学生には良い計算機システムを

湯淺 太一
京都大学

ずいぶん昔のことになるが，大学の計算機システムをリプレースするにあたって，米国製にするのか国産にするのか，大きな問題になったことがある．米国の大学ではUNIXが主流になっていたのに対して，国内の大学ではUNIXという名称すらあまり知られていない時代のことである．

計算機に関する日米の技術力の格差は歴然であったが，円が安く，外貨もまだ乏しかったので，米国製のシステムを国立大学で導入するのはきわめて困難であった．予算面だけでなく，政治的にも困難を伴ったようだ．明らかに劣勢な国内の計算機産業を，国の機関である国立大学が支えるのは当然だという風潮があった．その折りに，米国製の優れたシステムは，学生にとって必ずしも良い効果を与えるとは限らない，という主張を聞いたことがある．良いシステムを日常的に使っていると，それ以上良いものを開発しようという気力を失ってしまう，というのが理由である．国産機を使いこなして，良いソフトウェアを開発していくことが，学生にとっても，さらには国内の計算機産業にとっても望ましい，と考えていたようである．

当時は，国産システムを導入したい人たちの屁理屈としか思えなかったが，今想えば，まったくの空論とは言い切れない面がある．実際，身のまわりの計算環境を整えるためにプログラムを自作し，それによって実力を養った研究者も少なくないのではないだろうか．海外の論文は，今でいえばTeXに相当するソフトを使って，実に美しく仕上がっている．なのに，手元の計算機を使うと，タイプライタで打ち出したような貧弱なものしか出力できない．イタリック体やボールドフェースはもちろん使えないし，下つき文字すら使えない．しかたがないので，TeXまがいのソフトを自作する．プリンタに出力するために，プリンタの制御マニュアルをすみからすみまで読んで，一種のドライバを作りあげる．フォントが足りなければ，自作する．フォントの作成作業を軽減するために，専用のグラフィック・インタフェースまで作ってしまう．といった調子で，プログラムを書いていくものだから，自然とプログラミング能力は養われる．必要にせまられて書くプログラムなので，バグのために迷惑するのは自分自身である．だから，バグを徹底的に退治する習慣もつく．

もちろん，こういった努力をいくら積み重ねても，研究としては評価されない．米国製のシステムを購入すれば，同じようなソフト，もっと性能の良いソフトがすでに備わっている．プログラミング能力を養うためには，絶好の演習であったのは確かだが，研究として評価されるソフトウェアを開発するためには，その時代の最先端のものを超える必要がある．そのためには，性能の良い計算機システムを日常的に使用するのが最も近道である．

プログラミングというものは，他人の書いたものを真似ることから始まる．真似る対象は，ユーザインタフェースであったり，アルゴリズムであったり，コーディングの方法であったり，ドキュメントの書き方であったりする．真似る経験を経ないで，「さあ，これこれのプログラムを書いてみなさい」といわれても，どこから手をつけてよいか検討がつかないであろう．未知の外国語の文法書を渡されて，「さあ，この外国語で原稿を書きなさい」といわれるようなものである．優れたソフトウェアを手本にして，真似てよい部分は真似る．独自性を要求される部分，当面の問題に固有の部分は，手本を参考に自分で考える．そうこうしているうちに，自分なりのプログラミングが徐々に確立していく．この際に，手本とするソフトウェアの完成度が，あとあとまで影響する．すばらしいドキュメントを見て育った者は，自分のソフトウェアにも，できるだけ良いドキュメントを用意しようとするだろう．整然としたコードを見て育った者は，自分のコードを雑然としたままでほっておくことはしたくないだろう．概念的にまとまり，使い勝手の良いユーザインタフェースに慣れていれば，自分のソフトウェアのユーザインタフェースに対しても，相当の時間をかけて考慮するはずである．

PCが急速に普及し，学生たちにとって，最も身近な計算機システムとなっている．しかし残念なことに，PCのOSや応用ソフトが良い手本になるかというと，かなり疑問である．当たりはずれは当然あるだろうが，現在のPCの信頼度はきわめて低い．一般消費者が購入する製品で，これほど信頼性の低いものは他に存在しないだろう．ISDNが普及しはじめたころは，ターミナルア

ダプタの信頼性はきわめて低かった．数日に1度はハングアップしていた．数カ月後に，もう1台購入しに行くと，そのモデルはもう販売されていない．新しいモデルを購入すると，まったくハングアップしなくなっている．初期の問題個所が改善されて，信頼性はきわめて高くなっていた．PCの信頼性は，逆に，年を経るごとに低下しているように思える．かつて，百万円単位で販売されていた時代は，信頼性に関してメーカが相当な力を入れていたにちがいない．しかし，価格が低下し，機能強化に重点が置かれ，モデルチェンジが頻繁に行われたために，信頼性に力を入れる余裕が，メーカにはないのではないかと思われる．リセットボタン（電源スイッチであったり，いくつかのキーの組合せであったりすることもあるが）を押す回数が，年々増えているように思えてならない．消費者もこの現象になれてしまっている．自家用車が数日に1度の割合でエンストすれば，消費者は黙っていないだろう．もっと低価格の家電製品でも同じである．にもかかわらず，PCに限っては，メーカに苦情を言わずに，黙ってリセットボタンを押す．

