファイル作成メニューが出るので、howm-create2dir を選択。



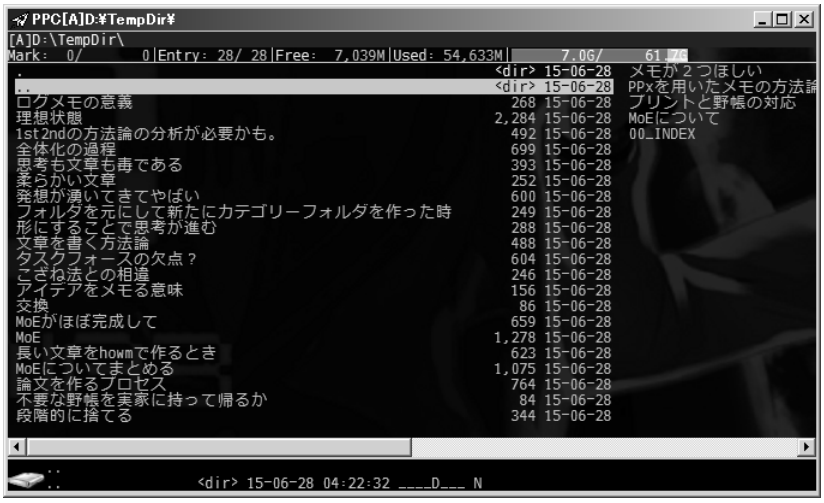
すると、howm 形式で xyzzy が起動する。一行目にはタイトル、それ以降には本文を書く。ここで作成したファイルは、カレントフォルダに保存される。



２．タイトル一覧表示

メモファイルがあるフォルダで、F6 を押すと、メモファイルの一行目とファイル名とを対応させたファイル「00_INDEX.txt」が作成される。

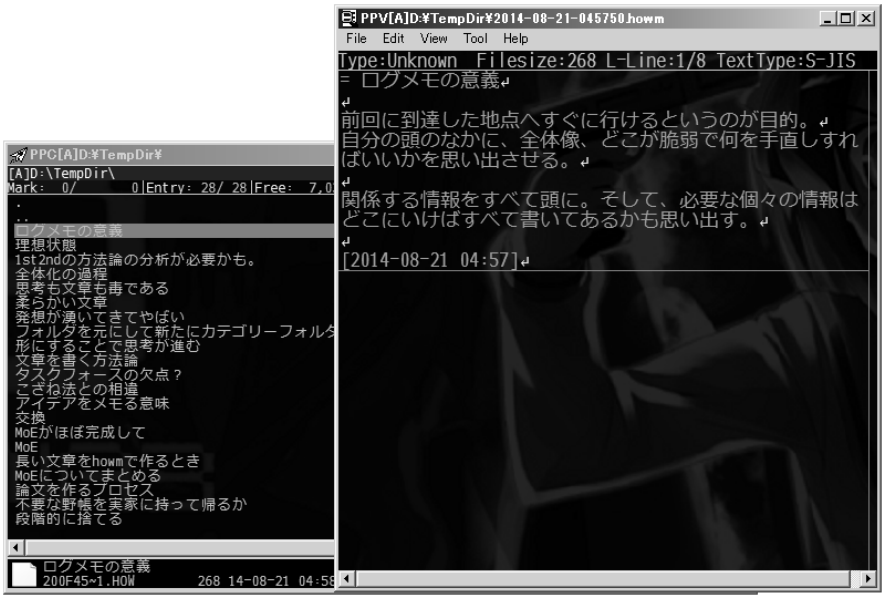
そののち、";"キーを押して表示形式メニューを表示し、howmtitle を選択すると、タイトル表示になる。



３．メモの閲覧

メモファイルにカーソルをあわせ、Enter キーを押すと、ビューアが起動。カーソルキーで表示ファイルを切り替えることができる。もう一度 Enter キーを押すと、ビューアは閉じる。

