UNIXにおけるフランス語テクスト処理について

重　見　晋　也

はじめに

本稿は，UNIX 上でフランス語テクストの処理し，文学研究に役立てるための方法を論ずるものである。

「UNIX 上で」処理すると限定したが，この点については第 I 章で考察する。ここでは，パーソナル・コンピューター（PC）のオペレーティング・システム（OS）と比較しても UNIX が後述する幾つかの点で，文学研究に応用しやすいとだけ述べておく。

コンピューターで「フランス語テクストを処理する」という問題設定が可能となる背景としては次の三つを挙げることができる。

まず第一に，広く人文学研究の分野にコンピューターが利用され始めたことが挙げられる。

PC の発達と普及とともに，人文学研究において，PC に限らずコンピューターを活用した研究が成果を上げつつある。コンピューターの活用はワープロだけという時代は遠ざかり，より積極的な活用が本格化している。

例えば，心理学，地理学，考古学などの分野では特に，コンピューターは研究活動になくてはならない存在になっている。考古学を例に取れば，日本でも遺跡物理探査法の開発研究プロジェクトが始動し，そうした成果として，エジプトの遺跡発掘調査や，アンコールワットの遺跡群の発見において大きな役割を演じたことが報告されている。こうした例に倣って，文学研究にもコンピューターを役立てるための模索が始まりつつある点を指摘することができる。

第二に，コーパスの充実が挙げられる。

コンピューターが利用しやすいものとなっても，文学研究の対象となるテクストがデジタル化されていなければ，文学研究にコンピュータを役立てることは不可能である。現在英文学では，Oxford Text Archive（OTA）では網羅的に文学作品のデジタル化を収集し，インターネット上で公開している[1]。またフランス文学作品についていえば，幾つかの文学テクストが部分的にはデジタル化されていたが[2]，バルザックの *Comédies Humaines* の全作品や，モンテーニュの全作品と資料がデジタル化され CD-ROM で入手可能になった。哲学作品でも，フッサールの全集が CD-ROM 化されている。こうした例に見ることができるように，研究対象となるテクストのデジタル化が行われることで，作品研究に活用するための素地が整ったといえる。

最後に，コンピューターの扱う文字コードの問題を挙げることができる。

周知の通り，コンピューターはアメリカで発達したものである。それ故，コンピューターはその誕生からしばらくは，英語しか表示・入力することができなかった。現在では，ASCII 規格が拡張され[3]，日本語や英語以外の言語をコンピューターで処理することも可能になった。しかし実際には，UNIX がデフォルト状態で扱うことができるのは ASCⅡコード７ビットの文字だけであり，英語以外の言語を用いるためには，特別なソフトウェアや設定を必要とする。PC の OS ではそうした問題は克服されているように見えるが，UNIX においては，まだ知られていないことが多い。それ故，今後コンピューター，特に UNIX をテクスト処理の道具として活用する際に，英語以外の言語をどのように扱うかを整理しておくことは重要であろう。

以上述べたようにして設定することができる，UNIX 上でフランス語テクストを処理する，という問題意識に基づいて，その方法と可能性について検証するのが本稿の目的である。
第Ⅰ章では先ず，論の冒頭で言及した「何故 UNIXか？」という問いかけに答える形で，UNIX をプラットフォームとして研究に活用する理由を述べる。第Ⅱ章では，デジタル・テクストの入手から，それを活用するための準備について述べる。そして第Ⅲ章では，UNIX 上でテクストを処理する際のツールの紹介と，その利用について論じ，研究に必要なツールの理想像について考察する。

Ⅰ．何故 UNIX か？

PC が普及し，Windows や MacOS 上で問題なくフランス語などを処理できる現状があるにもかかわらず，何故「UNIX 上で」フランス語を処理しなければならないのであろうか？現在研究者が個人で利用するコンピューターとしては，PC が主流である。本章ではまず，PC の OS と UNIX との比較を行い，それぞれの長所と短所を比較し，UNIX を活用することの意義を述べる。

Ⅰ.1 UNIX の場合

1.1.1 UNIX の長所

UNIX システムの解説書を開くと，しばしばその最も重要な特徴として挙げられるのは，次の六つである。１）マルチタスク能力，２）マルチユーザー能力，３）移植性のよさ，４）強力で豊富なプログラム，５）通信や電子メール，６）アプリケーション・ソフトのライブラリである[4]。

文学研究への活用という点からすれば，これら六つのうちで重要なのは，「マルチタスク能力」，「マルチユーザー能力」，「プログラムの提供」それに「通信」である[5]。

「マルチタスク能力」とは，同時に一つ以上のタスク（処理）を実行する能力をさす。現実生活において，電話をしながらメモを取ったりするように，ワープロソフトを使いながら別の文書ファイルを印刷したり，別のプログラムにバックグラウンドで処理をさせたりする能力をさして

いる。この能力により，限られた時間でより多くの処理を並行して行うことができるようになる。

「マルチタスク能力」は，現在では PC の OS においても，部分的には実現されているが，UNIX 程は完成されていない。例えば，ワープロソフトである単語を別の単語に置換させながら，電子メールソフトを使ってメールを書くといった作業は，残念ながら Windows や MacOS にはできない。

「マルチユーザー能力」とは，一度に複数のユーザーが一つの UNIX システムを使って処理を実行できることをさす。前述した「マルチタスク能力」とあわせていえば，UNIX システムは，複数のユーザーが同時に一つのコンピューターを使って，それぞれが複数の処理を同時にこなすことができることを示している。すなわち，先ほどの電話を例に取れば，PC では一台の電話機は一人しか使うことができず，しかも一人の相手にしか通話することはできない。それに対して，UNIX システムでは，一台の電話を五人が同時に使って，しかもその五人がそれぞれ複数の相手と同時に通話することが可能なのである。

「プログラムの提供」とは，UNIX には OS 自体に様々なプログラムが予めインストールされている，ということを示す。前者はツールとも呼ばれるが，OS を操作する際に欠かすことが出来ないものと，OS の操作には必ずしも必要ではないものとに区別することができる。これらツールの中には，文学研究に応用することができるものが幾つか含まれており，そうしたツールについては第Ⅲ章において紹介する。

