

Shibuya.abc#1

基調講演

新藤 愛大

Shindo Yoshihiro

新藤 愛大

Shindo Yoshihiro

www.be-interactive.org

Shibuya.abc

Shibuya.abc

“ActionScriptByteCode”

Shibuya.abc 誕生の理由

なぜかASバイナリの話を知くと喜ぶ人が多い(変態)

なぜかASバイナリのお話を聞くと喜ぶ人が多い(変態)

なぜかASバイナリのお話をして喜ぶ人が多い(変態)

「**変態は変態同士**仲良くすればいいんじゃないの」

「**変態は変態同士**仲良くすればいいんじゃないの」



Shibuya.abc誕生

3

予定参加者数

≥50

実際の参加希望者数

Thank you.

Shibuya.as は？

概要

開設日	2008年3月10日
管理人	babie さん
カテゴリ	音楽
メンバー数	4人
参加条件	だれでも参加できる
公開範囲	インターネットに公開
トピック作成権限	コミュ参加者なら作成可

プロフィール

アニソンカラオケサークルです。

as は AnimeSong の略です。ActionScript じゃありません。
主に渋谷パセラで活動してます。

Shibuya.**AnimeSong** に！！

僕たちにはもう**Shibuya.abc**しか残されていない

まずはABCについておさらい

AS2ファイル FLAファイル



FLAファイル AS3ファイル



AS2ファイル FLAファイル



FLAファイル AS3ファイル



Flash



asc.jar

AS2ファイル FLAファイル



コンパイル

FLAファイル AS3ファイル



Flash



JAR

asc.jar

AS2ファイル FLAファイル



コンパイル

FLAファイル



AS3ファイル



AS2バイナリ SWFファイル

SWFファイル AS3バイナリ

AS2ファイル FLAファイル



コンパイル

FLAファイル



AS3ファイル



埋め込み



AS2バイナリ SWFファイル



埋め込み



SWFファイル AS3バイナリ

AS2ファイル FLAファイル



FLAファイル



AS3ファイル



コンパイル



埋め込み



AS2バイナリ SWFファイル



埋め込み



SWFファイル AS3バイナリ



AVM1



Flash Player



AVM2

AS2ファイル FLAファイル

FLAファイル AS3ファイル



コンパイル



埋め込み



埋め込み



AS2バイナリ

SWFファイル

SWFファイル

AS3バイナリ

実行



AVM1

Flash Player

AVM2

2007. **AVM2** がオープンソースに (Tamarinプロジェクト)



AVM2

2008. **Flex SDK** がオープンソースに



asc.jar

AS3は**コンパイラ**も**VM**もオープンに



asc.jar



AVM2

ライセンスに従う限り

AS3でやりたい放題できる

Shibuya.abc 的には
これに乗らない手は無い

Shibuya.abc#1

たまりんと戯れる

第一章 Tamarinのビルド

Buildドキュメンテーションに目を通す

http://developer.mozilla.org/en/docs/Tamarin_Build_Documentation

ソースを入手

ソースリポジトリ (Mercurial)

<http://hg.mozilla.org/tamarin-central/>

一気にゲット

<http://hg.mozilla.org/tamarin-central/?archive/tip.tar.gz>

今回はLinuxなので...

g++を入手

```
$ yum install gcc-c++
```

zlibも入手して展開

<http://zlib.net/>

Makefile(makecommon)を修正

```
AVMPLUS_INCS := -I${AVMPLUS_ROOT}/core -I${AVMPLUS_ROOT}/pcre -I  
${AVMPLUS_ROOT}/MMgc -I${AVMPLUS_ROOT}/codegen -I$  
{AVMPLUS_ROOT}/regexp -I${AVMPLUS_ROOT}/extensions -I$  
{AVMPLUS_ROOT}/shell -I${ZLIB_ROOT}
```

makeをする

```
$ cd tamarin-central/platform/unix/  
$ make ZLIB_ROOT=/path/to/zlib
```

しばらく**どきどき**しながらコンソールを見守る

avmplusキタコレ!!

```
$ ls
```

```
DateUnix.cpp MathUtilsUnix.cpp OSDepUnix.o makecommon  
DateUnix.o MathUtilsUnix.o OSDepUnix.s manifest.mk  
DateUnix.s MathUtilsUnix.s avmplus unixcpuid.cpp  
Makefile OSDepUnix.cpp libavmplus.a unixcpuid.o
```

