

Netpbm について

Netpbm は画像の形式変換、画像処理を行うプログラム群で 300 以上のプログラムから成るものです。PBM 形式はジェフ・ポスカンザー (Jef Poskanzer) 氏が提唱しました。電子メールでの利用を意図した最初の画像形式です。ポスカンザー氏は 1988 年に PbmPlus の名で関連プログラム集を公表しました。しばらくポスカンザー氏の手から離れていたところ、1993 年に Netpbm と名を変えて再度公開されました。当時はオープンソースという言葉は無く、ネットワーク (主にニュースグループと電子メール) を介して技術者が協働するのは斬新でした。2000 年からブライアン・ヘンダーソン (Bryan Henderson) 氏が開発と保守を主導して現在に至っています。

1980 年代から 1990 年代前半は画像通信がまだ普及途上でした。パソコンやワークステーションも画像表示が可能になったものの、各機種乃至 OS の独自形式で行われていました。次々に画像形式が生み出されていくなか、変換ツールが必要になっていました。技術者が自作プログラムを持ち寄る形で Netpbm は成長していきました。Netpbm は画像処理のライブラリと言われますが、独立したプログラムの集合と言う方が正確です。

Netpbm の特徴は要素分解で、デコーダとエンコーダが分かれています。GIF と PNG はともに人気の高い形式で、どちらも Netpbm で扱えますが、直接変換するプログラムはありません。白黒画像の PBM、灰色画像の PGM、カラー画像の PPM、これらを総合的に扱う新形式の PAM のどれかを必ず介在させ、一般に変換は 2 つのプログラムをつなぎ合わせて行います。タテヨコの大きさや明暗、向きの変更などが必要な場合はデコーダとエンコーダの間に処理を挿入します。Netpbm は変換と加工を行うバッチ処理に向いていますが、特にサムネイル作成によく使われました。画像スパムの検出処理の一環として使われているとも聞きます。多くの一般人が意識せずに使っています。

現在は昔と違い、画像は機種依存しない JPEG や PNG などの一般的形式が普及定着しています。広く使われている形式は限られていて、それぞれに専用ライブラリがあり、応用ソフトはこのライブラリを利用して直接画像を読み書きしています。また画像の加工には Gimp などの専用エディタがあり、コマンドラインの Netpbm を使う必要はありません。オープンソースカンファレンス京都 2017 では「未だに更新が行われているのですか？」と驚いている人がいました。Netpbm は枯れている技術との認識があるようです。

筆者は 2004 年からプログラムの改良に協力しています。初めは処理効率化のために凝ったアルゴリズムをいくつか提供しましたが、年とともにセキュリティに影響しそうな不具合の修正作業の割合が増えています。最近では侵犯手口が巧妙化しているので小さな隙も放置できません。2012 年に構築後のテスト `make check` を作成しました。一般的な処理でも結構不具合が見つかるものです。多くの人がソースコードを提供して作り上げられたプログラム群なので重複論理が有り、これらの統合が進行中です。

検証や再構築が効率的に行えるのは信頼性の高い開発用ツールによるところが大きいと思います。アイデアが相互に参照できる環境があるのは良いことです。UNIX の設計思想、基本ツールの発案、GNU プロジェクトでの精錬と普及、`valgrind` などその後加わった用具など、多くの先人の努力と英知の上に今日の開発環境があると常々感じます。

ソフトウェアは利用者が多いほどよいとの考えがあるようです。この考え方には営利主義の影響を感じます。プログラムは用途に応じて最適なものを選ぶべきです。技術者は積極的に試行してそれぞれの道具の長所短所を把握しておくべきです。

日本の Netpbm 利用者の最大の関心事は日本語マニュアルの整備ではないかと思います。翻訳まで手が回らず申し訳ありません。この面で協力を頂けると有難いです。また、Debian とその派生の Ubuntu 等で使用されている Netpbm-Free 版がブライアン・ヘンダーソン氏が管理し、SourceForge.net で頒布している版から 10 年以上遅れていて齟齬が目立つという問題もありましたが、統合に向けて大きく動き始めています。

プログラミング教育について

今回のオープンソースカンファレンス京都では「プログラミング教育」は誰のため?と題する基調講演が行われました。

黎明期のコンピュータ技術者はプログラムを自力で2進数のコードに変換し、入力したそうですが、今日のプログラミングは高級言語のコンパイラ、ライブラリ、インタプリタ等を利用します。これらは他人が製作したものです。優れた開発用ソフトウェアの多くは無料で自由に使えるものです。今日、アルゴリズムの構想から実行形式の完成まで、全て自分で行う人はまずいません。この現実を教師も生徒も理解する必要があります。

プログラミングの先生は、教室の教師だけではありません。ソースコードを参考にする場合、その作者が先生に当たります。オープンソースの思想は、ソースコードを通して教える先生を尊重しようというものです。

「飲水思源」という言葉があります。水を飲むときは、その源を考えよう。井戸を掘ってくれた人がいたことを忘れないようにしましょうという意味です。情報技術は多くの人の協力によって成り立っているものであり、その教育においては「飲水思源」の精神が大切だと思います。

「論語とコンピュータ」について

セミナー講演 論語とコンピュータ についてはこちらの資料をご覧ください:

https://www.ospn.jp/osc2017-nagoya/pdf/OSC2017_Nagoya_netpbm.pdf

多くの方から有意義な意見を聞くことができました。参集された皆様方、並びにオープンソースカンファレンス実行委員会の皆様に厚くお礼申し上げます。

2017年8月24日 漆畑晶