

新 OS、OSASK（おさすく）の開発

“OSASK” The New Light OS

川合 秀実
Hidemi KAWAI

e-mail:kawai@imasy.org

ABSTRACT. Today, major operating systems for PC (ex. MS-Windows, Linux) are heavy and large. But I think that light and small operating systems can work as well, if re-designed carefully. “OSASK” is that sort of operating system, which we are developing now. It is full 32bit preemptive multitask operating system. It has graphical user interface and Japanese/Korean support. It needs only 1MB for disk space, 33MHz for CPU clock-speed and 8MB for main memory. It can boot in under 5 sec. This project is a part of the long project for developing OSASK. In this project, I ported GNU C Compiler and developed the display driver supporting high resolution and high color.

1. 背景

(1) OSASK の背景

今日、PC 用 OS として普及しているのはまずマイクロソフト社の Windows シリーズ、そして次点がオープンソースの Linux の各ディストリビューションである。これらのおかげで PC の使いやすさは格段に向上してきた。

しかし一方で、これらの OS はバージョンアップにつれて、より高速な CPU、より大容量のメモリ、HDD（ハードディスクドライブ）を要求し続けてもきた。私はこれが妥当だとは思わない。OS の作業内容と CPU の処理能力からすれば、33MHz 程度の CPU に 8MB 程度のメモリ、最低インストール容量 1MB くらいのハードウェア要件で、GUI かつ 32bit マルチタスク OS ができると思われる^[1]。

私はこの OS を OSASK（おさすく）と名づけ、オープンソースでの開発を始めて、大きな支持を集めた。OSASK は現在においても未完成であるが、誰でも無償でダウンロードできる^[2]。本プロジェクトは、その OSASK 開発のうちの 2002 年度後期部分に相当する。

(2) OSASK によってもたらされるもの

このような低処理能力かつ小容量しか必要としない OS は、もちろん旧型 PC をリサイクルするために使えるが、私たちはより大きな恩恵を考えている。遅い CPU でもストレスなく使えるということは、つまりそれだけ少ない処理で同等の機能を果たせるということであり、高速な CPU で OSASK を使えば他の OS 利用時よりも消費電力を節約できる。節約した処理能力をアプリケーションに振り分けることもできる。

インストール容量の縮小は起動時間の短縮という効果を生む。これは顕著であり、OSASK はフロッピーディスクからわずか 3 秒で起動する。またソフトウェアに OS をバンドルしたい場合などでは、OS のインストール容量が小さければ非常に有利である。

OSASK はオープンソースであり、ここで利用された技術は誰でも無償で利用できる。これにより単に OSASK のみならず、他の OS や他のソフトウェアが向上することもあるだろう。またオープンソースコミュニティには学生などの若いメンバーが多く、彼らのスキル向上という教育的効果も期待されている。

2. 目的

平成 14 年度末踏ソフトウェア創造事業ユース「新 OS、OSASK（おさすく）の開発」の目的は、OSASK の開発である。その中でも以下の 3 つが具体的な目標である。

1. ネイティブ開発環境の整備（C 言語）

◎ OSASK 上で利用可能な C 言語処理系を構築すること

2. 多色・高解像度対応

◎ ビデオドライバを改良して、高解像度と多色表示に対応すること

3. オープンソースコミュニティのサポート

◎ OSASK のコミュニティをよりいっそう支援して発展させ、各自のスキル向上やコミュニティとしての開発力の強化につなげる

3. 作業内容と成果

(1) ネイティブ開発環境の整備（C 言語）

2002 年秋の時点では OSASK 上で動作するコンパイラは一つもなく、これを克服することが目的である。新規に C 言語を独自に開発するということはせず、オープンソースである gcc-3.2 を OSASK に移植するという方針で開発作業を行なった。OSASK のクロス開発環境として win32 が充実していたこともあり、gcc-3.2 の win32 版である MinGW^[3] をベースに移植を試みた。以降特に明記しないが、以下ではこの gcc-3.2 ベースの MinGW を単に gcc-3.2 と表記する。