第二章 Hello, world

ascを入手

Adobe Open Source

<http://opensource.adobe.com/svn/opensource/flex/sdk/trunk/lib/asc.jar>

abcをasc.jarのあるディレクトリへコピー

tamarin-central/core/**builtin.abc**

tamarin-central/shell/**toplevel.abc**

hello.asを書いてみる

```
trace('Hello, tamarin!');
```

※asc+Tamarinの場合、AS3のように
必ずしもクラスから始める必要は無い

hello.asをコンパイルしてみる

```
$ java -jar asc.jar -strict -warnings  
-import builtin.abc -import toplevel.abc  
hello.as
```

hello.abcが生成される

```
$ java -jar asc.jar -strict -warnings  
-import builtin.abc -import toplevel.abc  
hello.as
```

hello.abc, 89 bytes written

hello.abcを実行してみる

```
$ ./avmplus hello.abc
```

Hello, tamarinキタコレ!!

```
$ ./avmplus hello.abc
```

Hello, tamarin!

第三章 コマンドライン引数

avmplusには、いくつかの

AS3にはないクラスやメソッド

がある

shell/toplevel.asを覗いてみる

```
package avmplus
{
    public class System
    {
        public native static function exit(status:int):void
        public native static function exec(command:String):int
        ...
    }

    public class File
    {
        ...
    }
    ...
}
```

これだけでもかなり遊べます

add.as (引数二つにとって足し算) を書いてみる

```
import avmplus.System;

var args:Array = System.argv;

if (args.length < 2) {
    trace('Usage:');
    trace(' ./avmplus add.abc n1 n2');
}
else {
    var n1:Number = parseFloat(args[0]);
    var n2:Number = parseFloat(args[1]);
    trace(n1 + n2);
}
```

System.argv でコマンドライン引数

```
import avmplus.System;

var args:Array = System.argv;

if (args.length < 2) {
    trace('Usage:');
    trace(' ./avmplus add.abc n1 n2');
}
else {
    var n1:Number = parseFloat(args[0]);
    var n2:Number = parseFloat(args[1]);
    trace(n1 + n2);
}
```

引数の数 をチェック

```
import avmplus.System;

var args:Array = System.argv;

if (args.length < 2) {
    trace('Usage:');
    trace(' ./avmplus add.abc n1 n2');
}
else {
    var n1:Number = parseFloat(args[0]);
    var n2:Number = parseFloat(args[1]);
    trace(n1 + n2);
}
```

Usage の出力

```
import avmplus.System;

var args:Array = System.argv;

if (args.length < 2) {
    trace('Usage:');
    trace(' ./avmplus add.abc n1 n2');
}
else {
    var n1:Number = parseFloat(args[0]);
    var n2:Number = parseFloat(args[1]);
    trace(n1 + n2);
}
```

足し算の結果 を出力

```
import avmplus.System;

var args:Array = System.argv;

if (args.length < 2) {
    trace('Usage:');
    trace(' ./avmplus add.abc n1 n2');
}
else {
    var n1:Number = parseFloat(args[0]);
    var n2:Number = parseFloat(args[1]);
    trace(n1 + n2);
}
```

add.asをコンパイル

```
$ java -jar asc.jar -strict -warnings  
-import builtin.abc -import toplevel.abc  
add.as
```

add.abc, xx bytes written

add.abcを実行してみる

```
$ ./avmplus add.abc -- 1 2
```

※引数を付ける場合"--"を忘れずに

計算結果キタコレ!!

```
$ ./avmplus add.abc -- 1 2
```

3

少数の計算キタコレ!!

```
$ ./avmplus add.abc -- 1.2 2.3  
3.5
```

第四章 標準入力

コマンドライン引数ときたら
次は標準入力

echo.as (エコーバック) を書いてみる

```
for (;;) {  
    var input:String = readLine();  
    if (input == 'quit') {  
        break;  
    }  
    else {  
        trace('You say: ' + input);  
    }  
}
```

readLine で標準入力から1行取得

```
for (;;) {  
    var input:String = readLine();  
    if (input == 'quit') {  
        break;  
    }  
    else {  
        trace('You say: ' + input);  
    }  
}
```

quit だったら終わる

```
for (;;) {  
    var input:String = readLine();  
    if (input == 'quit') {  
        break;  
    }  
    else {  
        trace('You say: ' + input);  
    }  
}
```

右から左に受け流す

```
for (;;) {  
    var input:String = readLine();  
    if (input == 'quit') {  
        break;  
    }  
    else {  
        trace('You say: ' + input);  
    }  
}
```

echo.asをコンパイル

```
$ java -jar asc.jar -strict -warnings  
-import builtin.abc -import toplevel.abc  
echo.as
```

echo.abc, xx bytes written

echo.abcを実行してみる

```
$ ./avmplus echo.abc
```

echo.abcを実行してみる

```
$ ./avmplus echo.abc
```

```
|
```

hogeと入力

```
$ ./avmplus echo.abc
```

```
hoge
```

エコーバックキタコレ!!

```
$ ./avmplus echo.abc
```

```
hoge
```

```
You say: hoge
```

fugaもキタコレ!!

