

報 文

CTASSで実現された情報処理教育環境

京都職業訓練短期大学校 電子・情報技術科 塚原 周信

Realized Enviromental Conditions in CTASS for Education of Data Processing

Kanenobu Tsukahara

要 約 従来の情報処理教育システムは、自立的な情報処理技術者を育成するためには不十分である。

情報処理技術者を育成するためには、一斉形式の実習ではなく、個々の実習者の創意と工夫を生かした実習が必要である。これらを支援する実習システムは、統合化された環境で、特定の機種に依存せずに、適切なエラー処理機能と再現性を持つことが必要である。また、容易にシステムを統一的に拡張することが可能なことも重要な条件である。

本論では、これらの条件を考慮したCTASS (Computer Training Assistant System for Students) を紹介する。

I はじめに

情報処理教育においては、教室における講義と同等に、実際に機器を操作しての実践教育が重要である。しかし、現実には情報処理教育の実習を行うべき環境の明確なイメージは固まっていない。計算機を使用した情報処理教育の環境としては、ビデオテープとなんら変わらないコースウェア教材とか、キーボード練習用のドリル程度しか提供されているに過ぎない。

この情報処理教育用の環境が貧弱な点を痛感して、今回報告する「CTASS」(「シータス」と発音)と名付けた情報処理教育支援システムを開発した。

本論では、職業能力開発教育において、初心者が陥り易い点と、それをCTASSを用いた情報処理教育環境によって補う方法を議論する。

なお、本論で扱う情報処理教育の範囲としては各種の言語教育、および言語を使用しての自然科学・工学・社会科学さらに人文科学上の問題解決のためのプログラム作成実習等に限定する。

II 情報処理教育の目標と実習システムの条件

情報処理教育でも講義と実習の関係が問題である。以

上で二つの異なる実習形態について議論する。

第1の形態は、密着した監視下における技能伝達型の一斉形式実習である。

実習をすべて指導者の監視下で一斉に行う方法で、ワープロ教育などで行われており、実習者が誤りを犯した場合や、操作法が判らずにいる際には、指導者が直ちに対応する。この方法では、実習を行う環境を特別に考える必要はないであろう。

この方法は、ワープロや自動車運転のように技能が重視される分野での訓練には有効であろう。また情報処理機器の使い方などを教える導入教育でも必要である。

しかし現在の社会における情報処理形態の急激な変化のために、原理を理解しないで「丸暗記」した技能は急激に陳腐化する。我々が育成を目指す情報技術者とは、新たな問題を与えられた時に、自分自身で解答を見出す能力を身につけた自立した技術者でなければならない。

そこで第2の形態、実習者の個性を生かした実習が重要になる。

自立した情報処理技術者養成を目的とした実習では、実習の初期の段階から、実習の過程で生じた問題に対して実習者自身が解答を探してゆく訓練が必要である。

この実習方法では、個々の実習者が解答を得るまでの時間が大幅に異なる。一斉形式の実習で個々の実習者に対応するには、実習時間を最も遅い実習者に合わせなけ

ればならない。これは、指導者にとっても他の実習者にとっても、遅い実習者の完成を待つのは負担であり、マンツーマンの指導でない限り不可能である。

実習は、指導者の監視を離れて実習者が自由に時間を設定して行う方法が有効となる。幸いなことに、情報処理教育における実習では、悪条件下に長時間にわたって実習を行った場合に問題になる視力等の保健問題を除いては、問題になる危険は皆無と言って良く、指導者は実習者を常に監視する必要はない。

この場合でも、指導者は実習者が解答を提出するのを待つだけで良いわけではないのは当然である。実習者が誤りを犯したり、独自の工夫を試みたりして、自身では解決できない時には、相談に応じて、個々に応じた解決方法を示さなければならない。指導者は常に監視しているわけではないので、実習者が発生させた問題の原因を正確に推定する作業が重要になってくる。