移植に際して最初に問題になったのは、gcc-3.2 の実行バイナリサイズである。コンパイラ、アセンブラの合計サイズは 3086KB にもなる。OSASK はこのとき FDD（フロッピーディスクドライブ）しかサポートしていなかったため、このサイズのまま移植しても HDD サポートを完了するまで全く利用できないことになってしまう。この 3086KB は OSASK に備わる圧縮展開オプションを使うことで 1482KB と半減するが、それでも FD の容量である 1440KB を上回っており、やはり何らかの改造が必須であった。

次の問題は、gcc-3.2 が多くのライブラリ関数の利用を前提にしていることである。〈stdio.h〉や〈stdlib.h〉などの ANSI 標準関数群のほか、〈sys/file.h〉や〈sys/stat.h〉などの OS 依存下位関数群なども要求する。OSASK

にはまだ ANSI 関数群すら一部しかそろっておらず、これも移植にあたって大きな問題になった。

この2つを解決するために、まず gcc-3.2 のソースのような「どんな処理系でも柔軟にコンパイルできるソース」であることをやめ、改造版 gcc-3.2 は x86 系の gcc-3.2 でコンパイルできればそれでよいということにした（改造 gcc-3.2 自身も x86 系の gcc-3.2 に含まれるので、当然セルフコンパイルもできる）。このような仕様変更をする以上 gcc-3.2 と名乗るのは不当だと考え、この改造版 gcc-3.2 を GO（じーおー）と称することにした。この方針によってソースの条件コンパイルを減らし、見通しが良くなった。

そしてデバッグチェック用のルーチンや、コンパイル時間測定ルーチンなど、コンパイルそのものに不要と思われる部分を削除した。ついでにコード生成部を改良し、無駄なパディングをやめさせ、必要かつ十分なコードを生成させるようにした。ライブラリについては GO に必要な仕様を満たすだけの簡易的なライブラリを作り、GO に内包させた。これにより、GO の OS 依存部はファイル関係の 4 関数のみしか必要としない、非常に移植性の高いコンパイラに生まれ変わった。

アセンブラについては、NASM^[4] の下位互換のアセンブラを既に作っていたこともあり、これを拡張することで対応した。これによりアセンブラ部のコードサイズを 85% 以上カットすることができた。これは最初から移植性を重視していたので、ドライバは 1 関数のみである。

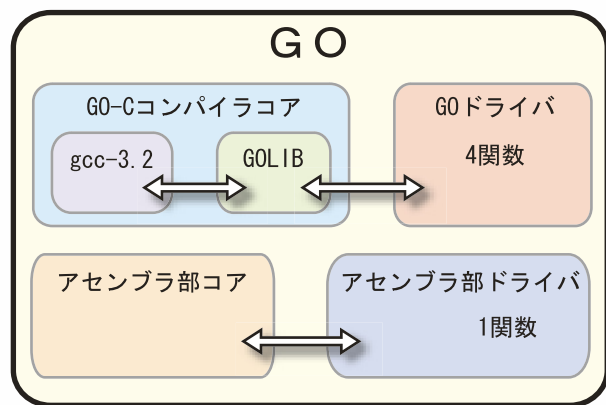


Fig.1 移植性の高い GO の構造

アセンブラ部も含めて、GO の移植性は gcc-3.2 の移植性を大きく上回っており（ただしコンパイラは限定されるが）、x86 の 32bit の OS であれば、OSASK に限らず容易に対応版を作ることができる。本プロジェクト期間内にも OSASK 版の他に win32 版の GO を開発した。

サイズは圧縮展開オプション利用時で 1152KB になり、これは先の 1482KB から 297KB も減少しており、無事に FD に納めて利用できる。機能的にはそのままである。もちろん win32 版も同様に小さくなっている。