\$./avmplus echo.abc

hoge

You say: hoge

fuga

You say: fuga

quitで終わる

```
$ ./avmplus echo.abc
```

```
hoge
```

```
You say: hoge
```

```
fuga
```

```
You say: fuga
```

```
quit
```

```
$
```

早くもAS3でコマンドラインアプリを
作れる域まで到達

Tamarin++

第五章 ABC直接実行

一々avmplus呼び出すの**ダサくね**？

```
$ ./avmplus hello.abc
```

直接実行したい

```
$ ./hello.abc
```

それ**binfmt_misc**で出来るよ

※参考

<http://d.hatena.ne.jp/kakurasan/20070714/p1>

http://www.linux.or.jp/JF/JFdocs/kernel-docs-2.6/binfmt_misc.txt.html

binfmt_miscを有効にする

```
$ sudo mount -t binfmt_misc binfmt_misc  
/proc/sys/fs/binfmt_misc
```

binfmt_miscに設定を追加

```
$ sudo echo ':ABC:E::abc::/path/to/avmplus:'  
> /proc/sys/fs/binfmt_misc/register
```

設定名

```
$ sudo echo ':ABC:E::abc::/path/to/avmplus:'  
> /proc/sys/fs/binfmt_misc/register
```

拡張子モード

```
$ sudo echo ':ABC:E::abc::/path/to/avmplus:'  
> /proc/sys/fs/binfmt_misc/register
```

擴張子名

```
$ sudo echo ':ABC:E::abc::/path/to/avmplus:'  
> /proc/sys/fs/binfmt_misc/register
```

実行ファイルへのパス

```
$ sudo echo ':ABC:E::abc::/path/to/avmplus:'  
> /proc/sys/fs/binfmt_misc/register
```

hello.abcに**実行権限**を付加

```
$ chmod a+x hello.abc
```

hello.abcを実行してみる

```
$ ./hello.abc
```

Hello, tamarinキタコレ!!

\$./hello.abc

Hello, tamarin!

add.abcも実行してみる

```
$ ./add.abc -- 1 2
```

計算結果キタコレ!!

```
$ ./add.abc -- 1 2
```

```
3
```

でも "--" ダサくね?

```
$ ./add.abc -- 1 2  
3
```

avmplus-wrapperを作る

```
#!/bin/sh
```

```
ABCNAME=$1
```

```
shift
```

```
/path/to/avmplus "${ABCNAME}" -- $@
```

予め"--"をつけて実行する

```
#!/bin/sh
```

```
ABCNAME=$1
```

```
shift
```

```
/path/to/avmplus "${ABCNAME}" -- $@
```

最初の設定を削除

```
$ sudo echo -1 > /proc/sys/fs/binfmt_misc/ABC
```

新しい設定を追加

```
$ sudo echo ':ABC:E::abc::/p/to/avmplus-wrapper:'  
> /proc/sys/fs/binfmt_misc/register
```

add.abcを実行してみる

```
$ ./add.abc 1 2
```

計算結果キタコレ!!

```
$ ./add.abc 1 2
```

3

binfmt_misc++

ActionScript

ActionScript



ActionScript



ActionScript



Client side

ActionScript



Client side



Server side

第六章 *Server Side AS*

CGIとして動かせば簡単

httpd.confに以下の設定を追加

Options +ExecCGI

AddHandler cgi-script .abc

hello.asを修正して**hello-www.as**を作る

```
trace('Content-type: text/html');  
trace("");  
trace('Hello, tamarin!');
```

