

NetBSD/dreamcast による Visual Memory デバイス活用の実験と考察

2012.5.12 OSC2012 名古屋

おおしま やすし

Nagoya BSD Users' Group

要旨

SEGA Dreamcast は 1998 年にセガ・エンタープライゼスにより発売された家庭用テレビゲーム機である。NetBSD/dreamcast は移植性に富んだ UNIX 系の流れを組む Operation System(以下 OS)である NetBSD を本ゲーム機向けに移植されたものである。

本文書ではゲーム機である Dreamcast と汎用的な機能を備える UNIX 系 OS である NetBSD を用い、ゲーム機ならではの機能として「コントローラ」「メモリカード」そして Dreamcast ならではの「コントローラに内蔵される表示装置」を活用することで手元デバイスの可能性と一般の OS を用いることの利点について実験および考察した。

ハードウェアスペックなどは製品自体の時間経過によって見劣りする部分も非常に多々存在するが、認証情報を取り外し可能/可搬型メディアに保持することでデバイス本体を持ち歩かずとも良いこと、また、本体側には一切情報が残らないこと、慣れ親しんだコントローラで操作できることなどの有用性を確認することができた。

これらの動作環境として NetBSD を使ったことにより数多くの既存ソフトウェアを用いることができ、実質的に 10 日程度の開発期間で動作/検証が可能であった。

なお、本内容は基本的に全て OSC2012 名古屋 NBUG 展示における一発芸、ネタ目的であるため、記載内容を無条件に真に受けたくないよう注意いただきたい。

Dreamcast コントローラと Visual Memory

Dreamcast の標準コントローラには拡張スロットが 2 つ存在し、そこには通常ゲームなどのデータ保存に使用するメモリユニット Visual Memory(以降 VM)を挿入して使用する。なお、このスロットにははぶるぶるユニットやマイクデバイスなどの I/O 装置が接続されることもある。

VM の容量は 128KByte である。また、内部に 8bit CPU が搭載されており、コントローラボタンもある単体のゲーム機としても動作する。VM には 1 つの液晶画面が存在しており¹、48x32pixel のドットマトリクス表示が可能である。

NetBSD/dreamcast におけるデバイスサポート

現在 NetBSD/dreamcast では、VM のメモリ領域をストレージ系ブロックデバイス `mmd(4)`として使用することができる。ただし、`ffs(4)`などの汎用ファイルシステムとしての用途はサポートされていないため、ブロックレコードアクセスに限られる。また、液晶画面は 2 階調ビットマップ表示を行う最低限のインターフェースをもつ `mlcd(4)`デバイスとしてサポートされている。テキスト表示などはサポートされない。

一方、コントローラとしては一般的な `mouse`のみ `mms(4)`としてサポートされており、メモリスロットを持つコントローラ自体を操作するデバイスドライバは現状では用意されていない。

手元デバイスとしての活用

小さいながらも画面があり、コントローラという操作デバイス、可搬型のストレージが揃っているという事実は、操作する個人だけが利用する手元デバイスとして活用できる可能性がある。この検証として近年マイクロブログとして知名度の高い Twitter の表示系をこの構成で利用することを考えた。Twitter で使用するユーザアカウントの認証情報を VM に保存し、タイムラインを VM の液晶画面に出力しコ

ントローラで操作するシステムを構築、実行した。

開発手法

開発箇所の局所化と開発期間短縮を目的とし、動作環境たる OS には汎用性のある NetBSD を使用することとした。また、既に動作実績のあるいくつかのアプリケーションをなるべく利用し、新規開発部分を極力減らす方針とした。

まず Twitter のタイムライン取得にはシェルスクリプトで作成された `shtter`²を使用した。これは NetBSD の標準機能と GNU `sed` と `wget` のみで実行可能な shell script として記述されているものである。これを利用した背景には、UNIX 系 OS 一般で利用可能であること、高機能な言語処理系を用いないため機能的には少なくともメモリ使用量や動作速度の面で軽く、リソースに限りがある Dreamcast においても軽い動作が期待できるという点がある。

今回、`shtter` の標準出力を受け取り、VM の `mlcd(4)`デバイスに文字列として表示する機能をコマンドとして新規開発している。

また、認証に必要なアカウント情報は `shtter.sh` オリジナルでは shell script 内に記録されて使用できるようになっているが、これを VM のメモリである `mmd(4)`デバイスから取得しシェル変数に設定するよう変更している。また、VM のメモリを参照するプログラムも新規開発した。

新規開発部分はデバイスアクセスなどはほぼ全て C 言語による開発である。またいわゆるユーザランドで動作する通常のプログラムとなっており、カーネル開発のような専門的な技巧は用いていない。

ユーザーインターフェースの一部となるコントローラの操作については標準ではドライバが存在しないが、筆者が数年前に `mms(4)`マウスドライバを改造して作成した `mjoyp(4)`ドライバを使用している。これは独自のものではあるが、デバイス特性の違いによりアナログスティックによる移動情報の変更やボタン数等を変更した以外ソースコード上もほ