このような実習方法を採用した場合には、計算機システムの実習環境の良否が極めて重要になってくる。そのシステムの持つべき条件として以下に示す 6 項目が重要である。

1) 統合化環境

情報処理実習システムの働く環境としては、大別して、マイクロコンピュータを使用したシステムと、メインフレーム・コンピュータを使用したシステムがあり、現状では両者には一長一短がある。両者の長所を生かすために、両者を有機的に結合したシステムが望ましい。

2) 特定の機種に依存しないシステム

実習者は、機種固有の煩雑で統一性のない JCL (Job Control Language) やコマンドを初期の段階では覚える必要がない。それらの原理的役割を理解していれば十分である。

3) 適切なエラー処理

初心者の段階から実習者が独自で使用できるシステムが望ましい。そのためには、操作が容易で、実習者の誤りには適切なメッセージが出る必要がある。

重要な事は、誤りを誤りとして認識させ、実習者に対処を促すシステムでなければならない。初心者は、計算機の出力するものは何でも正しいと思いがちで、出力結果を検算するなどの検討をしない。

実習者のプログラムに欠陥があるために、プログラムや OS を破壊して暴走したり、暴走しないまでも、何のエラーメッセージも出力されないにもかかわらず、結果が全く信頼できない事態が生じるのを、初心者は知らない。初心者を対象とした教育システムでは、このようなエ

ラーが発生した場合の処理が重要である。

4) 再現性

実習者の犯した誤りに再現性がなければならない。実習者が誤りを犯した後で、わけも判らずに再度試みると、今度は誤りが発生しないという事例が多々ある。また、指導者に相談して、指導者の監視下で再実行をすると誤りが発生しない、という事もある。これでは、どのような誤りを犯すと、どのような事態になるかを実習者自身で、把握するのに困難が伴う。

5) 記録と通信機能

指導者は実習者がどの学習段階にあるかを把握し、また実習者からの相談に適切に対応するためにも、実習状況のできるだけ詳細な情報を記録できることが望ましい。これらを統一的に管理するには通信機能が必要である。

今回目標とする教育支援システムでは、一斉形式でない演習を支援することが目標なので、商用のシステムと比較するとリアルタイムなモニタリング機能はそれほど重視されず、記録性がより重視される。しかし導入教育では一斉形式の演習も有効であり、商用のシステムと使い分けをしなくともすむように、目標とするシステムでも基礎的なリアルタイムなモニタリング機能は要求される。

またメール機能を利用して質問し解答を得られたり、「穴埋め」的な演習問題のファイルを取り込む機能、レポートの提出機能などは、一般的 C A I システムと同様に備えることが望ましい。

6) 統一性と拡張性

実習システムは統一性と拡張性がなければならない。Fortran だけしか、Cobol だけしか、あるいは特殊な仕様の c だけしか使用できないシステムで、各言語ごとに別なシステムを使用するのは、実習者の負担が増すだけである。既存の主要な言語、およびこれから将来にわたって開発される種々の言語（簡易言語等も含める）に対して、システムの大幅な変更をせずに簡単な操作でシステムに組み込めて、すべて同じ操作で扱える統一性と拡張性が必要である。

とくに提供される操作性の面のみを注目して、ある特定の一社だけの規格のベーシックや c だけしか動かないシステムで教育を行うのは問題がある。しかしながら商業ベースのシステムでは、自社やグループ内で開発したソフトの世界に閉じこもる傾向が強く、教育訓練機関などの第三者がシステムを開発することの意義がある。