hello-www.asをコンパイル

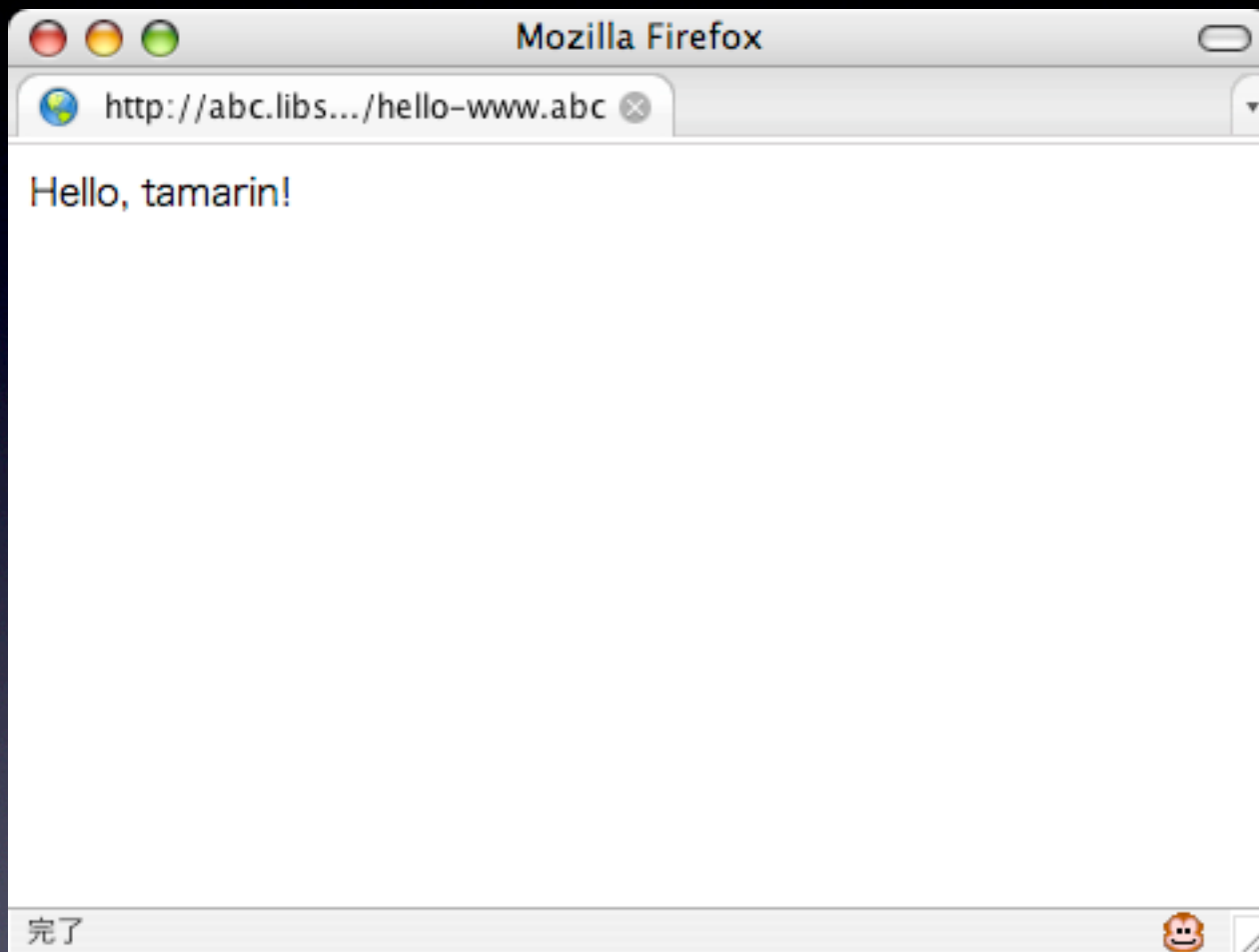
```
$ java -jar asc.jar -strict -warnings  
-import builtin.abc -import toplevel.abc  
hello-www.as
```

hello-www.abc, xx bytes written

hello-www.abcに**実行権限**を付加

```
$ chmod a+x hello-www.abc
```

<http://abc.libspark.org/hello-www.abc>
にアクセス



Server Side ActionScript

キタコレ!!

早速SSASで遊んでみよう

第七章 GET引数

GET引数は

環境変数**QUERY_STRING**に入る
(CGIの仕様)

が、しかし！

AVM2には環境変数を取るためのメソッドが**無い**

早くもSSAS終了のお知らせ

ご清聴ありがとうございました

~~ご清聴ありがとうございました~~

Tamarinはオープンソース

無いなら作ればいい！

shell/toplevel.as に System.getenv 追加

```
package avmplus
{
    public class System
    {
        ...
        public native static function readLine():String
        public native static function getEnv(key:String):String;
    }
    ...
}
```

shell/SystemClass.cpp に実装追加

```
Stringp SystemClass::getEnv(Stringp key)
{
    Toplevel* toplevel = this->toplevel();
    AvmCore* core = this->core();

    if (!key) {
        toplevel->throwArgumentError(kNullArgumentError, "key");
    }
    UTF8String* keyUTF8 = key->toUTF8String();
    return core->newString(getenv(keyUTF8->c_str()));
}
```

※周りに書いてある他のメソッドを参考にする

引数keyをUTF8Stringに変換

```
Stringp SystemClass::getEnv(Stringp key)
{
    Toplevel* toplevel = this->toplevel();
    AvmCore* core = this->core();

    if (!key) {
        toplevel->throwArgumentError(kNullArgumentError, "key");
    }
    UTF8String* keyUTF8 = key->toUTF8String();
    return core->newString(getenv(keyUTF8->c_str()));
}
```