ファイルをエディタで編集したい場合は、E キーを押す。

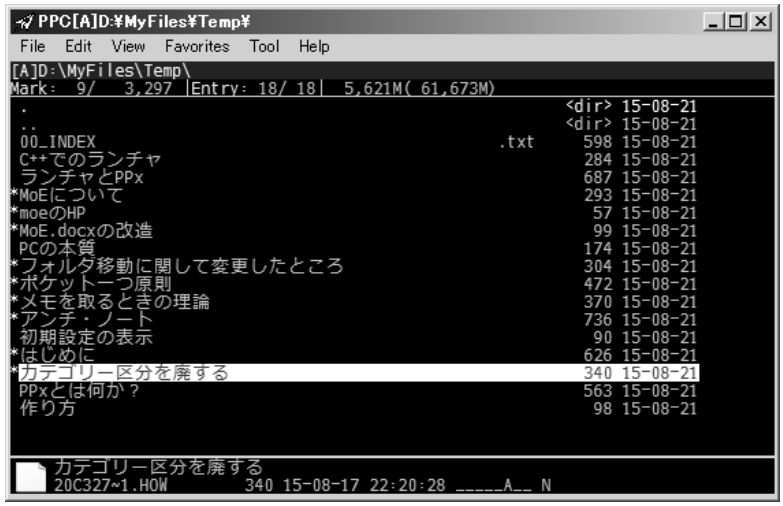


４．複数のメモを一つのフォルダにまとめる

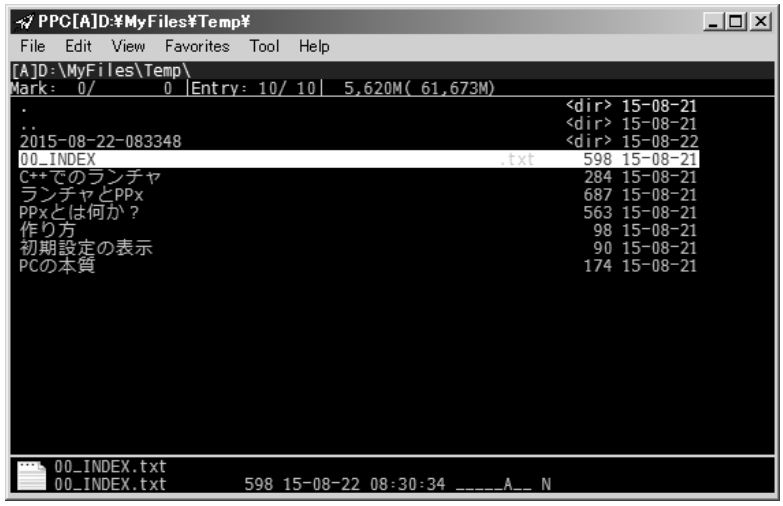
SPACE キーを押すと、カーソル下のファイルをマークすることができる。関係するメモをマークしたあと、Alt+M を押す。すると、カレントフォルダに新しくフォルダを作成し、その中にマークファイルが移動する。

この時、フォルダ名はタイムスタンプから自動で付けられる。表示名を変更したい場合は、フォルダにカーソルをあわせ、Ctrl+R を押して変更する。

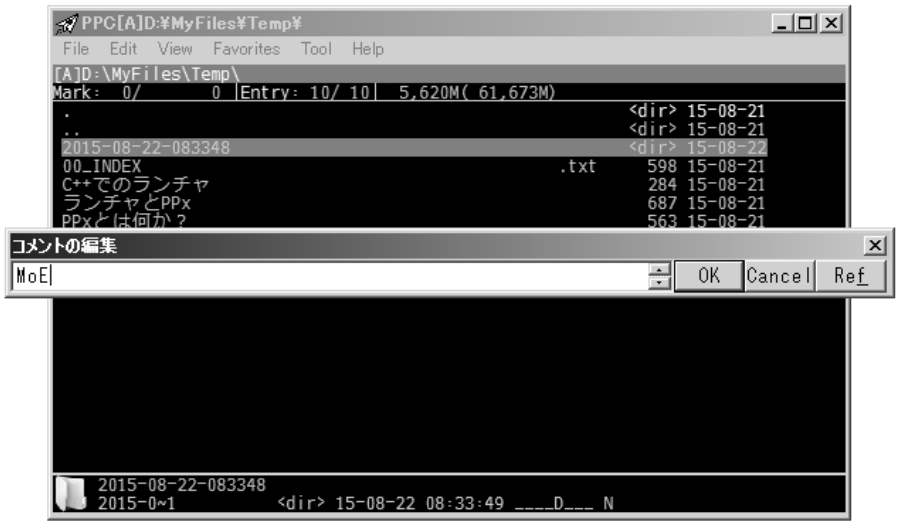
SPACE でマークし、Alt+M



フォルダを新規作成し、その中にマークファイルが移動



Ctrl+R でフォルダの表示名を変更



５．PPx のキーバインド

PPx とは正確にはファイル操作用のツール集であり、ファイラ部分は PPc、ビューア部分は PPv という名前である。以下は PPc のキーバインドの一部。

[TAB] 窓間移動。PPc が一枚のみの場合はもう一枚開いてから移動
[BS] 親のディレクトリへ移動
[F5] 表示ディレクトリの再読み込み
[Q] 終了
[E] テキストエディタを用いて編集
[K] ディレクトリの作成
[C] エントリの複写
[M] エントリの移動
[N] PPv による表示
[SPACE] カーソル位置のファイルをマーク ＆[↓]
[U] 書庫の展開

６．xyzzy のキーバインド

[C-c , c] 新しいメモを開く（Ctrl と C を同時押ししたあと、「,」を押して、C を押す）
[C-x s] メモを保存（Ctrl と X を同時押ししたあと、S を押す）
[C-x C-c] 終了（Ctrl と X を同時押ししたあと、Ctrl と C を同時に押す）
[C-Insert] コピー（Ctrl＋Insert）
[C-Delete] 切り取り（Ctrl＋Delete）
[S-Insert] 貼り付け（Shift+Insert）

メモの運用

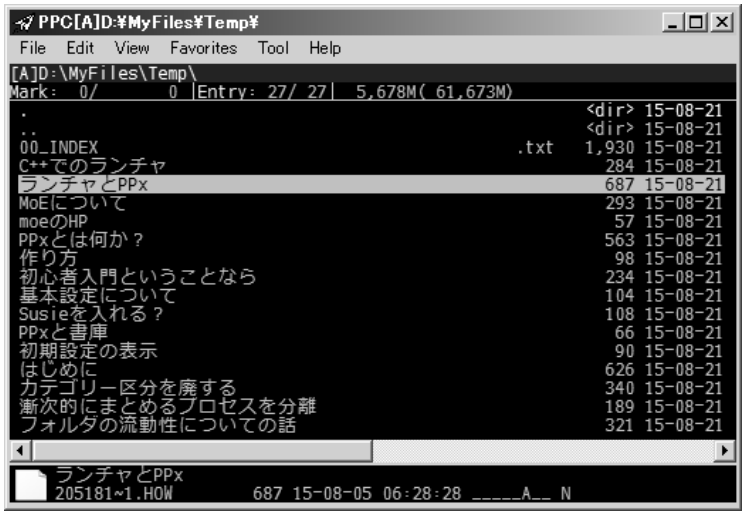
では、実際にメモを運用する段階へと移ろう。まずは、2 つのフォルダを適当な場所に作成する。