III C T A S S で実現された機能

前節の条件を考慮して教育支援システム C T A S S が

開発された。

CTASSは富士通のメインフレーム・コンピュータにおいて、同社製のパーソナル・コンピュータFMRシリーズを端末として使用しているシステムに対して開発されたものである。なお、パーソナル・コンピュータだけの運用も可能である。CTASSでは、メイン・フレーム（以下ホストと呼ぶ）側と、パーソナル・コンピュータ（以下ローカルと呼ぶ）側のそれぞれで制御プログラム（シェル）を稼働させ、両者が協調して実習を支援する。現在のところ、ホスト側の（制御）プログラムはF4/MSPで、ローカル側はMS-DOS配下で稼働する（稼働OSをホスト側ではFSPにローカル側ではMS OS/2に拡張する作業を実行中である）。CTASSでは、前節の条件に対して以下の方法を採用した（なおCTASSのプログラムの内部構造等については別報する）。

1) システムの起動

ローカルのパーソナル・コンピュータの電源を入れること等によってCTASSが起動されると、CTASSのメニュー画面〔図1〕が表示される（以下このメニューをメイン・メニューと呼ぶ）。

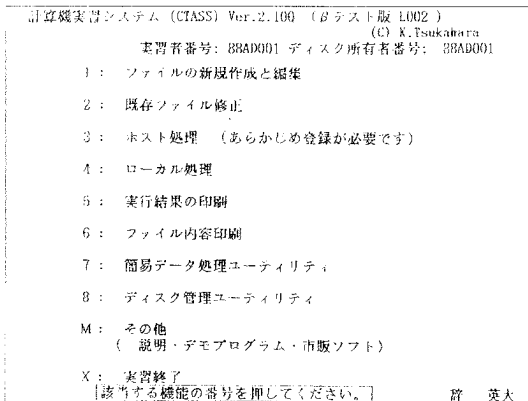


図1 CTASSのメニュー

CTASSではフロッピー・ディスクを必ず使うので、起動の際にフロッピー・ディスクがセットされていない

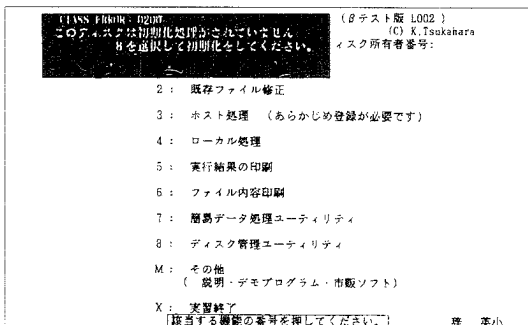


図2 初期化していないディスクへの警告

かったり、フォーマットされていないフロッピー・ディスクがセットされていると、警告のメッセージが表示されて〔図2〕、実習者の操作を促す。なおディスクの初期化は、メニューに従って操作をするとCTASS独自の初期化プログラムが起動する。実習者は「新しいディスクは一回だけ初期化が必要」ということを知っているだけで、他の予備知識を必要としない。これにより、初心者の実習に対して、指導者が初期化済のディスクを配付するという作業は必要なくなる。

2) プログラム等の作成

CTASSでは、実習者のプログラム等はローカル側のフロッピー・ディスクに作成される。このような方法を採用した契機は、CTASSの開発目的がかなり多人数の学生実習を考えていたので、ホスト側のディスクにはプログラムを収容出来なかったためではある。しかし結果として、ホストとローカルを統合した自由で拡張性のあるマイクロ・メインフレーム統合環境を作るには最適の方法であった。CTASSで、この作業を行うにはメイン・メニューで「1」または「2」を選択する。「1」と「2」の違いは、「1」では新規にファイルを作成してから編集するのに対して、「2」では既存のファイルの再編集をする点にある。あえて2つに分離したのは、実習者が既存のファイルの名前を誤って指定して、別のファイルを新規に作成してしまわないようにするためである。