また，UNIX は元来プログラマーにプログラムの開発環境を提供することを目的として開発されたオペレーション・システムである。それ故，ユーザー自らがプログラムのソース・コードを作成し，必要であればそれをコンパイルすることもできる。こうした環境は，PC では提供されていない。この点こそが，UNIX が学術研究に広く利用されていることの由縁であろう。

「マルチタスク」や「マルチユーザー」が可能となるには，「通信」が重要な役割を演じている。コンピューターは通常キーボードやマウスを用いて，モニタを見ながら操作するが，一台の UNIX を複数で同時に使う場合には，当然その人数分のキーボードやマウスが必要になる。ところで，UNIX は，OS レベルでネットワーク機能を備えている。それ故，ユーザーはネットワークを介して，UNIX システムにログインし，処理を行うことになる。そうすることによって初めて，複数のユーザーが同時に複数の処理を行うことができるのである。

この点は非常に重要である。PC の場合，ユーザーは自分のコンピューターの管理から操作に至るまで，全てに責任を持つことが必要となる。コンピューターが思うように動かなくなったとしても，基本的にはユーザーが責任を持ってその問題を解決することが望まれる。しかし，UNIX を利用する場合は，コンピューター自体の管理は，管理者に任せておき，ユーザーはプログラムを使用するだけである。新しいプログラムが必要となった場合でも，自らインストールするのではなく，管理者に依頼すれば良いわけであり，その点においては，ユーザーの負担は格段に軽くなるのである。

こうした特徴を安定した環境でユーザーに提供しているのが，UNIX システムである。

1.1.2 UNIX の短所

前節で見たように，UNIX は，非常に強力な OS であり，また安定しているという点でも信頼性が高い。

しかし，UNIX を文学研究に応用する場合に，短所を指摘することができるのも事実である。

UNIX の短所としては，次の点を挙げておく。1）コンピューターが使えなくなることがある，2）市販の UNIX システムは高価である，3）操作を憶えるのに時間がかかる。

「コンピューターが使えなくなる」とはどういうことであろう。

UNIX も，コンピューターであり，PC と同じく定期的なメンテナンスが必要となる。その場合，UNIX は全てのユーザーからのアクセスを拒否する「シングル・ユーザー・モード」になる。そうした場合には，ネットワークにつながったコンピュータから UNIX にログインすることができなくなる。

また，基本的にネットワーク経由の利用が前提となっているために，ネットワークに接続されていないコンピュータでは UNIX を使うことができない。電話回線を利用し，PPP 接続や無手順接続した後に利用することもできるが，その場合は PC が必要になる。

次に UNIX は，一番安価なものでも，PC と較べた場合，価格が三倍から五倍する。この点は UNIX を使う場合に大きな足枷となる。

しかし，ほとんどの大学で，研究用に UNIX システムが利用可能であり，個人で UNIX を所有する必要はない。また，個人的に UNIX を使用する場合でも，現在では，Linux や Free BSD に代表されるように，従来の PC のハードウェアに無料の UNIX システムをインストールすることができるようになった。こうした，オープン・ソースの流れは，従来市販されていた SunOS（あるいは Solaris）などにも影響を与えており，今後は UNIX システム自体は，非常に安価に構築できるようになると予想される。

UNIX を利用する際に最も問題となるのは，「操作を憶えるのに時間がかかる」という点であろう。

これは，UNIX の操作が CUI（Character User Interface）環境で行われることに起因しているであろう。確かに UNIX にも，Xwindow システムという GUI（Graphical User Interface）環境があるが，ツールを使う場合には，コンソール・ウィンドウと呼ばれるウィンドウで，キーボードからコマンドを入力することが不可欠になる。しかしながら，UNIX 操作に必要な基本的なツールは限られているし，MS-DOS を利用したことがある場合には，それ程戸惑うことはないだろう。

I.2 PC と UNIX との比較

PC を利用する際の長所としては，次の三つの点が挙げられるだろう。1）GUI で操作が容易，2）豊富で多機能な市販アプリケーションの充実，3）UNIX に較べると安価であること，である。

MacOS や Windows95 の登場以降，PC は，GUI を駆使して，UNIX に必要なコマンドラインを憶えることなく，マウスだけでほとんどすべての操作を行えるようになった。この点はUNIX では実現しておらず，PC を特徴づける点である。

また，PC の処理能力の向上と共に，市販のアプリケーションも，多機能になってきており，それらを用いて文学研究に活用する例も報告されている[6]。こうした PC の活用は，PC が現在多くの研究者によって利用されている現状を考えると，方法として有効であると考えられる。しかし，処理の対象となるテクストが大きい場合には，極端に処理が遅くなったり，あるいはかなり処理能力の高い PC でなければ実用的でない。

PC の短所として次の二点を特に挙げておきたい。

第一に，前述したように，PC の場合には，ユーザー自身がコンピューターの維持・管理にあたる必要があり，プログラムの使用だけに集中できないという点が問題となる。

一度でも PC を使ったことがあれば，コンピューターが「フリーズ」してしまいどうすることもできなくなったとか，OS の再インストールを余儀なくされた，といった経験があるのではないだろうか。しかし，UNIX であれば，前述したように，安定した OS のおかげで「フリーズ」することは稀であるし，また万一コンピューターにトラブルが生じた場合でも，管理者がその問題を解決してくれる。「UNIX は難しい」という神話があるが，この点からいえば，UNIX は PC よりも，ユーザーにとっては，「難し」くはないのである。

第二に，PC は既にあるアプリケーションを使う場合には非常に使いやすいのであるが，自分でプログラムを書く場合には，それほど使いやすくはない。確かに PC を用いてソフトウェアの開発を行うことは可能である。しかし，PC を用いて PC 用のプログラムを開発する場合には，先ず専用の開発ソフトが必要である。さらに，C や C^{++} などのコンパイルを必要とする言語で開発を行った場合には，各 OS にあわせてソース・コードをコンパイルする必要がある。

それに対して UNIX は，前述したように，元来プログラマーのために開発された OS であり，コンパイラなどのツールも充実している。また，プログラムは一般的には，開発する言語にあわせてソース・コードといわれる文字列をエディタで欠くという作業になるため，ウィンドウ・システムの有無はプログラムの開発の容易さを決定づけるものではない。