C言語の **getenv** 関数を呼び出す

```
Stringp SystemClass::getEnv(Stringp key)
{
    Toplevel* toplevel = this->toplevel();
    AvmCore* core = this->core();

    if (!key) {
        toplevel->throwArgumentError(kNullArgumentError, "key");
    }
    UTF8String* keyUTF8 = key->toUTF8String();
    return core->newString(getenv(keyUTF8->c_str()));
}
```

結果を **Stringp** に変換して返す

```
Stringp SystemClass::getEnv(Stringp key)
{
    Toplevel* toplevel = this->toplevel();
    AvmCore* core = this->core();

    if (!key) {
        toplevel->throwArgumentError(kNullArgumentError, "key");
    }
    UTF8String* keyUTF8 = key->toUTF8String();
    return core->newString(getenv(keyUTF8->c_str()));
}
```

ASとのマッピングを追加

```
BEGIN_NATIVE_MAP(SystemClass)
```

```
...
```

```
    NATIVE_METHOD(avmplus_System_getEnv, SystemClass::getEnv)
```

```
END_NATIVE_MAP()
```

builtin.abc, builtin.h, builtin.cppを再生成

```
$ java -ea -DAS3 -DAVMPLUS -classpath ../../abc/asc.jar  
macromedia.asc.embedding.ScriptCompiler -abcfuture -  
builtin -import ../../abc/builtin.abc -out toplevel  
toplevel.as Domain.as StringBuilder.as ByteArray.as ../  
extensions/Sampler.as ../extensions/Trace.as ../  
extensions/Dictionary.as Endian.as Java.as
```

avmplusをリビルド

\$ make clean

\$ make

query-string.as (GET引数を出力) を書いてみる

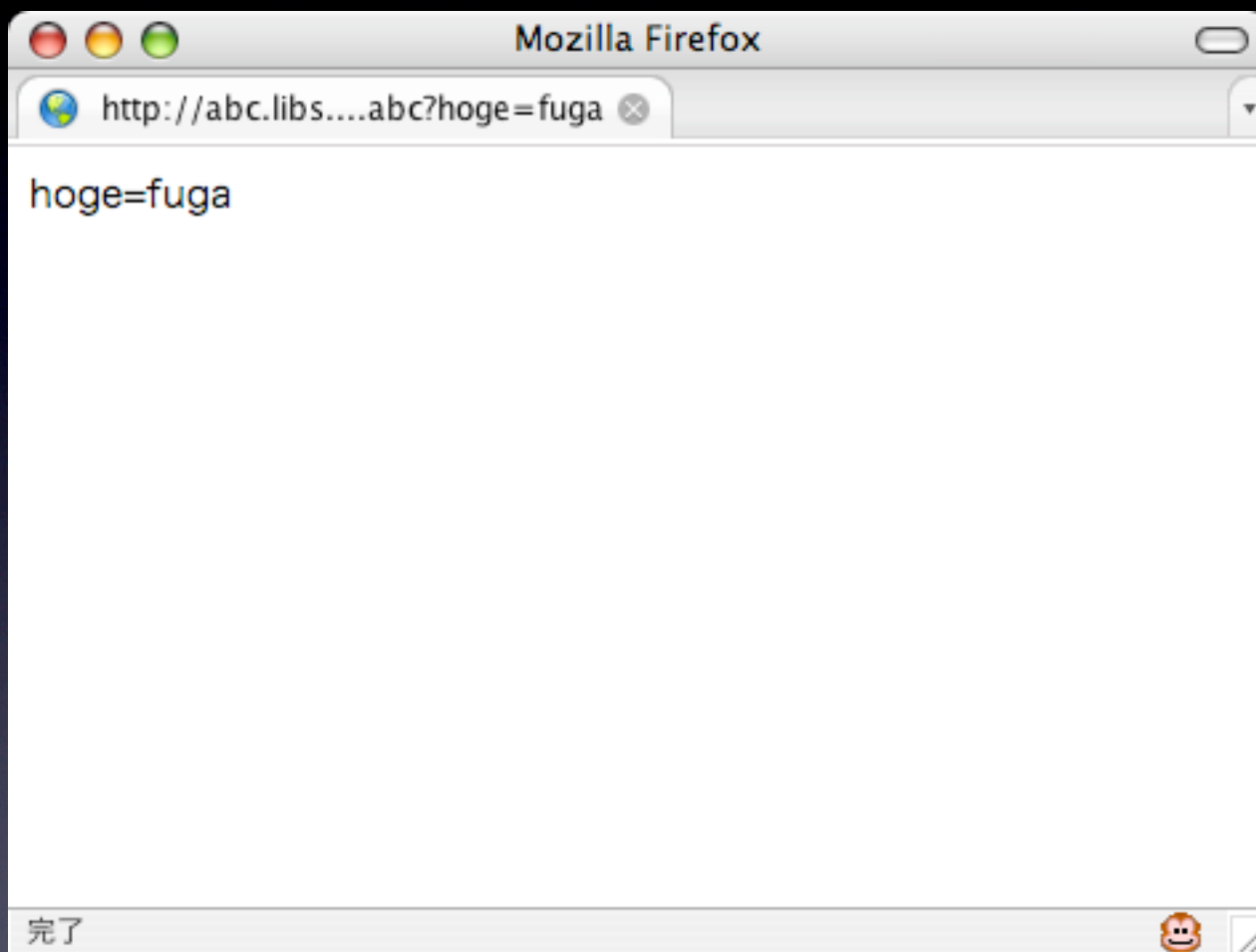
```
import avmplus.System;
```

```
trace('Content-type: text/html');
```

```
trace("");
```

```
trace(System.getEnv('QUERY_STRING'));
```