「1」または「2」を選択することで編集用サブ・メニュー〔図3〕が表示され、新規の場合はファイル名を打

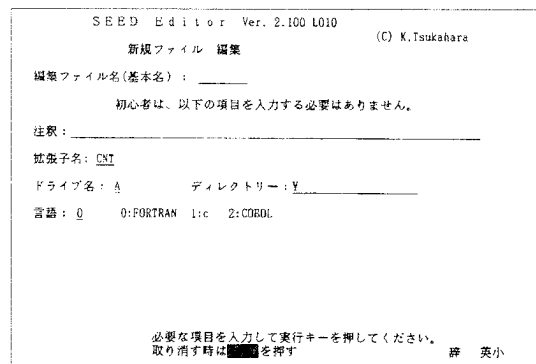


図3 編集サブメニュー

ち込む。当然ながら、既存のファイルと同じ名前を指定した場合はエラー表示をする。さらに、OSが許さない名前を指定した場合もエラー表示をする。これにより、一部のエディターのように、編集内容を保存する段階になって（ファイルが作れないと）エラー表示され、対処法が判らずに初心者が闇雲にキーを押して編集内容を消失させるという事故が防げる。

また「2」を指定した場合は、デフォルトとして直前に指定したファイル名が表示される。実習者は、必要があればデフォルトのファイル名を変更して、必要がなければデフォルトのファイルを編集する。

編集サブ・メニューで「実行キー」を押すことにより、ファイルの編集モードになり、編集画面が表示される(図4)。

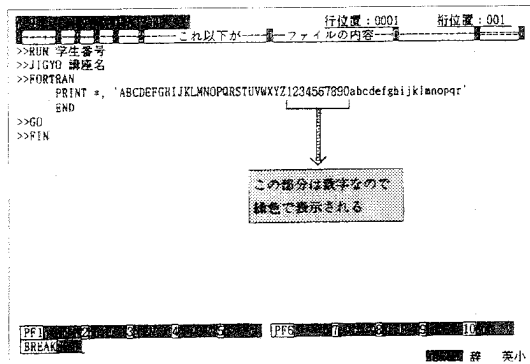


図4 編集画面

3) 編集機能

CTASSのエディターSEED (Simple Easy Editor)は、現在のところ最小限の機能しか有していないが、実習者の誤りを防ぐ独自の工夫をしている。

その中でも最も好評の機能は、文字の種類(英字・数字・全角文字)ごとに表示色を変更できる点である。初心者には、英字のO(オー)と数字の0(ゼロ)、小文字のl(エル)と数字の1(イチ)等を混同する誤りが多い。この誤りは機種によっては画面でも印刷でも非常に判りにくい場合があるのに対して、SEEDでは一目瞭然である。

またカナ漢字変換の機能がサポートされたことにより、気軽に漢字が使用できるようになった。これにより、全角文字と半角文字の混同による誤りが増えた。現在、全角文字に対して別の表示色を選択できるような機能を試験的にサポートして誤りを防ぐ試みをしている。

4) 実行ファイルの形式

CTASSの実行ファイルの形式は原則として、プログラムとデータを同一のファイルに記述して、その間を予約された記号と文字列を記述した特別のレコードで分離する形式になっている。CTASSでは、この特別なレコードを「JCL」と呼んでいる。この命名は、バッチ処理におけるJCLとの類推からである。CTASSでは、一度このファイルを作成すれば、(メニューで)実行ファイルを指示するだけで、プログラムのコンパイル・リンク・実行を順次自動的に行う「バッチ形式」になっている。

この形式を採用した理由は、CTASSの原型ともいえるシステムが稼働した計算機ではバッチを主体として実習を行っていたので、そこで稼働する教育システムもバッチ形式のシステムになった¹⁾。この旧システムの運用経験からCTASSを開発する際にも、II節(4)で述べた再現性を保つために(TSS配下で稼働しているにもかかわらず)「バッチ形式」を採用した。

これにより、コマンドのタイプミスやデータの入力ミスなど、再現性のない誤りを排除することができた。学生はプログラムの作成は熱心でも、データに対しては無関心で、入力したデータの値を忘れてしまう事も多々あるし、データによってプログラムが正常に動作したり、異常が現れたりすることを理解していないことが多い。