I.3 UNIX の利点

以上のように見てくると，PC と UNIX とを比較しても一長一短であることが分かる。しかし，UNIX にはプログラミング言語を用いる環境が予め整っているため，研究者が独自にプロ

グラムを開発する場合には，非常に有効な選択肢であると考えられる。

また，PC にはつきものの「フリーズ」といった現象も，ユーザー側が対処する必要がない，という点でも UNIX を研究に活用することには，充分な利点があるといえるのではないだろうか。

Ⅱ．テクスト処理の注意点

UNIX でテクストを処理するといっても，処理するべきテクストがデジタル化されていることが必要なことは自明である。本章では，デジタル・テクストの入手と，テクストを処理する際の前準備について述べる。

Ⅱ.1 デジタル・テクストの入手

デジタル・テクストを入手するには，三つの方法がある。一つは，市販されているデジタル・テクストを購入する方法であり，もう一つは自分で必要なテクストをデジタル化する方法である。そして最後にインターネットなどを経由してデジタル・テクストを入手する方法である。

PC の高性能化と共に，PC に接続する周辺機器もそれと比例して高性能になってきている。それに伴い，以前なら大規模なプロジェクトが必要であったテクストのデジタル化も，研究者個人で行えるような環境が整ってきている。

研究者が個人でテクストをデジタル化する場合には，スキャナと OCR ソフトを用いて作業をする必要がある[7]。UNIX でも，同様の作業が必要であるが，UNIX 用のスキャナは依然高価でありまた一般的でもないため，UNIX を利用するよりも PC を使った方が良いといえる。また，OCR ソフトといえども100%の認識率を持つわけではないため，最終的には，オリジナルと比較しながら，デジタル化したテクストを手作業で校正することが必要になる。

実際には，PC でスキャンしデジタル・テクストを作成する時間よりも，テクストをより完全なものにするために校正する時間の方がより多く必要であるし，また校正も不完全になりがちである。

ところで，電子メールを用いた研究者間の討論の場となっているメーリング・リストの中には，研究者個人がデジタル化したテクストを研究者間で共有しようという試みもある。こうした場合には，デジタル・テクストの誤りも発見されやすい。それ故，個人レベルでテクストをデジタル化した場合には，Web でテクストを公開するなどした方が良いだろう。

次に，デジタル・テクストを入手する方法としては，市販のテクストを購入することも可能である。しかしながら，現在までにデジタル化されているテクストの多くは，ある作家の一作品にとどまっているものがほとんどである。また，INaLF（CNRS）のフランス文学テキスト・データベースである≪FRANTEXT≫をもとに編纂された≪Discotext≫は，著者100名，収録文献300点を数えるが，一人の作家の全集というわけではない。

現在全集で CD-ROM による販売が始まっているのが，モンテーニュとバルザックのデジタル・テクストである。バルザックの全集は初版本をもとに CD-ROM 化されている。モンテーニュの全集は，デジタルテクストだけではなく，手稿の画像データなどの資料も収められているが，非常に高価である。今後も多くの作家のテクストがデジタル化されることが予想されるが[8]，著作権や版権の問題，さらにどのエディションをもとにデジタル化するかなど，課題が多いことも事実である。例えば，フランス文学研究では Pléiade 版を用いることが多いが，バルザックの場合は意識的に Pléiade 版に準拠するのを避けたわけであり，今後こうした方法が採られるのか興味深いところである[9]。

最後にもう一つインターネットからのデジタル・テクストの入手についても触れておく。インターネットから入手する場合には，FTP を用いるのが一般的であろう。それ以外にも，個人的にホームページを開設しテクストを公開してある場合もある[10]。しかし，後者の場合には，そうしたホームページを探し当てるのが困難であり，時間の浪費に終わる危険がある。また，デジタル・テクストへのリンクを集めたホームページもあるが[11]，必ずしも必要なものが見つかるわけではない。

一方で，英文学作品のデジタル・テクストは，冒頭で紹介した OTA が有名であり，非常に多くの作家の作品を集め充実している。しかしフランス文学作品のデジタル・テクストを大規模に収集し公開しているサイトはまだない．今後フランス文学作品についても，こうした FTP サイトなどの設立が望まれるところである。

Ⅱ.2 テクスト処理の準備

研究対象となる作品などのデジタル・テクストを入手したとしても，それをそのまま使うことはできない。もちろん，PC 上で単にワープロ・ソフトで検索するというだけならば問題ないが，語数を調べたり，ある語彙の頻度を調べたりするのであれば，以下で述べるような処理が必要になってくる。

まず，語数を調べるときに重要になってくるのが，élision している語を apostrophe の後で次の語から切り離すことである。élision をおこしている語は，次に続く単語とは本来別のものなのだから，apostrophe の後にスペースを入れることにより，一つの単語として区別しておく必要がある。この処理を怠ると検索をする際に，例えば≪une école≫と≪l'école≫が別の単語として認識されてしまう。また，≪l'≫や≪qu'≫などの élision している語を直後の単語とスペースで区切っておけば，ある作品の総語数を数えるときにも≪l'≫や≪qu'≫を一単語として数えてくれるようになる。élision をおこす場合は，フランス語では非常に多いため，こうした作業はテクストを処理する際に重要である。

また，≪,≫や≪;≫などの signes de ponctuation も，必ずそれらの前後の単語とスペースなしで並べておいた方が良い。これにより，語数を数えるときに，誤りを少なくすることができる。

次に，これは文学作品のコンコーダンスを作る場合には特に重要であるが，デジタル化を進める際に使用したエディション，あるいは研究において参照したいエディションに合わせて，適宜改行などを入れる必要がある。語学研究では，用法や用例などを調べることが主になるため，それがどのエディションの何ページの何行目にでてくるかはそれ程重要ではないかもしれない。次章で述べる KWIC (Key Word In Context)[12] 検索が語学研究に良く用いられているのも，そのような背景がある。しかし，文学研究においては，やはり，エディションや出現ページの情報は重要である。その場合，あらかじめデジタル・テクストが参照エディションのどこにあたるのかマークをつけておかなければ，テクストを処理した後でそうした情報を付け加えることはできない。特にコンコーダンスを作成するような場合には，処理されたテクストはアルファベット順に並ぶわけであるから，ページなどの情報なしでエディションの該当個所を探し出すのは非常に困難だといえよう。