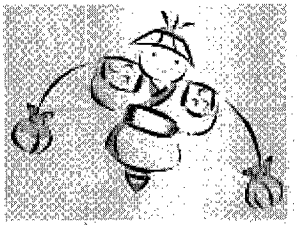
PCは，通常の家電製品などとは本質的に異なっている．ハードウェアがあって，OSがあって，さまざまなドライバが常駐し，市販のソフトやインターネットでダウンロードしてきたソフトを使用する．トラブルが生じた場合に，そのどこに原因があるのかを特定するのが難しい．とはいえ，OSも含めたPCそのものの信頼性が低下しているのは事実であろう．

困ったことに，PCを使っている学生たちは，応用ソフト使用時のトラブルに慣れっこになってしまっている．長い文章を入力していて，不意にPCがハングアップする．ここで怒り狂うのではなく，「あーあ」と言いながらリセットボタンを押して，打ち直し始める．あるいは，使いたいソフトが動かないので，何度か起動をやり直す．そのうちに動き出すようになって，「まあ，こんなもんでしょ」とそれを使い始める．そのような学生が，信頼性の高いソフトウェアを開発できるとは思えない．ひととおりプログラムを書いて，そこそこ動作することを確認して作業を終える．もちろんバグがあちこちに残っているはずだが，平気で次の仕事にとりかかる．

OSも含め，PCのソフトには，マニュアルが完備していないものが多い．高機能なソフトほど，その傾向が強いようだ．分厚いマニュアルがあったとしても，個別の事項を羅列しただけのものが多く，ソフト全体の設計理念のようなものが記述されていることは滅多にない．それでも，いろいろといじくりまわしているうちに，学生たちはそのソフトを使い始めることができる．この能力たるや，たいしたものである．マニュアルがあったとしても，それをじっくり読んでから使い始めることはあまりしない．とにかく使い始めてみて，困ったときにオンラインヘルプでも分からなければ，マニュアルの説明を探し始める．彼らにとって，マニュアルとは，メモ程度の資料に過ぎないのではないだろうか．個別の事項は，ソフトを使いながら必要に応じて学習していけばよいが，本来マニュアルに書いておくべき設計理念や必要な諸概念を理解せずにソフトを使い始めるのは，かえって時間を無駄にする．マニュアルを書くのは，単にユーザのためだけではない．マニュアルを書くことによって，設計理念や諸概念を整理することが可能になる．マニュアルに説明をうまく記述できない場合は，これらが整理されていないことが原因であることが多い．マニュアルを書きながら整理していくことによって，ソフト自体の改良個所に気づくことも少なくない．

最近のPCソフトは，文書やメイルアドレスやURLなどを入力するとき以外は，マウス操作だけで利用できる．ディスプレイ上のボタンをクリックするだけで操作できるのは便利だが，ボタンの配置やマウス操作方法（どのマウスボタンを押すか）といったユーザインタフェースが整理されていないものが多い．概念的に相互の関連が小さい項目が，同じメニューに並んでいることもある．学生に教えられて，どうしてこんなところにこの機能ボタンがあるのかと疑問に思うことも少なくない．しっかりしたマニュアルを作成していないのが，その一因であろう．また，次々と機能追加していった過程で，機能ボタンを設置するメニューに困って，適当に置いてしまった，と想像できることもある．ゲーム感覚で機能ボタンを探す学生は楽しそうであるが，ソフトを開発する際の手本にはしてほしくない．

いろいろとPCの欠点をあげてきたが，もはや教官さえもPCなしでは仕事ができない時代である．電子メイルにPCソフトで作成した文書が添付されていれば，PCを使って読むしかない．この提出文書は，これこれのPCソフトで作成すること，という通達も届く．長期出張には，ノートPCを携帯しないと仕事が溜ってしまう．メディア系の最新ソフトは，PC用にまず開発されることが多い．今後もPCは普及し続けるだろうが，ソフトウェア関係者にとっては，現状のPCは，必要悪といえるかもしれない．いっそのこと，反面教師として学生の教育に役立てるのが良いかもしれない．いずれにしても，学生には，手本にできる良い計算機システムを与えたいものである．軽自動車にしか乗ったことがないエンジニアが，F1を設計できるはずがない．



（平成11年10月5日受付）