しかし、データを会話的に入出力できないのでは、プログラムの目的によっては不都合なので、データの会話的

《JCLの種類と機能のまとめ》

言語の翻訳と実行に関するJCL

- H.L. >>FORTRAN Fortranの翻訳
- H.L. >>COBOL COBOLの翻訳
- H.L. >>Pascalの翻訳
以上オプション NOSCOURCEなどが可能。
- H.L. >>SPSS SPSSの実行
- H.L. >>ANALYST ANALYSTの実行
- H.L. >>GO プログラムの実行
オプション T 標準入出力を端末にする。
なお、入力のリダイレクション機能を使用できる。

以上のJCLではオプションに「<ファイル名>」を指定することでコンパイラあるいはユーザー・プログラムの入力を、指定されたファイルから行う。現在ローカルでは、「リダイレクション」が可能なのは、「>>GO」だけである。

- H.L. >>LIB ライブラリーの結合を行う。このJCLは「>>GO」の前に、置かなければならない。

その他のJCL

- H.L. >>RUN オヘラントとして学生番号を記述する。ジョブの始まりを示す。
- H.L. >>JIGYO オヘラントとして受講する講座の略号を記述する。ジョブの始まりを示す。
- H.L. >>FIN ジョブの終わりを示す。
- H.L. >>ALLOCATE ファイルのアロケーションをする。他人のファイル等を入力する場合などに使う。

【注意】

- 先頭に が付いているJCLは、その直後にJCLでないレコードを置くことが許される。
- 先頭に が付いているJCLは、その直後にJCLでないレコードを置くことが許されない。その直後はJCLでなければならない。
- また、先頭の「H」「L」は、そのJCLが、それぞれホスト、ローカルで実行可能なことを示す。

表1 「JCL」の例：文献3から一部を引用

入出力モードを可能にするオプションもサポートしている。これら機能については「表1」に示す。なおこれらの「JCL」は、文献2を生かすように構成されている。

5) プログラムの実行

プログラムの実行は、編集を終了させてから、メイン・メニューで「3」または「4」を選択することで行える。「3」と「4」の違いは、「3」では、実行コンピュータとしてホストが指定されるのに対して、「4」では、ローカル（パーソナル・コンピュータ）が指定される点である。

プログラム実行のサブ・メニューでは、直前に操作したファイルの名前がデフォルトとして表示され〔図5〕、そのまま「実行キー」を押すと、そのファイルが実行される。

ファイル・実行	
実行ファイル名(基本名):	EXAMPLE
初心者には、以下の項目を入力する必要はありません。	

拡張子名:	CNT
ドライブ名:	A ディレクトリ:
実行計算機種別:	L H: ホスト実行
リジョン・サイズ:	1024
ホストによる実行はあらかじめ登録が必要です。	
必要な項目を入力して実行キーを押してください。	
取り消す時はESCを押す	
詳 英大	

図5 実行サブメニュー

プログラムの実行過程を次に示す。

まず、「JCL」を手掛かりにCTASSは、実習者のファイルからプログラムとデータを取り出し別々に格納する。次に指定のコンパイラ（状況により異なる）を呼び出してコンパイル・リンクを行い、この段階でエラーが発生しなければ、ユーザー・プログラムを実行する。この間、コンパイラおよびユーザー・プログラムの出力は原則として、画面（プリンター）ではなく、ワーク・ファイルと結合される。CTASSは、実行の各段階をチェックして、発生したエラーに対応したメッセージを挿入する。

なおホストでの実行を指示した場合には、ファイルのコード変換（ここでも独自の工夫をしている）をした後に、ホストへファイル転送し、ホスト側のシェルが前述の処理を行い、結果を端末側に転送する。