最後に，アクサン文字の問題を取り上げることにする。フランス語では，アクサン付きの文字があり，それらを PC 上で扱えるようになったのは比較的最近のことである。しかも，PC によって，さらには使用するアプリケーションによっても，互換性がないこともしばしばである。研究にデジタル・テクストを用いる場合には，できるだけ汎用性の高い形式で保存することが望ましい。というのも，テクストは個人で利用するだけではなく，研究者で共有することが臨まれるからである。マルセル・プルーストの *À la Recherche du temps perdu* のデジタル・テクストが存在するのであれば，多くのプルースト研究者はそのテクストを使いたいと思うだろう。そこで，Aという研究者は UNIX を使い，Bが MacOS を使い，Cが Windows を使っているとして，この三者で *À la Recherche du temps perdu* を共有するとする。この場合，この三者に配ることができる共通のテクスト形式というのは，現在のところ存在しない[13]。そうなると，三人分のファイルを作らなければならない。

この問題を回避する一つの解決策は，フランス語を ASCII の文字セットだけを用いて記すようにするということである。ASCII 文字セットは，いわゆる英数字と特殊記号だけからなる文字セットで，コンピューターの文字セットの中では，最も基本的でかつ最もシンプルなものである。むろんフランス語などのアクサン付き文字は，入力したり出力したりすることは出来ない。そこで，頻度の少ない特殊記号をアルファベットに組み合わせて使えば良いのである。例えば，≪é≫は≪e+≫で，≪â≫は≪a^≫のようにである。このようにした場合，*À la Recherche du temps perdu* の冒頭の一節は次のようになる。最初に述べた élision に対する処理も併せて示しておく。

Longtemps, je me suis couch e+ de bonne heure. Parfois, a # peine ma bougie e+teinte, mes yeux se fermaient si vite que je n' avais pas le temps de me dire:≪Je m' endors.≫[14]

話者の自問に先立つ≪de me dire:≫という部分に見られるように，≪:≫の前にはスペースがなく後ろにスペースを置いている。このことは guillemet にもあてはまり，こうしておくと語数をより正確にカウントすることができるのである。

またここに示した，文字の書き換えの法則はあくまで個人的なルールであるが，用いる記号は本文にでてこない限りで任意の記号で良い。むしろ気を付けなければならないのは，ABC順で並べかえすることを考え，記号をアルファベの後に置くこと。そして，いったんルールを決めてしまったら，一貫して符号を付すことである。

Ⅲ. UNIX 上でのテクスト処理ツール

UNIX 上の標準的なエディタとして vi や ed などがあるが，これらは多言語対応プログラムではないためアクサン付きの文字を入力することができない。しかし，GNU プロジェクト[15]によって開発されたMule[16] は多言語に対応しており，フランス語の入力が可能である。そこで，本章では先ず，Mule でフランス語を入力する際の操作を説明する。

また，文字列の検索は，テクストを対象とした処理で最も頻繁に用いられるものであろう。「パターン・マッチング」と呼ばれる操作が UNIX などではあるが，パターン・マッチングを用いる場合には，「正規表現」という独特な検索方法が採られる。それ故，「正規表現」の書式についても本章で扱うことにする。そして，「正規表現」によるパターン・マッチングをするための UNIX コマンドである grep についても説明する。

Ⅲ.1 Mule

vi や ed といったエディタとは異なり，Mule[17] は前述したように多言語に対応している点で，特徴的なプログラムである。Mule は元々 GNU プロジェクトによる Emacs というプログラムに多言語機能を付加したものである。Emacs では，文章の作成だけではなく，WWWのブラウズや電子メールの受送信，ニュースグループの閲覧なども可能であり，この特徴は Mule にも継承されている。また，Mule では，英語と日本語の他に，中国語，韓国語，タイ語，ベトナム語，そしてヨーロッパ系のアクサン付きの文字を用いる言語を扱うことができ，これらの言語を一つのファイルに混在させることもできる点が特徴である。Mule 自体は，Emacs Lisp というプログラム言語で書かれており，Emacs Lisp を用いることで，Mule の機能を拡張することもできる。さらに，Mule 上でシェル・コマンドを操作することもできるため，UNIX で Mule を利用する場合には，シェルと Mule を切り替える必要がない。このように非常に多機能な Mule であるが，実際に Mule でフランス語を使う方法は，あまり知られていない。それ故，本節では Mule でフランス語を入力する方法についてみていくことにする。

先ず，UNIX サーバーにアクセスする必要がある。telnet 端末がある場合には問題ないのだが，ネットワークにつながった PC を利用して UNIX を使う場合には注意が必要である。PC

から telnet で UNIX を使う場合，Windows95／98 であれば，「スタート」ボタンから「ファイル名を指定して実行」を選択し，でてきたウィンドウに「telnet」と入力する。MacOS であれば，NCSAtelnet というアプリケーションを使う。

しかし，実際には，これでは PC から telnet 経由で Mule を利用し，フランス語を入力することができない。これは telnet アプリケーションの扱う文字コードが，フランス語の入力を可能にする規格である「ISO 8859-1」[18] に対応していないためである。このため，Windows では，例えば TeraTermPro など OS に標準添付されているものとは別の telnet アプリケーションを使用する必要がある。MacOS でも，NCSAtelnet の日本語版は「ISO 8859-1」を理解することができない。それ故，MacOS では，NCSAtelnet の英語版を使う必要がある。これら二つのソフトは，インターネットで入手可能であり，PC から telnet 経由で UNIX を使い，英語や日本語以外の言語を扱うときには必ず必要である。

これらの telnet アプリケーションを使って，UNIX にログインするときには，アプリケーションの設定を確認しておく必要がある。前述したように，フランス語の入出力が可能になるためには「ISO 8859-1」という文字セットが必要である。そのため，telnet アプリケーション側でも「ISO 8859-1」を使用文字コードとして設定しておく必要がある。NCSAtelnet であれば，「Session」メニューの「Translation」から「ISO 8859-1」を選択し，TeraTermPro であれば，「Setup」メニューの「Terminal」を選択するとでてくるウィンドウで「Character Set」の「client」を「ISO 8859-5」に設定する。