<http://abc.libspark.org/query-string.abc?hoge=fuga>
にアクセス



GET引数キタコレ!!

第八章 POST引数

POST引数は

そのサイズが環境変数**CONTENT_LENGTH**に入り、
その内容が**標準入力**に入る

(CGIの仕様)

post.as (POST引数を出力) を書いてみる

```
import avmplus.System;
import flash.utils.ByteArray;

var postSize:int = int(System.getenv('CONTENT_LENGTH'));

trace('<ul><li>size: ' + postSize + '</li>');

if (postSize > 0) {
    trace('<li>');
    var post:ByteArray = ByteArray.readStdin(postSize);
    post.writeStdout();
    trace('</li>');
}
```

(一部抜粋)

CONTENT_LENGTH を取得

```
import avmplus.System;
import flash.utils.ByteArray;

var postSize:int = int(System.getenv('CONTENT_LENGTH'));

trace('<ul><li>size: ' + postSize + '</li>');

if (postSize > 0) {
    trace('<li>');
    var post:ByteArray = ByteArray.readStdin(postSize);
    post.writeStdout();
    trace('</li>');
}
```

(一部抜粋)

サイズをチェック

```
import avmplus.System;
import flash.utils.ByteArray;

var postSize:int = int(System.getenv('CONTENT_LENGTH'));

trace('<ul><li>size: ' + postSize + '</li>');

if (postSize > 0) {
    trace('<li>');
    var post:ByteArray = ByteArray.readStdin(postSize);
    post.writeStdout();
    trace('</li>');
}
```

(一部抜粋)

そのサイズ分だけ **標準入力から取得**

```
import avmplus.System;
import flash.utils.ByteArray;

var postSize:int = int(System.getenv('CONTENT_LENGTH'));

trace('<ul><li>size: ' + postSize + '</li>');

if (postSize > 0) {
    trace('<li>');
    var post:ByteArray = ByteArray.readStdin(postSize);
    post.writeStdout();
    trace('</li>');
}
```

(一部抜粋)

取得したデータを **標準出力へ書き出す**

```
import avmplus.System;
import flash.utils.ByteArray;

var postSize:int = int(System.getenv('CONTENT_LENGTH'));

trace('<ul><li>size: ' + postSize + '</li>');

if (postSize > 0) {
    trace('<li>');
    var post:ByteArray = ByteArray.readStdin(postSize);
    post.writeStdout();
    trace('</li>');
}
```

(一部抜粋)