- 一時フォルダ（Temp）……一時的にメモをためるためのフォルダ
- ゴミ箱フォルダ（Dust）……不要なメモを捨てるためのフォルダ

PPx では、K キーを押せばフォルダを作成することができる。

1．メモをためる

何か思いついたことがあれば一時フォルダに行き、Shift+K からメモを作成する。
この時、カテゴリー分けをしようなどと考えてはいけない。どのようなメモであっても、まずは一時フォルダに、一切の区別なく保存するのだ。メモの種類も、カテゴリーも、以前に書いたメモとの関係性も気にせず、どんどん思いつくままにメモをためていこう。



書式の説明

ここで使っている書式は、howm と同じものである。howm とは Hitori Otegaru Wiki Modoki の略であり、本来的にはオフラインで wiki を実現するための方法である。自動でつけているファイル名も、ここから借用している。拡張子は howm であるが、内容はただのテキストファイルである。

とりあえず書くことが重要

データ化する際のポイントは、とりあえず書く、ということを優先し、文章の完成度を無視することである。内容が曖昧でもいいし、文章として成り立っていない、片言のものであってもいい。とりあえず、書くのだ。また、その際には既存のメモの内容を全く考慮しないで書く。もしかしたら以前、同じ内容のものを書いているかもしれないし、かつて書いたものと、今書いているものの内容は、矛盾しているかもしれない。それでもいいから、とりあえず書くのである。整合性に関する考慮は、メモをどんどん作ることの妨げになるのだ。

そうすると、表現が異なるだけで内容は同じであるメモが、いくつもできることになるだろう。それはそれで構わないのである。同じ内容のことを違う言葉で書いている内に、メモの内容は、だんだんと要領を得たものになっていっているはずだ。そして、文書を作るときには、それら「同じ内容を違う言葉で語っているメモ」を見比べながらやれば、短期間で適切なものができるだろう。

理論的考察ータイトルの実装（ファイル名か一行目か）

タイトルをテキストファイルでどう扱うかだが、二つの方法が考えられる。

- ファイル名をタイトルにする
- テキストファイルの一行目をタイトルにする

ファイル名をタイトルにする、というのはそのままだ。たとえば、「MoEーテキストメモの考察」というタイトルにしたいのなら、ファイル名を"MoEーテキストメモの考察.txt"とする。

テキストファイルの一行目をタイトルにする場合は、ファイル名はタイムスタンプから自動的に付けるようにして、テキストファイルの一行目を、タイトルにする。howm がこの形式だ。

ファイル名をタイトルにするという試みは、メモをためるという用途では不適切である。できるだけ余計なことを意識せずに書けることが、メモという性質上必要になるが、それとそぐわないからだ。理由を順に見ていこう。

同一タイトルにできない

同じフォルダ内に、同一タイトルのメモを複数作ることができないことがネックになる。ファイル名が重複してしまうからだ。そこで、一々ファイルを作るときに、フォルダ内に同じタイトルのファイルがないかを一々意識しなければならなくなる。面倒だ。

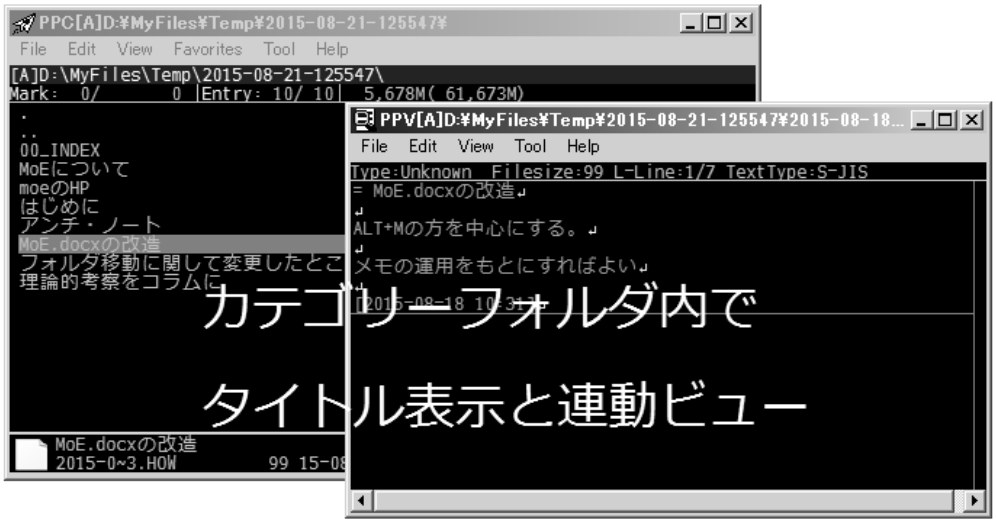
タイトルが最初から決まっているわけではない

アイデアメモの場合は、書き出したときにタイトルが決まっているわけではない。書いている途中で、何を書きたいのかがはっきりする、という場合があり得るわけだ。タイトルを決めずに内容を書き、そのあとでタイトルを付ける。あるいは最初に付けたタイトルを途中で変える、ということがよく起こる。タイトルは、好きなときに付けられて、好きなときに変更できる方がいい。だが、ファイル名をタイトルにしてしまうと、その融通がきかない。だから、一行目をタイトルにするほうが適切なのである。