さらに，telnet アプリケーションの表示用フォントとして日本語フォントを指定したのでは，文字コードが正しくても，フランス語は正しく表示されない。それ故フォントを Windows，Mac OS 共に，アクサン付き文字を表示可能なフォント，例えば≪Courier≫などに指定しておくことも必要である。

これによって，Mule でフランス語を入力するための準備が整った。

では実際に Mule でフランス語を入力するにはどうすれば良いのか。先ず，Mule を UNIX サーバーにアクセスして起動する必要がある。使用するサーバーによってパスが変わるので，この点注意が必要である。しかし，多くの場合，PATH 環境変数に，Mule がインストールされているディレクトリが指定されているため，通常はプロンプトで mule と入力すれば，Mule が起動する。

```
$ mule
```

これで Mule が起動するのであるが，このままでは，アクサン付きの文字の入力はできない。Mule は多言語に対応するために，それぞれの文字セットをパッケージにし，さらに同系列の言語を一つのファイルにまとめて保存している。デフォルトでは，このファイルは Mule 本体に

ロードされていないため，使用に際して必要なファイルをロードする必要がある。Mule では，≪quail-latin≫[19] という名前の文字セットを収めたファイルを用いることで，フランス語の入力可能となる。

次に，“Ctrl” キーを押しながら “]” を入力する。ここで，≪quail-latin≫では複数のパッケージを含んでいるので，“Esc” キーを押して “s” を押す。するとどのパッケージを使用するか聞いてくるので，“french” と入力する。

これで Mule でフランス語を利用する際の下準備は終了である。下の図は Windows95（図1）と MacOS（図2）上でそれぞれ TeraTermPro と NCSAtelne（英語版）を使って Mule を起動したところである。

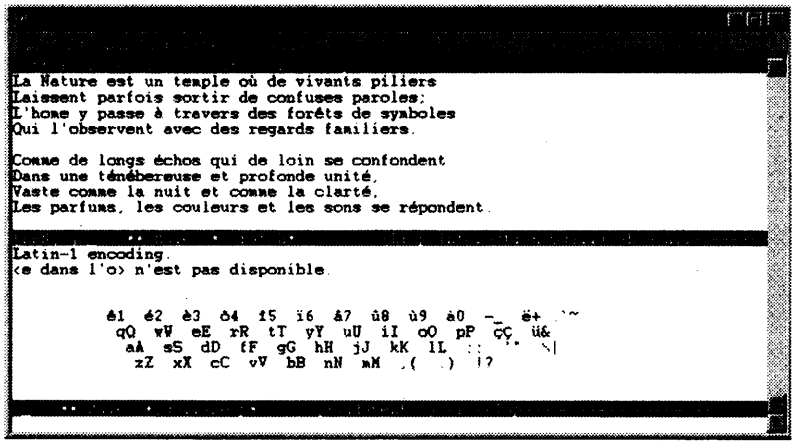

図1

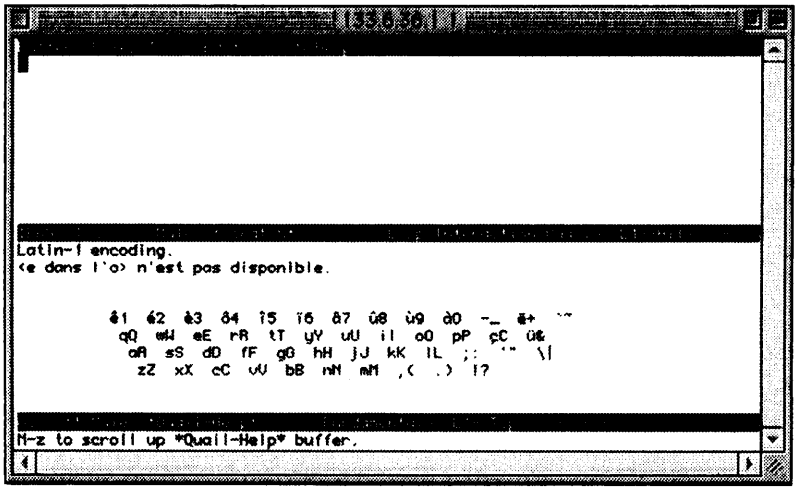

図2

フランス語のパッケージが選択されていることは，画面左下角に「FRANÇS」と表示されていることから分かる。また，図に表示されているのは，図1の上段は telnet アプリケーション

を使って入力したフランス語の文章，下段がどちらもフランス語を使う際のキー配列である。

このように，Mule をみてきたが，PC 経由で Mule を使う場合には，問題点もある。図3は Mule で作ったフランス語のテクストをシェル上で more コマンドを使って表示したところである。ここで分かるように，Mule で入力したフランス語は Mule 内部でしか使えないのである。シェルでフランス語を表示できるようにすることはできるが，そうしてしまうと，日本語が表示されなくなってしまう。目の前に UNIX 本体をしながらコンソールから利用する場合にはこうした問題は起こらないのであるが，そういう環境は一般的ではないだろうから，理論的には可能な多言語混在も，実際に PC から利用する場合には問題があることが分かる。

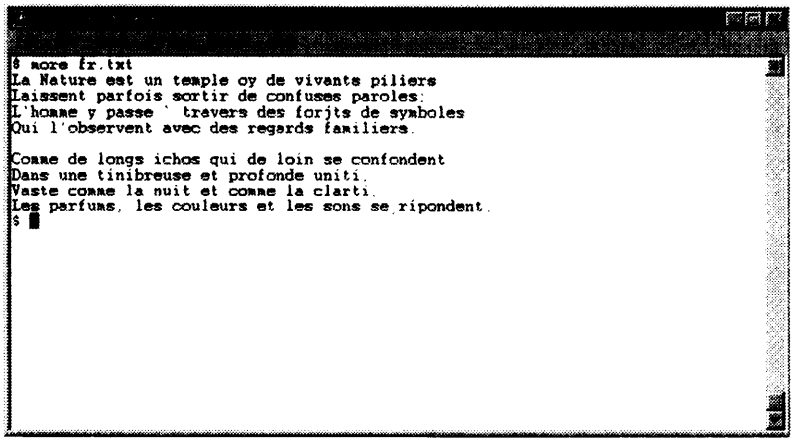

図3

ところで，ある語彙の検索は，文学研究に寄与するところが大きいと考えられるが，Mule にインプリメントされている検索法は「インクリメンタル・サーチ」と呼ばれるものである。これは，現在のカーソルの位置から前方あるいは後方へ向かって検索をし，該当する語彙があるとその語彙にカーソルが移動するというものである。しかし，この方法では検索履歴を残すことが困難であるため，テクストが大きくなればなるほど利用価値が下がってしまうのである。