shell/ByteArrayGlue.cpp readStdin実装

```
ByteArrayObject * ByteArrayClass::readStdin(uint32 length)
{
    Toplevel* toplevel = this->toplevel();

    unsigned char *c = new unsigned char[length + 1];

    Atom args[1] = {nullObjectAtom};
    ByteArrayObject *b = (ByteArrayObject*)AvmCore::atomToScriptObject(construct(0, args));
    b->setLength(0);

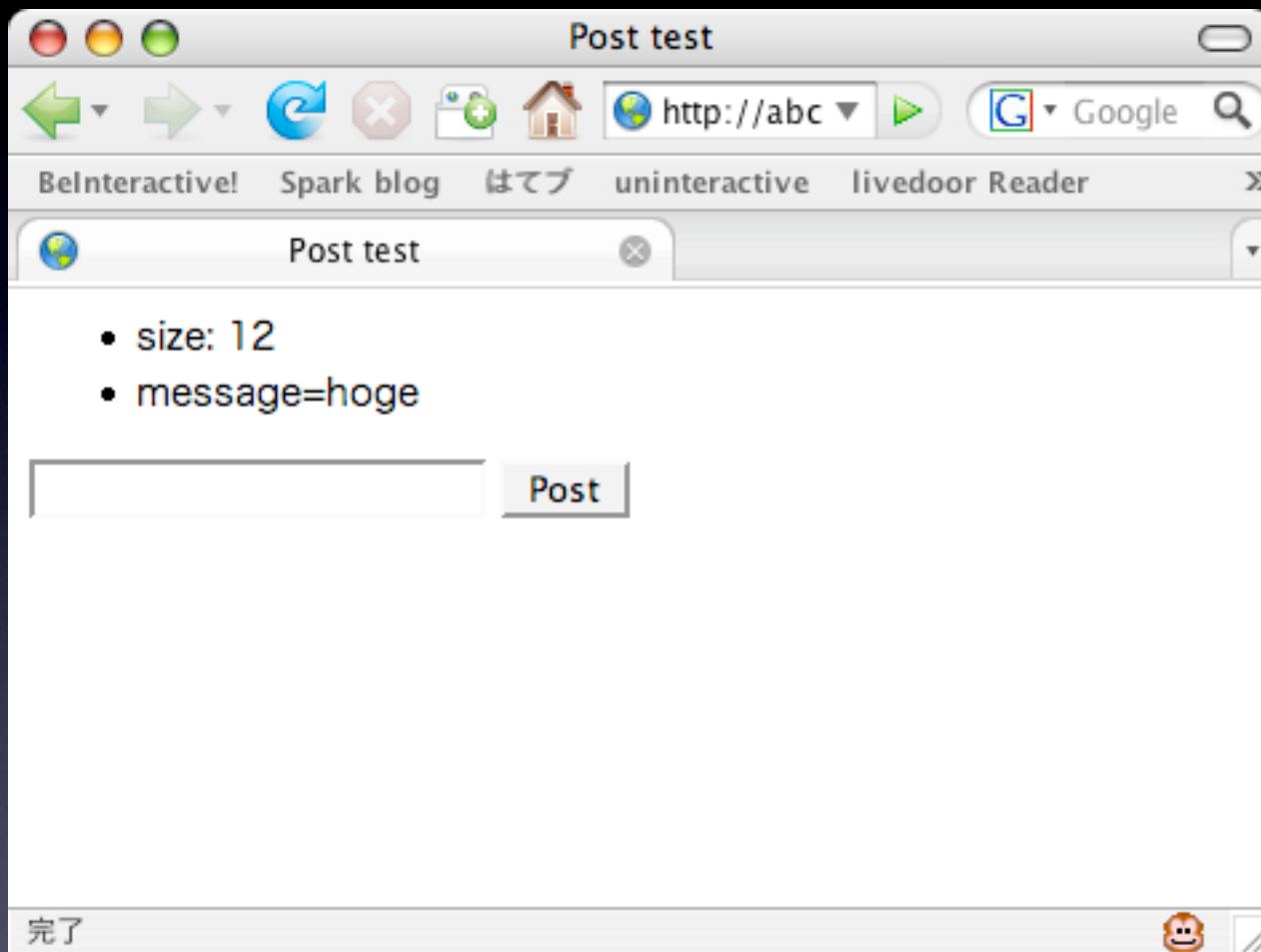
    while (length > 0) {
        int actual = fread(c, 1, length, stdin);
        if (actual > 0) {
            b->fill(c, actual);
            length -= actual;
        }
        else {
            break;
        }
    }
    b->seek(0);

    delete [] c;
    return b;
}
```

shell/ByteArrayGlue.cpp writeStdout実装

```
void ByteArrayObject::writeStdout()
{
    fwrite(&(this->GetByteArray())[0], this->get_length(), 1, stdout);
}
```

<http://abc.libspark.org/post.abc>
にアクセス



POST引数キタコレ!!

第九章 ゲストブック

Webアプリの王道**ゲストブック**を作る

前に、簡易的なWebフレームワークを
作ったのでそれを使って実装

guestbook.as を書いてみる

```
class GuestbookGateway extends AbstractWebGateway
{
    protected override function execute(request:IWebRequest):IWebResponse
    {
        if (request.requestMethod == 'POST') {
            return post(request);
        }

        return view();
    }
}
```

(一部抜粋)

AbstractWebGateway を継承

```
class GuestbookGateway extends AbstractWebGateway
{
    protected override function execute(request:IWebRequest):IWebResponse
    {
        if (request.requestMethod == 'POST') {
            return post(request);
        }

        return view();
    }
}
```