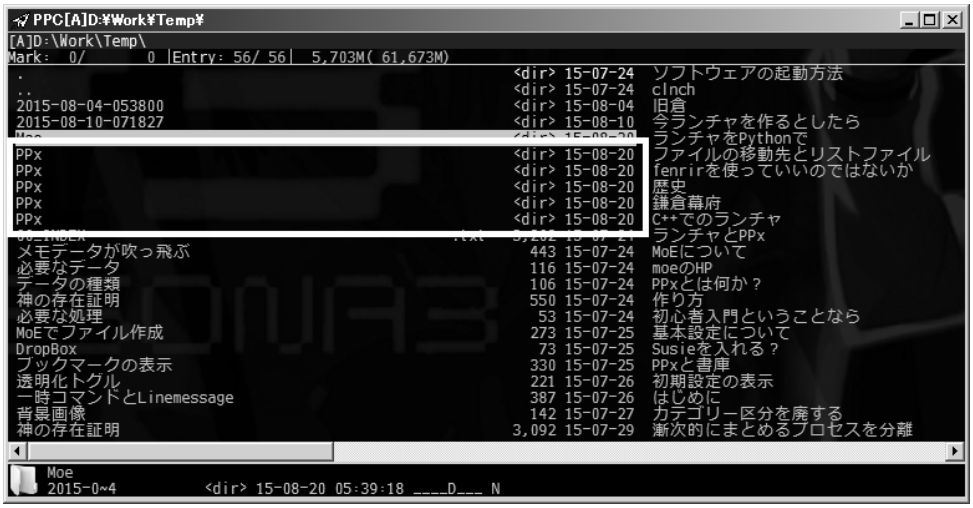
２．メモをカテゴリー分けする

ある程度時間が経過すれば、各々のメモが、どのような問題意識で書かれたものなのかが明確になってくる。そうしたら、そのとき初めてカテゴリー分けを行う。

一時フォルダに行き、タイトル表示とビューアで内容を確認しながら、同じカテゴリーのメモを Space キーでマークしていく。マークが終わったら、Alt+M を押す。すると、カレントフォルダにフォルダが自動で作成され、その中にマークしたフォルダが移動する。こうやって、個々のカテゴリーフォルダを作っていく。あとは、そのフォルダ内で作業をすればいいわけだ。



この時、フォルダ名はタイムスタンプから自動で付けられる。名前をつけて区別したい場合は、そのフォルダにカーソルをあわせ、Ctrl+R を押して変更しよう。この際、コメントという仕組みを利用するため、同フォルダに同じ名前のフォルダがあるかどうかを気にする必要はない。以下の画像のように、いくつでも同じ名前をつけることができる。



理論的考察ーフォルダ名

日時から自動でフォルダ名をつけ、コメントで表示名を変更することには、いくつか利点がある。その一つが、あるメモ郡をまとめる時に、一々そのまとまりに名前を付ける手間を省けることである。例えば「さっき考えたこと」「この論文のこの箇所について考えたこと」といったように、明確にカテゴリー名を決められないものに対してでも、カテゴリー分けが可能になるのである。このことにより、メモをカテゴライズするための精神的なハードルを下げるのだ。

また、メモのまとまりに名前を付ける際に、そのフォルダ名が他のフォルダ名とかぶっていないかを、一々気にかける必要が無くなる、というのも利点である。

３．不要なメモを捨てる

一時フォルダ内やカテゴリーフォルダ内のメモの内容を、ビューアで閲覧している時に、不要になったメモが見つかることがあるだろう。例えば、すでに作成した文書にその内容が反映されているメモや、状況の変化によって、すでに問題意識自体が古くなっているメモだ。そのようなメモを見つけたら、その都度、ゴミ箱フォルダに移動する。

私の場合は、カテゴリーフォルダのメモを利用して、一つの文書を完成させたあとに、最後の確認がてら、そのフォルダ内のメモをざっと閲覧して、不要だと判断したものを一気に捨てる事が多い。

理論的考察ーフォルダ移動の理由

ここでは、メモを捨てるとは言っても、実際に捨てているわけではない。別のフォルダに移動して、見えなくしているだけである。

ファイル削除ではなく、ファイル移動で済ませている理由はいくつかある。その一つが、捨てるハードルを低くするためである。ファイル移動であれば、ゴミ箱フォルダに捨てたとしても、あとで頑張れば、Grep なり何なりを使って取り戻すことができる。つまり、不要なメモを捨てる際に、一々これは本当に捨てていいのか、後で後悔しないか、という究極の決断をする必要がなくなるのである。こうして、削除のハードルを下げることで、一時フォルダ内とカテゴリーフォルダ内の秩序を維持しやすくしているのである。