結論としては，確かに Mule は多言語に対応した多機能なエディタであるが，フランス語が入力できることで文学研究へ応用するのは，むしろ避けたほうが良いといえる。Ⅱ.2 で述べたように，「ISO 8859-1」規格の文字セットを用いなくとも，ASCⅡの文字セットでフランス語を入力することを考えるべきであり，そうすることによってシェル上でも問題なくテクストを参照できるだけでなく，UNIX やそれ以外の PC とのファイルの互換性を高めることができるのである。

Ⅲ.2 grep と正規表現

UNIX のシェル上で使用することのできる検索ツールとして grep を挙げることができる。

これは，検索するパターンにあわせて指定されたファイルを検索し，マッチした全ての行を表示する。また“-n”オプションを指定すると，マッチした文字列を含む行の行番号をあわせて表示する。使用法は簡単で，シェル上で，プロンプトに続いて，grep と入力し，スペースを一つ空けて検索したい文字列（あるいは文字パターン），そしてスペースをおいて次に検索したいファイルの名前を入力すれば良い。次の図は，*Roman de Renart, édition γ* の第二章のデジタル・テクストから≪Renart≫および≪renart≫という語を grep で検索し，結果を表示したものである。

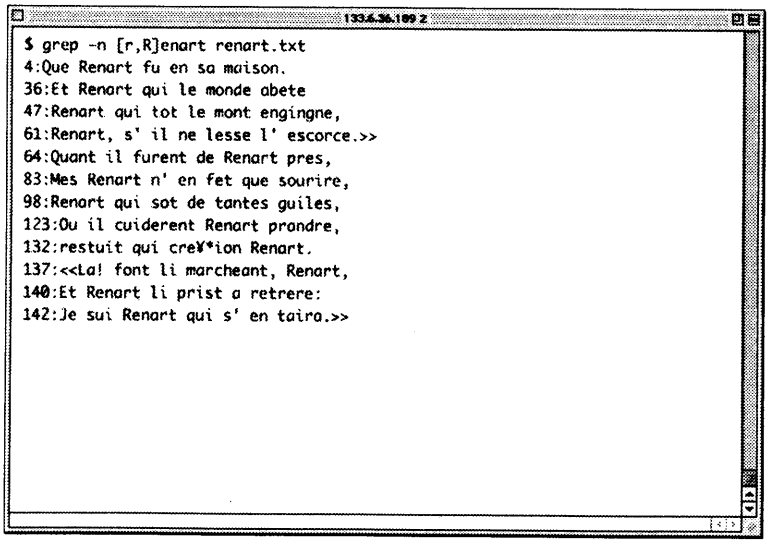

図4

プロンプト“$”の後に“grep -n [r, R] enart renart. txt”と入力して検索を始めている。これにより≪renart. txt≫というファイルの中から≪renart≫と≪Renart≫を含む行がその行番号と共に表示されているのが分かる。ここでは，検索結果を画面に表示させたが，結果をファイルに保存することもできる。図4の検索結果を≪result≫というファイルに残すのであれば，次のように入力して結果をファイルにリダイレクトすれば良い。

```
$  grep -n [r, R] enart renart. txt > result
```

ところで，上にあげたコマンドラインで，検索文字列を≪[r, R] enart≫と指定している[20]。これは，≪renart≫と≪Renart≫の両方にマッチさせるための書き方であり，正規表現という。正規表現は次表[21]のように定義できる（但しここで P や Q はそれ自身正規表現である）。

文字	その文字自身
PQ	P と Q をつなげたもの
$P \mid Q$	P または Q
$P*$	P の0回以上の繰り返し
(P)	p

こうした書き方は，元来プログラムを書く際に使用される検索の方法である。プログラミング言語が各々違う文法を持っているように，正規表現も以上のような定義を基礎に持ちつつも，実は微妙にその文法が違っている。上に挙げた grep の例は，UNIX シェル上の正規表現の例とは異なる[22]。

正規表現を用いることの利点は，語形が変化する単語を網羅的に検索できる点にある。無論，まったく語形が変わってしまう単語，例えば動詞≪avoir≫をその変化形である≪ont≫に結びつけることは，正規表現を使ったのではできない。しかし，正規表現による検索を使えば，例えば≪des-≫という接頭辞がついた単語や，また，語尾が≪-ere≫のものだけを見つけることもできる。これらの場合には，次のように検索すれば良い。また，検索の結果を図5に示す。

```
$ grep -n ' des.' renart. txt
$ grep -n ' des.' renart. txt
```