1 液晶画面を持たない 4x メモリなども存在する。

2 <http://lostman-worlds-end.blogspot.com/search/label/Twitter>

ぼ mms(4)と同等のものである。

処理方式

液晶表示の基本部分は非常に単純であり、入力された文字コードに応じてあらかじめ用意したフォントパターンを液晶画面である mlcd(4)に書き込んでいるだけである。フォントパターンは NetBSD 標準添付の X window system に含まれる K14 フォント等³を利用した。このフォントは文字コードとしていわゆる JIS コード配列を採用している。これは標準の Twitter 出力である UTF-8 とは異なるため、shtter.sh の出力をあえて EUC-JP としている。このため、表示可能な文字は日本語(半角カナ領域含む)および英数字のみとなっており、この範囲に属さない言語/文字については現状では出力できないものになっている。

表示インターフェースの工夫

画面サイズが 48x32pixel と非常に狭いため、仮に 14pixel フォントで埋め尽くしたとしても 6 文字しか同時に表示することはできないことになる。そこで表示は 1 行 3 文字を最大とし、8pixel 単位でのスクロールとした。これは内部的に 1 行(140 文字分)を取めるのに十分な領域のビットマップをメモリ上に用意し、実画面である mlcd(4)への書き込み時に位置をずらすという、いわゆる仮想画面方式を採用した。

また、タイムラインは 20 行分取得するため、コントローラの上下ボタンで移動できるようにもした。その他自動的に行移動するモードの切り替え、行頭に戻るボタン、移動のための画面上のアイコンなどもこの液晶上に配置している。

認証情報の保存と参照

Twitter の認証のためのいわゆる鍵を保存/取得する場所として mmem(4)を使用する。ブロックデバイスであるため自由に使用することもできるが、一応 Dreamcast 標準の FAT 管理に従った簡易アクセスルーチンを作成し使用した。ただしあくまでも簡易型であるため、現状では Dreamcast 本体の VM 管理画面では正しく内容が見えないなど不具合が残っている。また、認証情報の新規作成はブラウザや PIN コードの入力等が必要であるため、いわゆる PC 等と連系して操作する必要がある。

その他

Dreamcast 本体の特性として GD-ROM ドライブだけで動作させたい(いわゆる Live-CD)と考え、システムを構築した。通常 Dreamcast 本体には書き換え型の補助記憶装置(容量が大きく、固定ディスク装置のようなものは持たないため、稼働中に書き換えが必要となる部分(DHCP クライアントや Twitter 取得時の一時ディレクトリ等)は全てメモリファイルシステム mfs(4)上で行うこととした。実際には/tmp を mfs(4)とし、書き換えが必要なものは全てこのディレクトリに集約している。また、VM とコントローラだけで動作することが可能だが、画面上にはデモ用の説明を表示するため紙芝居(画像ファイルを順次表示)するようにしている。この表示プログラムも以前に作成したものを流用している。

検証および開発行程

今回アイデアの具体化、検証プログラムの作成およびデモシステム構築までのおおよそのスケジュールを図に示す。



実際のプログラム開発に費やしたのはほぼ 1 週間程度であり、2012 年 5 月の連休中にほぼ完了し、5/12 までの間に細かいバグ修正や UI 調整およびデモシステムとしてのライブメディア化を行った。

新規に作成したプログラムは僅かに 3 本であり実質的には mlcd(4)上に文字表示を行うプログラム 1 本、450 行程度の小さなものである。

プログラム開発当初は NetBSD/amd64 上の cross-tool chain によるクロスコンパイラ環境を用いたが、デバッグ調整の段階では NFS 環境による NetBSD/dreamcast 上でのセルフ開発スタイルとしている。これは十分にソースが小さいため、セルフ環境で合っても性能上のデメリットがほとんどなかったことによる。

実行結果と考察

NetBSD 本体と開発したプログラムおよび実行スクリプトを組み込んだ CD-R を Dreamcast 本体にセットし電源を入れ、画面のメッセージに従いアカウント情報のある VM をセットすると自動的に TL が取得され、VM の画面に表示およびコントローラで操作と、全て手元で完結するという当初の目的は達成できた。画面の小ささもスクロールすることでそれほど気にならず、むしろ Twitter という短いメッセージには向いているという面もあると言える。

コントローラ的的操作、手元ディスプレイ、持ち運びできるアカウント情報という点は便利であると実感できた。また、この操作感は現在の携帯電話(特にスマートフォン)等+ハードキー操作に通じるものがある。このインターフェース自体にあまり間違いではないことが確認できた。

なお、コントローラを複数繋ぎ、それぞれ別のアカウントを使った別のタイムラインを表示することも可能であるが、今回は未検証である。

一方で、Dreamcast のハードウェア制約によりシステム起動時の CD-R 読み込みに伴うアクセス音や、性能上の問題よりかなりの待ち時間を要する点は大きな問題であるといえ、実用には程遠い結果となった。しかし、20 世紀末に発売され既に 10 年以上前のゲームコンソールにおいてこのような実験ができること、既存の資産を流用することで実質 10 日間程度で動作確認までこぎつけることが可能という点は注目すべきところである。

結論

さんざん色々書いてきたが、無理矢理にでも今風の使い道を Dreamcast ならではのデバイスを使って NetBSD/dreamcast 上でやってみたかっただけである。何の役にも立たないものではないです。

³ pkgsrc から非商用フリーの恵梨沙フォントも使用し、8x8 フォント表示も利用可能としている。