それに、ここでためているのはただのテキストファイルなので、不要なファイルを一々削除しなくても、ディスクの容量面で問題になったりすることがない。これが、削除をしないもう一つの理由である。例えば私の場合、これまで５年間かけて作成したメモを全て合わせても、35MB 程度しかない。数百 GB から数 TB の HDD が主流になっている今、この容量は微々たるものである。

変化の無いメモを別フォルダへ

変化の無いメモとは

時間が経過するに従い、一時フォルダ内のカテゴリーフォルダの中に、長期間変化も無いまま居座っているものが出てくるだろう。数ヶ月間、その中にあるファイルが増減したことも、編集されたこともない。それどころかほとんど参照すらしていないようなフォルダである。

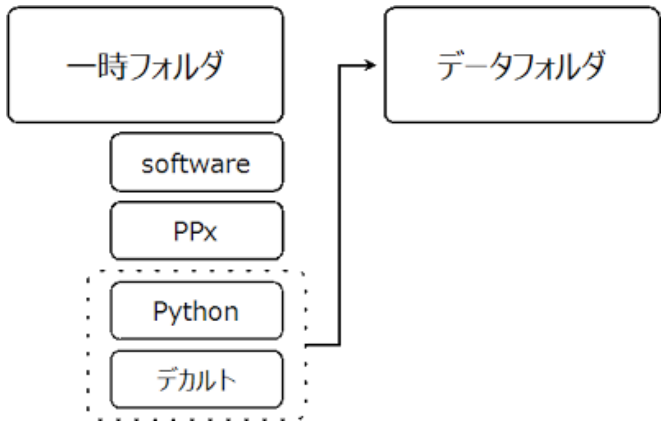
かつて、眼前にある問題を解消しようとして、そこにまとめられているメモを作ったかもしれない。だが、状況の変化によって、実際にはその問題を解決してしまったわけではなくても、当面は問題にしなくてもよくなることがある。そのような場合、このような形でフォルダが取り残されてしまうわけだ。

そのようなフォルダを、一時フォルダ内にずっと置いておくのは鬱陶しい。現在の問題意識から外れたフォルダが存在することは、思考のノイズになるからである。一時フォルダには、今取り組んでいる課題に関係するファイルとフォルダだけが存在していてほしいのだ。しかし、だからといってそれをゴミ箱フォルダに捨ててしまうわけにもいかない。特にその問題意識が解決したというわけではないので、後々必要になるかもしれないからである。

やり方

これを解決するため、新たにフォルダ（ここではデータフォルダと呼ぶ）を作成する。

そして、一時フォルダ内でほとんど変化の無くなったメモファイルは、フォルダごとデータフォルダに移動する。こうすることで、一時フォルダ内に使っていないフォルダがあって気が散る、ということも無くなるわけだ。



データフォルダについて

一時フォルダからデータフォルダに移動する際は、内容が似通ったフォルダを一つのフォルダにまとめてしまう。細かい区分は後々無意味になるからだ。一時フォルダでは、同じタイトルのカテゴリーフォルダがいくつも並んでいて問題なかった。今、取り組んでいる課題であれば、細かな区別をしていたとしてもそれを把握できるからである。しかし、データフォルダの場合は、その前提が違ってくるのである。また、それゆえ、一時フォルダでそうであったように「フォルダ名はタイムスタンプで付けてコメントで区別する」という方式を取る必然性もない。フォルダ名自体を変更してもいいだろう。

また、データフォルダ内にあるカテゴリーフォルダの扱い自体は、基本的に、一時フォルダ内のものと同様でいい。タイトル表示と連動ビューで活用し、不要なメモを見つけたら適宜ゴミ箱フォルダに移動しよう。

アナログツール

常に PC の前に座っている人でも無い限り、手帳なり何なりにメモを書き付けて、あとでそれを元にデータ化をする、ということになるだろう。そこで、アナログのメモの方法について考察する。

必要な要素は、

- 方眼あるいは白紙
- バラバラにならない
- 安価
- 持ち運びがしやすい

あたりである。あとでデータ入力をすることを考えると、アナログツールは一つに集約しておいた方がいい（ポケット一つ原則と言う）。そこらへんにある紙切れに適当にメモをしたとしても、あとでそれが紛失する危険性があるからだ。メモは一冊に集約し、何でもそれに書き込んでおいて、後でそれをデータ化することになる。また、適当なアイデアでも気兼ねなく書けるように、安価であるほうがいい。

私は現在、コクヨの測量野帳を利用している。一冊 40 枚（80 ページ）あり、200 円以下で購入可能である。新書の丈を少し短くしたくらいの大きさなので、ポケットに入れて持ち歩くことも容易だ。また、カバーが硬いので、外で立ったままメモをすることも可能だ。他の選択肢としては、無印良品のらくがき帳がある。携帯性は野帳に劣るが、コンビニで買えるためにストックが必要ないという利点がある。また、書いた箇所を破り捨てるのが容易いため、野帳のようにデータ化して用済みになったものがたまる、という問題もない。後はモレスキンもいいが、値段が張るのがネックである。

書き方

外にでるときには常に野帳を持ち歩き、ボールペンと共にすぐに取り出せるようにしておく。そして、思いついたことがあったら、すぐに野帳に記入する。人との会話の記録でも思いつきでも本を読んだ感想でも、とにかくすべて、これ一つで済ませる。文字は汚くても問題ない。どうせすぐにデータ化するからだ。書いてからデータ化するまでの期間、自分が読めればそれで十分なのである。