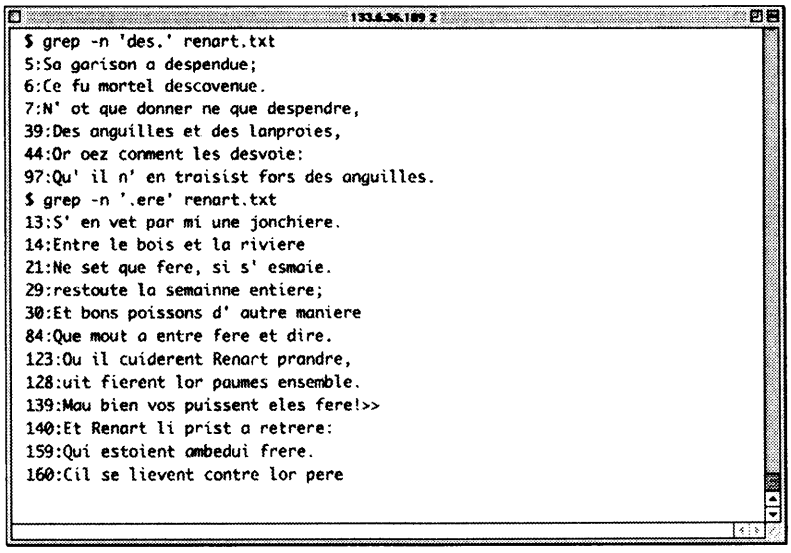

図5

この例では，“.”（ドット）は「0回以上の任意の一文字」を示している。そのため，上の検索では≪des≫自身にもマッチしているのが分かる。

このように，grep コマンド，および正規表現を使うことによって，より強力な検索を実行できることが分かる。

Ⅲ.3 その他のツール

Mule や grep の他にも，文学研究に応用できるコマンドが UNIX にはある。comm コマンドは二つのファイルに共通する行を表示するし，unique コマンドは，ソートされた二つのファイルに重複した行を表示する。さらに unique コマンドでは重複していない行を表示したり，重複回数を表示させることもできる。また，wc コマンドは，指定した一つ以上のファイルの行数，語数，文字数を表示する。その中で sort をここでは取り上げる。

sort コマンドにはバージョンによって異なる様々なオプションがあるが，その機能は ASCⅡ コード順に並べ替えることである。ASCⅡ コードを sort した場合には，次のような順序になる。

!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`
abcdefghijklmnopqrstuvwxyz{|}〜[23]

さらに sort では隣り合う空白以外の文字の集まりをそれぞれ「欄」と見なし，「欄」ごとに並べ替えることができる。図6は，*Roman de Renart* の第二章の最初の三行を第一欄でソートしたものと第二欄でソートしたものである。

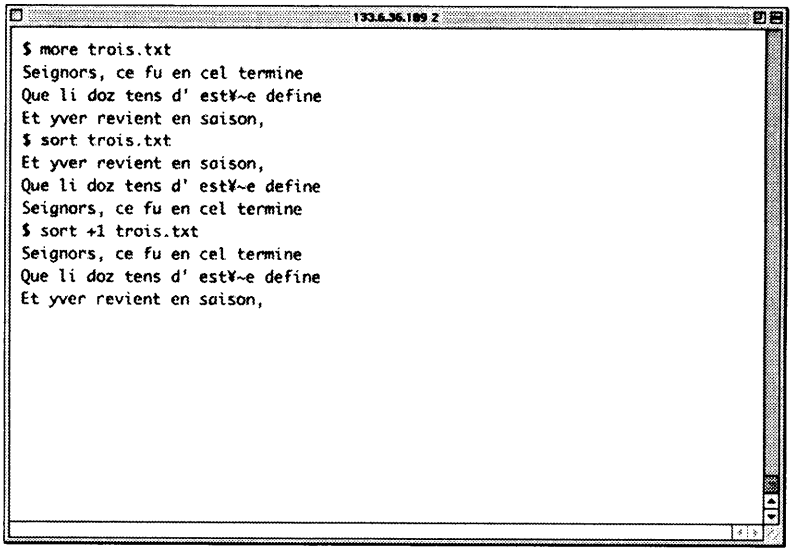

図6

こうした操作は，冠詞を除いて次の単語を基準に並べ替えるときなどに利用できるだろう。

Ⅲ.4 来るべきテキスト処理ツール

以上みてきたように，UNIX に標準のツールの中にも文学研究に応用可能なものがあることは分かった。しかし，文学研究へのコンピューター活用の主流がコンコーダンスの作成などにあることを考えると，これまでに見てきたツールでは，必ずしも十分であるとは言い難い。

上田博人氏は，『パソコンによる外国語研究(Ⅰ)』[24] のなかで，「度数分布」，「標準偏差」，「相関係数」，「クラスター分析」，「パターン分類」などの統計学的手法を外国語研究に応用している。上田氏の研究対象が語学であるため，そこで用いられた手法のすべてが，文学研究に応用できるとは必ずしも限らない。しかし，ある作家の全作品を対象として語彙分析を行う場合には，このようにして数値化され視覚化されたデータが，文学研究に役立つことは間違いない。

上田氏は，これらのデータを得るために，jawk というプログラミング言語を用いて，自らプログラムを作成し，研究に活用している。ところで，UNIX のツールや PC のソフトウェアを用いただけでは，柔軟に文学研究に応用できるわけではないことが分かった。

文学研究において必要なのは，コンコーダンスを作成するプログラムであり，複数の作品からある語彙を検索してくれるプログラムであり，検索で見つかった語を数値化して提示してくれるプログラムである。

KWIC（Key Word In Context）と呼ばれるコンコーダンス作成プログラムがある。しかし，このプログラムはマッチした単語のコンテクストを表示してはくれるが，その語の属するページ数などエディション固有の情報は示してくれない。その点で，語学研究には向いているが，文学研究には不向きである。

また，UNIX や PC の区別なく利用することのできるプログラムも必要である。contexpro[25] は，UNIX でも PC でも動作するが，処理に時間がかかる。また，データの数量化はできない。

これらの特徴をすべて備えたプログラムがあるのなら，文学研究に躊躇することなく活用することができるだろう。すなわち，エディションを参照するようなコンコーダンスを作成し，さらに複数の作品にまたがってコンコーダンスの比較を行うことができ，そして統計処理も行ってくれる，そういう統合プログラムこそが，来るべき文学研究に応用可能なプログラムとして考えることができるのである。

結　論

UNIX でも PC でも動作する統合プログラムという考えは，非常に魅力的である。しかし実際には，そうした多機能なプログラムは現在のところ存在しない。しかも，そうしたプログラムが，研究者が現在進めている研究に有用であるとは限らない。例えば，サルトルにおける「愛」というテーマ研究をしている場合には，こうしたプログラムは有効かもしれないが，詩を研究対象としている場合には，コンコーダンスや語彙の統計データの他にも，韻律のデータなども重要になってくるだろう。