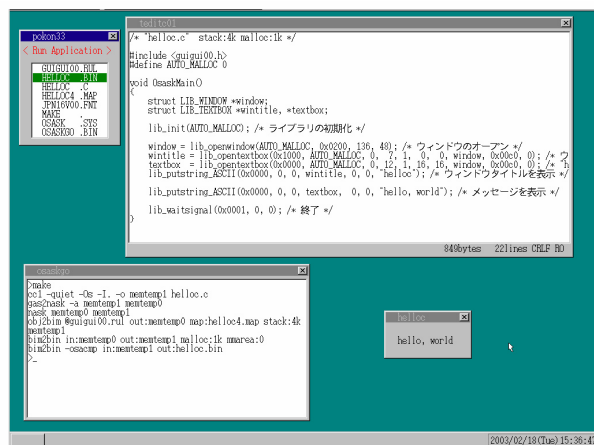


Fig.2 helloc.c をビルドし、実行したところ

(2) 多色・高解像度対応

2002 年秋の時点では OSASK の画面表示能力は 800x600 の 16 色が最大であり、これは主にグラフィックドライバの開発が進んでいないためであった。今回 VESA 用のグラフィックドライバを開発することによって、解像度と色数の両方の制限をなくした。

また一部のグラフィックカードは VESA のサポートが十分でない場合があるので、そのような場合でも多色環境向けに作られたアプリケーションが利用できるよう、自動タイル機能もつけた。しかしタイル機能も万能ではなく、負荷が増えたり色合いによっては文字が読みにくくなる場合もあると思われる。それを回避できるようにするために、タイルングを抑制する機能もつけた。

アプリケーション側は特別な処理をすることなく、このタイル機能が必要に応じて利用できる。アプリケーション開発側の負担を増やすことなしに大きな効果を生んでいる機能といえるであろう。

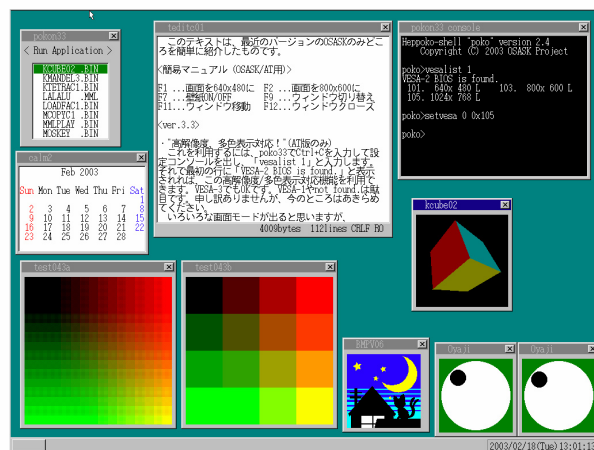


Fig.3 自動タイルングの効果
(解像度 1024x768 の 256 色モード時)
test043a(左)と test043b(右)は同じイメージだが
test043a(左)のみ自動タイルングが有効

(3) オープンソースコミュニティのサポート

オープンソースコミュニティの規模の測定方法としてはいろいろあると思われるのが、ここでは私のホームページ内にある OSASK のページの、トップページ^[2]のアクセス数を指標にして説明したい。これでは不十分であるという批判もあるだろうが、私はコミュニティのメンバーに対して年齢や性別、所属などを特定するような質問を極力避けており、結局アクセス数くらいしか分からないということが背景にある。これは大きく開かれたコミュニティを作るには欠かせないことである。もちろんこのアクセス数も、同一人物のものが複数含まれているのはまず間違いない。コミュニティのメンバーには学校などからアクセスする例もあり、リクエストのあった IP の数だけをもってユニークアクセスとは特定できないため、総アクセス数を扱っている。