アイデアは、思いついた時間順にびっしりと、詰めて書いていく。京大式カードのように、一枚一項目原則にはこだわらない、ということである。もちろん、こうすると同じカテゴリーのメモが飛び飛びに現れることになる。だが、それは後でデータ化する際に解消されるのだから、気にする必要は無いのである。

書く時の書式は自由でいいが、日付だけは入れるべきである。メモ全般に言えることだが、日付の無いメモは使い物にならないことが多い。また、データ化の際の参考になるよう、簡単な見出しをつけてもいい。

理論的考察ーメモと編集の分離

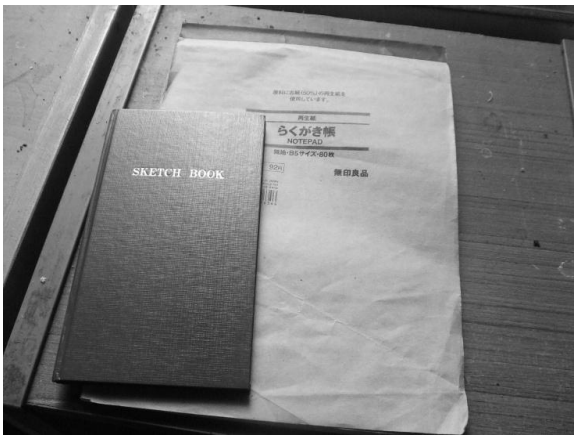
手帳でデータベースをつくろうという試みは、基本的に失敗すると考えていいと思う。というのは、メモをする際に求められることと、後でそれを利用する際に必要になることとが原理的に矛盾するものであり、個々の工夫で克服できるようなことではないからだ。

思いついたアイデアをメモすることと、その後それを活用することとは別物である。前者で必要なのは、何も考えずに書けることだ。他のメモとの関係性だとか、以前に何を書いたかなどは一切無視するべきであり、時系列で書くことが望ましい。後者で必要なのは、カテゴライズであり、時系列とはまた

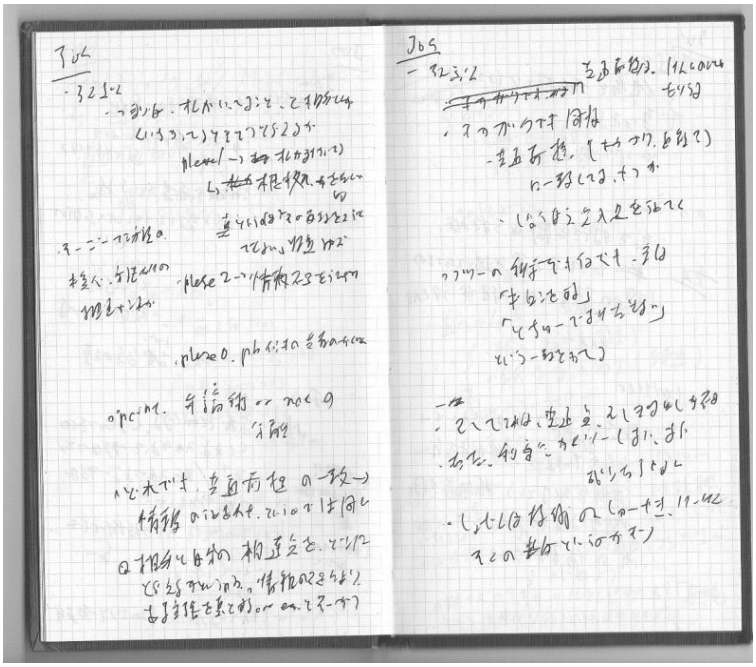
別の秩序を必要とする。この両者は本質的に矛盾するのだ。だから、この二つは物理的に分けた方がいい。そこで、まずアナログツールによって時系列順にメモをとってから、あとでそれをデータ化する、という段階的な処理をしているのである。

手帳が一冊程度であれば、目次をつける等のちょっとした工夫によって、この問題は何とかなるかもしれない。しかし、時間の経過とともに扱うべきメモは増大していく。だから、この手の試みは、必ずどこかで破綻することになるのだ。

コクヨの野帳と無印のらくがき帳



野帳の使用例



論文の書き方

特に喫緊の問題が無い場合、アイデアは思いついた時に適当にメモするのでいい。だが、それ以外に、特定の期日までにまとめる必要のあるものがある。論文の締め切りが近い、現在取り組んでいる問題を整理して解決したい、読んだ本の内容を覚えている内にまとめたい、等々の時だ。このような場合は、意識的にアイデアを出して、それを短期間の内にまとめることが必要になるだろう。以下では、その際の方法論について考察する。

全体図

まず、喫茶店などに行って、短期間に集中して、アイデアを野帳か計算用紙に書き出す。この時は、とにかく頭にあることをすべて書き出すことを心がける。

このメモをデータ化し、それを元にしてとにかく一つ、文書をつくってしまう。アイデアメモを作ってから、文書にするまでの期間は、できるだけ短い方がいい。せいぜい一週間程度だろうか。そうしたら、それを徐々に修正する、という形で、完成に近づけていく。