もちろん実習者には以上の手順は見えず、ローカルあるいはホストのシステム自身が「JCL」を処理しているように見える。

以上の説明で明らかなようにCTASSでは、実行コンピュータはメイン・メニューで指示するだけで変更が

可能で、（プログラムが国際的規準を守っている限り）他の変更を全く必要としない。この点がCTASSの特徴である。この機能を使用して、ローカルの処理では原因をつかめないエラーも、ホストで処理することで原因を解明することが可能になる。

実行処理が終わると、メイン・メニューが再び表示される。

6) 実行結果の印刷・表示

プログラムのコンパイル・リストおよび実行結果は、メイン・メニューで「5」を選択し、サブ・メニューでパラメータを入力することで表示／印刷することができる。

なお、プログラムを修正した後に、実行処理をせず（テキスト・プログラム・ファイルの方が、実行結果ファイルより新しい場合）に実行結果を表示しようとする等の誤操作をした場合には、警告のメッセージが表示される。

7) 各段階の明確化

プログラムの編集・実行・表示を、このように3つの段階に分離した理由は、（1）実行結果がプログラムの進行と共にスクロールされ消失してしまわずに、実習者が実行結果を十分に時間をかけて検討できるようにするため、（2）種々のコンピュータに対して、同一の操作性でプログラムを実行することが可能であるため、（3）実習者に対して、プログラムの作成・実行・結果表示の3つの段階の違いを確実に認識させるためである。

8) システムの拡張性

エディターSEEDを除いて、基本的にCTASSは、ユーザーからの入力をシステム固有の方式に変換するシェル・プログラムである。そのため、新たな言語やシステムに対して拡張するのは比較的簡単である。現在は新たな「JCL」の追加はプログラムの変更で行っているが、将来の課題としては、「JCL」を定義する簡易言語の導入を検討している。

拡張が困難なのは、入力を会話型の操作で行う簡易言語やSmalltalk等への対応である。これらをCTASSのメニューから起動させることは可能だが（一部分は実現している）、誤りの指摘などのメッセージ出力やデバックを統一的な方法でCTASSが支援することは不可能である。

なぜなら、これらのプログラムではユーザーとのインターフェースを、BIOS割り込みや、直接IOを操作することで支配しているため、CTASSのようなシェルが、ユーザーの操作中に（誤りを指摘するなどの）CTASS固有のメッセージを表示する事は不可能である。これを解決するには、OS自体を創造せねばならない。単な

る操作のトレーニングではない、実際的な情報処理を実習するシステムのOSを作成するのがどんなに困難かはTRON計画の例を見れば明らかであろう。

9) 記録機能

現在のCTASSでは、ホストの実行に対してのみ、すべてのリストとエラー・メッセージ、プログラムの出力のログを採っている。ローカルの実行に対しては、個々のファイルに記録されたログを集める手間の問題もあり、行っていない。LAN機能を利用してログの採取と実行のリアルタイムのモニタリングを行う必要があるだろう。

これとは別にCTASSでは、実習者のフロッピー・ディスクに実習者番号と各ファイル毎に修正の記録と実行回数を記録するようになっている。これは、学習の記録であると同時に実習者間でディスクの貸し借りを防ぐ為であったが、問題の提示方法を工夫すると、後者の機能はあまり必要としなかった。その他にも実習者の不正を防ぐ機能が用意されているが、ここでは触れない。

IV 検討

1) 運用上の評価と課題

以前に、学生に対して1時間弱の講習を行った後に³⁾、CTASSを使って自由に実習をさせる、という運用をした。その程度の講習でも機器の操作については、あまり問題は生じていない。

初心者教育は操作性の観点からインタープリターが望ましい、と主張するむきもあるが、学生の理解度を見る限りは、単一の言語・計算機のみを学習している場合でもCTASSはインタープリターにひけをとらない。そして実習者が、複数の言語を学習したり、ローカルとホストの両方の計算機を利用する段階になると、以前に学習した操作方法が生かされるという点でインタープリターよりも優れている。