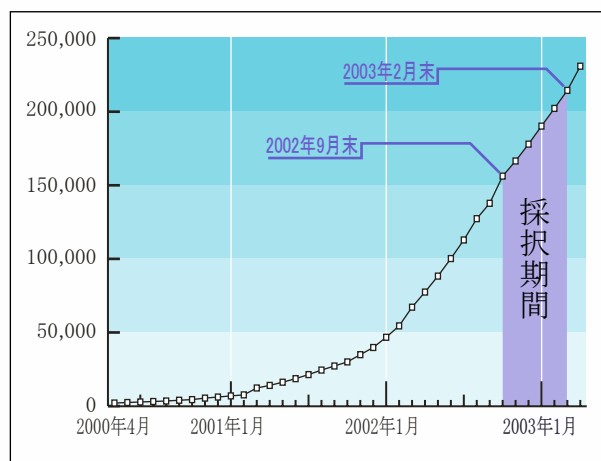


Fig.4 OSASK トップページ延べアクセス数推移

このグラフに示されるとおり、成長は安定して続いている。しかし採択期間とそれ以前を比べると、どうも採択期間はリニアな成長であり、それ以前の指数関数的成長ではないように見える。この原因として私が考えているのは、4月～8月の(つまり採択期間前の)開発ペース鈍化である。今までの傾向を分析すると、アクセス数の伸びは開発進度をすぐに反映するわけではなく、数ヶ月の遅れがあるように見える。したがって期間中のアクセス数は、それ以前の状況を反映していると考えられるわけである。また G0 開発や高解像度対応が完了したのは当然のことながら採択期間の最後であり、これらの影響はむしろ今後あらわれると思われる。

一方、採択期間内であっても ML や掲示板などを見るとコミュニティの充実とは着々と感じられるため、この期間は主に「量の向上ではなく質の向上」があった期間だったのかもしれない。

1月26日には横浜市立大学にて第一回の OSASK オフ会を開催することができた。15人の参加があった。これはそれほど多いとは言えないが、コミュニティの多くが学生であることと、受験・試験時期直前だったことが影響していると思われる。内容は大変に充実したものとなったので、今後回を重ねるごとに参加人数は増えるだろう。

(4) その他

当初の具体的な目標には含まれなかったものの、OSASKの開発という面で小さな前進がさらにいくつかあった。簡単にまとめておく。

簡易プレゼンテーションツールを作った。これは単に画像ファイルをばらばらと表示しているだけであり、そう難しいものではないが、これによって OSASK でプレゼンテーションを行なうことができるようになった。OSASK にはほ

とんど CPU パワーがいらないため、安価でかつ軽量の PC が使える。このような PC は持ち運びに大変便利な上、バッテリーの持ちも良く、さらに OSASK の起動が瞬時なのでプレゼンテーションもすぐに始められる。

また OSASK を HDD でも使えるようにもした。まだ試験的なレベルであり十分ではないが、これにより起動時間は1秒にまで短縮された。当然ながらファイルアクセスは速く、非常に快適である。最近では FDD がオプションになっている PC もあり、それらの機種での利用が容易になるという効果もある。

4. まとめ

2003年3月の時点での OSASK の特徴をまとめておく。
・要求ハードウェア要件の低さにより幅広い機種に対応

33MHz 程度の CPU (さらに遅くても可)
4MB のメモリ
500KB 程度のインストール容量
(全角フォントやアプリを入れなければ、64KB にまで縮小)
PC-9801 版や FM-TOWNS 版もある

- ・32bit マルチタスク、GUI
- ・高解像度、多色画面にも対応
- ・gcc 互換の C 言語も利用可能
- ・起動時間: 1 秒 (HDD)、4 秒 (FDD)
- ・オープンソース、フリーソフト

5. 今後の展望

OSASK はまだインターネットを利用できないとか、日本語入力環境が十分ではないという問題をかかえている。また細部の詰めも十分ではない。今後着実に改善していく。

OSASK のインストール容量の小ささと起動の速さは PC における GUI-OS の常識を覆すものであり、今後この技術を利用した魅力的なソフトウェアが出てくるだろう。私たちも、この技術を応用して、ほぼすべての PC でどんなデバイスからでも OS の起動をも可能にする、「KHBIOS」というソフトウェアの作成を予定している。これによって例えばメモ리카ードに OS を納めて起動することが、BIOS の書き換えなしに可能になる。これは電源投入直後にコンパクトな OS が起動して BIOS の仕事を肩代わりするという仕組みになっている。

6. 参加企業及び機関

本プロジェクトは、OSASK のオープンソースコミュニティの多大な協力によって実施された。またプロジェクトの管理は、株式会社メディアフロント^[5]が行なった。この場を借りて、協力いただいた全ての人に感謝の意を表したい。

7. 参考文献

- [1]<http://www.imasy.org/~kawai/osask/boyaki08.html>
- [2]<http://www.imasy.org/~kawai/osask/>
- [3]<http://www.mingw.org/>
- [4]<http://nasm.sourceforge.net/>
- [5]<http://www.mediafront.co.jp/>