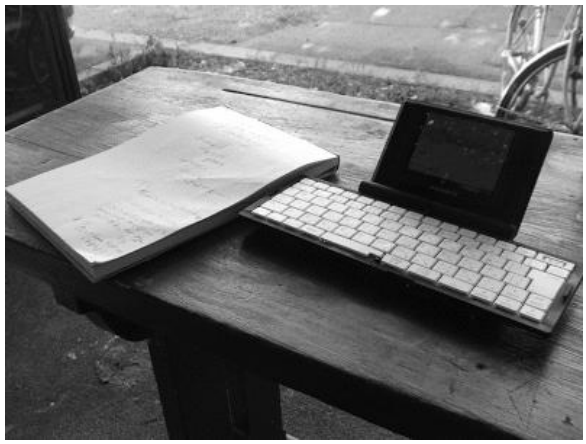
完成した文書を一度に作るのでは無く、荒削りでもとりあえず形にし、それを徐々に修正して完成に近づける、というイメージである。一度文書を作ったならば、自分の理論のどの点が不十分であり、何について知らなくて、何を新たに考察する必要があるのかが見えてくる。そして、次はそれを念頭に置いた上で考察し、またそれをメモの形で書き出し、それを元にして、文書を修正する。それでまた不十分な点が見つかれば、またその箇所を修正する……というように、漸次的に文書を完成させるのである。そうやって全体の形を整えていき、最後に文体や誤字や表現といった、細かいところに手を入れる。

そのような方針が無いところで、いくら考えたとしても、それは結果に結びつかないことが多い。全体的な指針が無いと、それが文書の完成に寄与するかどうかと無関係に、その時たまたま目についたことを、闇雲に考察することになってしまうからである。労力としても時間としても、無駄なのだ。準備を万全にして、個々の論証の流れや全体の論理展開を頭に思い描き、いきなり完成した文書を作り出す必要性など無い。最初はどれだけ適当なものであってもいいのである。

ポメラの活用

この時、ポメラがあると便利である。ポメラを持ち歩いておけば、野帳にアイデアを書きだしたあとすぐ、記憶が鮮明な内に、データ化することができる。あるいは、野帳を経由せずに、ポメラにアイデアを直接打ち込めば、さらに手間が省けるだろう。

右の画像は、私が使っているポメラ DM10 である。ただ、ファイル名をタイムスタンプから自動で付ける機能が、この機種には備わっていないため、新しく購入する場合は、後継機を選ぶほうが適切だと思う。



ポメラ DM10

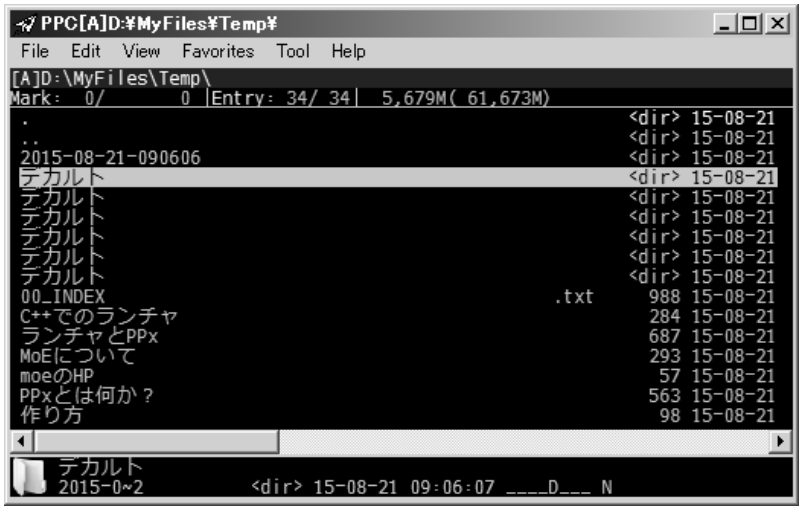
メモのデータ化

メモのデータは、通常のメモと同様、一時フォルダに保存する。野帳にメモをしていた場合は、それを見ながら通常通りメモファイルを作成する。ポメラで既にデータ化していた場合は、それを howm 形式にする。Shift+K で xyzzy を起動してから、そこにコピペをして保存、というやり方でいいだろう。もし、扱うファイル数が多くなり、その作業が面倒になったら、その時にスクリプトの利用等を考えればいい。

メモをすべてデータ化したら、カテゴリーフォルダにまとめる。関係するメモにマークをしてから Alt+M だ。この時は、自分の主観に合わせて、細かい区分をしていいだろう。現在取り組んでいる課題であれば、細かい主観的な区分であっても把握できるからだ。そうして、個々のカテゴリーフォルダ内でメモを閲覧し、文書を修正するなりなんなりの作業を行う。

フォルダ名を変更したいときは、Ctrl+R だ。私の場合、細かなフォルダ区分をしても、それらに付ける名前は全て同じ、大雑把な括りのものにして、後はそのフォルダに入って区別する、というようにしている。もちろん、そのカテゴリーフォルダ内での作業がすぐに終わり、そのフォルダがすぐ不要になるというのであれば、一々名前を付ける必要はない。

同じ名前を複数のフォルダに付ける



メモをこのように細かく区分することは、メモの削除を容易にもしている。一つの文書を作り終えたら、利用した各々のフォルダに行き、そこの中にあるメモをざっと見て、反映されていないものが無いかを確認する。同じフォルダには、同じような内容のメモばかりがあるはずだから、このことは容易なはずだ。そうして、中にあるファイルがもはや不要であると判断したなら、フォルダごとゴミ箱フォルダに移動すればいいのである。