このように，文学研究においては画一的で汎用的なプログラムよりは，研究者個人の研究に合致したプログラムが必要とされているのである。もちろん，コンコーダンス作成ソフトや語彙の数値化など様々な作品に共通して用いることができるツールもその開発が望まれるところである。しかし，それだけではなく，やはり個別の研究に対応するプログラムの開発も必要だと考える。

そうであれば，これからの文学研究では，研究者自らがプログラムを作成し，自ら望んでいることをコンピューターを用いて実現することが重要になってくるだろう。但し，自ら作成するのであれば，高機能ではなく，機能がそれ程多くないプログラムを作れば良い。そして，UNIX はプログラムを開発する重要なプラットフォームなのである。しかも，UNIX でフランス語を扱う方法については考察した通りである。文学研究にもプログラム開発環境としての UNIX を活用する時期が到来した，といえるのではないだろうか。

注

1）次の URL で参照することができる：http://ota.ox.ac.uk/。

2）Larousse 社から，ラシーヌの *Bérénice* やボードレールの *Les Fleurs du Mal* などが，デジタル化され商品化されているし，Discotext はフランス文学のアンソロジー。前者は MacOS，Windows の両 OS で利用可能であるが，後者は Windows のみで利用可能である。

3）≪ASCII≫は，≪American Standard Code for Information Interchange≫の頭文字をとったもの。*cf.* Ken Lunde, *Understanding Japnese Information Processing*, 春遍雀來・鈴木武生訳，『日本語情報処理』，ソフトバンク株式会社，1995年，p. 40-42，および長尾真・黒橋禎夫・佐藤理史・池原悟・中野洋，『言語情報処理』，岩波書店，1998年，p. 3-10。コンピューターの文字コードについては多くの解説書があるが，ASCII だけではなく，日本語などの2バイトコードについても論じられている。

4）R. THOMAS and J. YATES, *A User Guide to the UNIX System*, 2nd Edition, 東風訳，『UNIX 入門改訂版』，工学社，1993年，p. 21-26。

5）「移植性のよさ」とは，新しいコンピュータが開発された際に，そのコンピュータで動くように UNIX システムを簡単に書き直すことができることを示している。また，「通信や電子メール」が UNIX の特徴であるといえるのは，特別なソフトを用いることなく，UNIX がインストールされているコンピュータをネットワークに接続したり電子メールを利用したりすることができることをさしている。「ライブラリ」は，一般には「アプリケーション・ソフト」と呼ばれるものであり，サードパーティー（ハードウェアおよびソフトウェア会社）によって販売提供される。

6）原野昇・中川正弘・太古隆治・前田弘隆，「中世文学研究におけるコンピュータ利用」，『広島大学フランス文学研究 15』，広島大学フランス文学研究会，1996年，p. 67-76。および，久後貴行，「マッキントッシュ上での単語リストとコンコーダンスの作成」，*Travaux de Linguistique et Littérature Médiévale Français* [TLLMF] 第7号，大阪市立大学大学院文学研究科 TLLMF 研究会，1996年，p. 1-10。

7）*cf.* 原野昇・中川正弘・太古隆治・前田弘隆，前掲書。

8）現在 CNRS で，サルトルの全作品のデジタル化が企画されている。

9）但し，バルザック全集の場合には，オリジナルのエディションと共に，誤植のみを訂正したものなどいくつかのエディションを併録している。

10）本論で用いている *Roman de Renart* のデジタル・テクストは，広島大学文学部フランス語フランス

文学教室のホームページに部分的に掲載されている。http://www.ipc.hiroshima-u.ac.jp/~france。

11) 例えば，次の URL を参照。http://www.planet4u.com/book/search/pages_f/ring.htm

12) *cf.* 長尾真他，前掲書，p. 27-29。

13) Unicode を用いることにより可能ではあるが，まだ一般的ではなく Unicode をサポートしていないアプリケーションが数多く存在する。また，OS レベルでサポートされているわけではない。

14) Macel PROUST, *À la Recherche du temps perdu*, tome I, coll.≪Pléiade≫,Gallimard, 1987, p. 3.

15) GNU プロジェクトは，1984年に始まったプロジェクトで，無償の UNIX 系の OS の開発を目的としていた。プロジェクトは，≪Linux≫カーネルを補完するものであり，現在では UNIX だけでなく，Windows などの OS でも動作するソフトウェアも開発されている。GNU プロジェクトについては次を参照：http://www.gnu.org/。

16) Mule は≪Multilingual Enhancement of GNU Emacs≫の略。

17) Mule は UNIX だけではなく，Windows で動作するものも開発されている。それぞれの OS 用の Mule は，FreeSoftwareFoundation から CD-ROM を購入するか，あるいは GNU の FTP サイト（ftp.gnu.org）およびそのミラー・サイトから入手することができる。日本でのミラー・サイトとディレクトリのパスを二つ挙げておく。このほかにも日本のミラー・サイトは数多くある。

tron.um.u-tokyo.ac.jp/pub/GNU/prep

ftp.cs.titech.ac.jp/GNU

18) ISO 8859 規格は 8 ビットの文字セットを規定したもので，ISO 8859-1 から ISO 8859-10 まであり，1987年から1992年にかけて作成された。このうち，フランス語の属するいわゆる「Latin-1」コード（ISO 8859-1）は1987年に規定されている。

19) ≪quail-latin≫の他にも≪quail-cyrillic≫，≪greek≫，≪hebrew≫，≪chinese≫などのパッケージがある。

20) バークレー・バージョンの grep には "-y" オプション，ベル・バージョンの grep には "-i" オプションがあり，大文字・小文字の区別を無視することができる。

21) 長尾真他，前掲書，p. 39。

22) 正規表現は UNIX でだけ使えるものではない。例えば，Microsoft 社の Word というアプリケーションでも，検索の際に正規表現に近い方法で検索することができるし，MacOS 用のワード・プロセッシング・アプリケーションである Nisus Writer でも，正規表現による検索が可能である。

23) *cf.* R.THOMAS and J. YATES, 前掲書，p. 464。

24) 上田博人，『パソコンによる外国語研究(Ⅰ)』，くろしお出版，1998。

25) ≪contexpro≫については，拙稿，「Perl によるコンコーダンス作成ソフトの開発」，『広島大学フランス文学研究 17』，広島大学フランス文学研究会，1998年（印刷中），参照。