しかし操作の軽快感が「QUICK BASIC」などに比べると少ないのは短所である。この傾向は、特にホストを使用した場合に感じられる。この点は今後の改良点である。

唯一の操作上の問題点は、プリンターの紙送り不良や紙づまりである。紙送り不良の場合は、紙をセットしなおして、印刷を再度指示すれば良いが、紙づまりの場合は、指導者や指導補助員の介入が必要になる。

このように、CTASSは初心者用教育支援システムとしては初期の目的をある程度達成したが、職業訓練短期大学など、より広範囲での使用を目的とするには、改良すべき点も多い。改良すべき最大の点は、プログラムの

エラー対策である。この点は、情報処理システムの永遠の課題であると言えるが、CTASSも例外ではない。

現在のCTASSは、(操作の誤りではなく)プログラムの誤りに対しては完全にOSに依存しており、OSが処理できないエラーは、CTASSでも対処できない。これはシェルの宿命ではあるが、現状を改良する余地も残っているので、プログラムの暴走と「エラー表示されないエラー」について以下に検討した。

2) プログラムの暴走対策

ホストでプログラムが暴走した場合でも(特にMSPでは)、OS自身が破壊されることは考えにくい。ホストでユーザー・プログラムが暴走すると、CTASSのシェルあるいは「スーパーバイザー・マクロ」が異常終了することによりCTASSの手を離れ、実習者がOSのメッセージを直接読み、OSを操作する必要が生じる。このような状況を改善するには、現在使用されているメーカー提供の端末エミュレーターを、独自のものを開発し置き換えることで、かなりの程度可能であろう。

これに対して、ローカルでのプログラム暴走では、MS-DOS自身が破壊される可能性があるので、OSの変更が唯一の解決策である。この点を改善するためと、ユーザー空間が広いことやマルチタスクが可能なることを考慮して、CTASSのローカル・システムをOS/2で稼働させることを進めている。

3) エラー表示のないエラー対策

これは、最終的にはコンパイラーまたはインタープリターの開発・改良につながり、また個々の言語毎に対応しなければならないなど、前述の課題以上に困難である。

現在検討しているのは、プリ・プロセッサ方式でユーザー・プログラム中にテスト・ファンクションを埋め込むことで、値の未定義な変数の使用やプログラム領域への書き込みなどの不当なメモリーのアクセスを検出する方法である。

いずれにせよ、「エラー表示のないエラー」はプログラムの暴走に先行して起こることが多いので、このエラーをいかに少なくするかが大きな課題である。

4) その他の課題

CTASSが市販のシステムに比べて劣る点は、エディターの機能と、通信機能である。多くの人々にこのシステムが受け入れられるためには、この点の改良も不可欠である。しかしこの問題は純粋に技術的な問題であり、本論の趣旨から離れるので、これ以上は議論しない。

V 結論

本論では、情報処理技術者を育てるためには、学生の自主的・自律的訓練が必要であり、そのためには、教育支援環境が重要であることを主張した。今回開発されたCTASSは、操作性の面ではインタープリターに劣らず、実習者に適切な指示を出せること、ローカルおよびホスト間での統合性の面と、異なる言語間での統一性と将来への拡張性、さらにエディターに独自の工夫がされていることで優れている。

VI 謝辞

CTASSの初期の段階からシステムを使用し温かく気長に見守ってくださった藤木顕房氏・宮内文隆氏はじめ立教大学コンピュータ・センターの諸氏および同大学の学生諸君に深く感謝いたします。開発の初期の段階においてFHLの故・折原克己氏にはMSPの内部構造等について親切にお教え頂きました。ここに氏の冥福をお祈りします。

- (1) 塚原周信 「教育用計算機利用支援システム (CTASS) の開発について」
私立大学FACOMユーザー研究会 1986.
10. 28発表資料
- (2) 座間宣夫・塚原周信「Fortranによる情報処理入門」(近代科学社) 1986
- (3) 「教育実習システムの使用法」立教大学コンピュータ・センター発行