文書の修正過程

文書の修正をする際は、PC 上だけでやるのではなく、紙媒体も併用するのがおすすめである。そのほうが、全体の構造を把握しやすいのだ。

プリントアウトしたら、それを読みながら、ペンで書き込みを加えていく。それが終わったら、その書き込みを元のデータに反映する。そうしたらまた、そのデータをプリントアウト。また書き込み、反映……という過程を繰り返すのだ。

印刷例。xyzzy の印刷設定で二段組にしている

H:/hown2/2014-03-25-122802.hown Mon, 31 Mar 2014 17:18:33

= メモを修正する過程

前回の続き。これも独立記事として。
この次が、終了過程？ログだとかファイルの整理だとか

**メモの修正過程

三つの過程からなる。

プリントアウト
修正
データへの反映

**プリントアウトする

前回のファイルを開き、プリントアウトする。

読みやすさ
ある程度文章が詰まっているほうが紙の無駄にならなくていい

というののバランスを考えて、以下の様な形式にしている。二段組。あとファイル名とか日付とかも入るように。

**修正過程

これを持ち歩き、喫茶店とか集中できるところに行って修正をする。
文章を読み、何か気になることとか不十分だなと感じる箇所があれば、その点について書き込みをする。

入れ替え
文章の付け加え
文章の削除
読みにくい文章を直す
不要な箇所の削除
必要な箇所の追加
誤字の修正

などなど。
基本的に突き詰めて考えなくても、適当にやればいいのだが、個々の作業過程を意識的に分離すると捗ることもある。「今回はどこを削れるか中心に読んでみよう」みたいに。文章を削る作業と、わかりやすくするために追加する作業とは別のものなのかな、という印象はある。

基本的には余白に書き込むのだが、長くなりそうで書ききれないという場合は、野帳だとからくがき帳だとかに、わかるようにそれを書き込む。場所がわかるように、見出しをタイトルにするのがいい。

**反映

次は、この書き込みを元のデータに反映する。
先のフォルダに行き、元データを開く。もしも、元のデータも残しておきたいという場合であれば、そのコピーをhown形式で作る。末尾に連番を降るといいかもしれない。あと、修正箇所が多すぎる場合には、元データの修正ではなく、一から書き直す場合もある。

この過程で、元の文章から削ぎ落とす節があるけれども、またもしかしたらつかうかもしれないという場合は、同じフォルダに新しくそのファイルを作り、そこにコピペして保存する。

このようにして修正を終えたら、またそれをプリントアウト。修正して反映して…というのを繰り返す。

フォルダ内には、連番が振られた各バージョンと、反映されずに残っている個々の箇所のファイルとが存在することになる。

以上が、hown形式でやること。

**その他

***DocとかTexとか

ある程度まで行くと、見出しだとか階層構造だとかで、この方法では対処しきれなくなる。
また、どこかに投稿する場合であれば、それを保存するための形式というのにあわせなければならないだろうし、そのためにも先の形式から、この形式に移行する。

先の過程で修正をしたhownファイルをコピペ。見出しとかについて整える。

次は、これに対して先にしていた修正を行っていく。やることは同じで、プリントアウト→書込み→元のデータに反映。

***全体と個別

ある程度まで文章が大きくなると、自分でそれを全て読んでから修正、という過程が面倒になる。個々の修正が必要だけど全体はそんなでもないというとき。そのようなときは、作業を分割するのがいい。

全体的な文章を一部プリントアウト。これについては、基本あまり書き込みをしないようにする。可読性の確保のため。基本、これは比較的長く残す。
これを読み、修正が必要だと思われる箇所をチェック。ページ数だとかをどっかにメモっておく。どこどこは修正が必要だというように。作業過程を分割。あとこれこれの数だけやればとりあえずよくなりそうだというの。
そして、その修正が必要だと判断したページだけをプリント。それを修正して、それを元データに反映。元のプリントは、たまに参照するだけで基本読まない。読むのはこの個々の箇所のみ。

そうして、最初に気になった箇所を全て修正し終えたら、全体的な文章をまたプリントアウト、また同じ過程を繰り返す。

***CatMemoNoteの利用

ある程度文章が煮詰まってくると、PCのデータ上だけでも修正できるようになる。
その時にはCatMemoNoteを併用する。

データを修正しながら、思いついたことはここにメモ。全体的な構成に関する思いつきは、とりあえずここに書き込む。このようなものは賞味期限が短く、hown形式で保存しても後々ノイズになる。
それ以外の思いつきについても、とりあえずここに書き込んでもいい。その場合は、ある程度時間がたてばそれをhown形式で個別に保存する。

その他、プリントに書き込まれたものを反映させる過程で使う場合がある。その書き込みを元に一旦ここで文章化。CatMemoNote上で編集をし直し、体裁を

- 1 -

参考文献

梅棹忠夫『知的生産の技術』1969
野口悠紀雄『「超」整理法』1993
野口悠紀雄『続「超」整理法・時間編』1995
野口悠紀雄『「超」整理法〈3〉』1999

参考サイト

PoIC（ http://pileofindexcards.org ）

関連サイト

Memo on the Electron（ https://sites.google.com/site/moesystem/ ）
つかさのほえほえ日記（ http://hoehoetukasa.blogspot.jp/ ）

Memo on the Electron
発行日　：2015 年 8 月 23 日
発行者　：柊つかさ
Twitter　：@tukasa1900