(一部抜粋)

execute メソッドをオーバーライド

```
class GuestbookGateway extends AbstractWebGateway
{
    protected override function execute(request:IWebRequest):IWebResponse
    {
        if (request.requestMethod == 'POST') {
            return post(request);
        }

        return view();
    }
}
```

(一部抜粋)

IWebRequest が渡ってくるので

```
class GuestbookGateway extends AbstractWebGateway
{
    protected override function execute(request:IWebRequest):IWebResponse
    {
        if (request.requestMethod == 'POST') {
            return post(request);
        }

        return view();
    }
}
```

(一部抜粋)

IWebResponse を返す

```
class GuestbookGateway extends AbstractWebGateway
{
    protected override function execute(request:IWebRequest):IWebResponse
    {
        if (request.requestMethod == 'POST') {
            return post(request);
        }

        return view();
    }
}
```

(一部抜粋)

リクエストメソッドが **POST** であれば

```
class GuestbookGateway extends AbstractWebGateway
{
    protected override function execute(request:IWebRequest):IWebResponse
    {
        if (request.requestMethod == 'POST') {
            return post(request);
        }

        return view();
    }
}
```

(一部抜粋)

post メソッド呼び出し

```
class GuestbookGateway extends AbstractWebGateway
{
    protected override function execute(request:IWebRequest):IWebResponse
    {
        if (request.requestMethod == 'POST') {
            return post(request);
        }

        return view();
    }
}
```

(一部抜粋)

view メソッドを呼び出してレスポンスを返す

```
class GuestbookGateway extends AbstractWebGateway
{
    protected override function execute(request:IWebRequest):IWebResponse
    {
        if (request.requestMethod == 'POST') {
            return post(request);
        }

        return view();
    }
}
```

(一部抜粋)

post メソッド

```
private function post(request:IWebRequest):IWebResponse
{
    var args:Object = request.createPostArgs();

    if ('name' in args && 'message' in args) {
        var name:String = args['name'];
        var message:String = args['message'];
        var logXML:XML = XML(File.read(LOGFILE));
        logXML.appendChild(
            <message>
                <name>{name}</name><value>{message}</value>
            </message>
        );
        File.write(LOGFILE, logXML.toXMLString());
    }

    return view();
}
```

(一部抜粋)

POST引数 を取得

```
private function post(request:IWebRequest):IWebResponse
{
    var args:Object = request.createPostArgs();

    if ('name' in args && 'message' in args) {
        var name:String = args['name'];
        var message:String = args['message'];
        var logXML:XML = XML(File.read(LOGFILE));
        logXML.appendChild(
            <message>
                <name>{name}</name><value>{message}</value>
            </message>
        );
        File.write(LOGFILE, logXML.toXMLString());
    }

    return view();
}
```

(一部抜粋)

名前とメッセージが存在しているかチェック

```
private function post(request:IWebRequest):IWebResponse
{
    var args:Object = request.createPostArgs();

    if ('name' in args && 'message' in args) {
        var name:String = args['name'];
        var message:String = args['message'];
        var logXML:XML = XML(File.read(LOGFILE));
        logXML.appendChild(
            <message>
                <name>{name}</name><value>{message}</value>
            </message>
        );
        File.write(LOGFILE, logXML.toXMLString());
    }

    return view();
}
```

(一部抜粋)

ログファイル をXMLとして読み込み

```
private function post(request:IWebRequest):IWebResponse
{
    var args:Object = request.createPostArgs();

    if ('name' in args && 'message' in args) {
        var name:String = args['name'];
        var message:String = args['message'];
        var logXML:XML = XML(File.read(LOGFILE));
        logXML.appendChild(
            <message>
                <name>{name}</name><value>{message}</value>
            </message>
        );
        File.write(LOGFILE, logXML.toXMLString());
    }

    return view();
}
```

(一部抜粋)

XML を追加

```
private function post(request:IWebRequest):IWebResponse
{
    var args:Object = request.createPostArgs();

    if ('name' in args && 'message' in args) {
        var name:String = args['name'];
        var message:String = args['message'];
        var logXML:XML = XML(File.read(LOGFILE));
        logXML.appendChild(
            <message>
                <name>{name}</name><value>{message}</value>
            </message>
        );
        File.write(LOGFILE, logXML.toXMLString());
    }

    return view();
}
```

(一部抜粋)

ログファイル に書き戻す

```
private function post(request:IWebRequest):IWebResponse
{
    var args:Object = request.createPostArgs();

    if ('name' in args && 'message' in args) {
        var name:String = args['name'];
        var message:String = args['message'];
        var logXML:XML = XML(File.read(LOGFILE));
        logXML.appendChild(
            <message>
                <name>{name}</name><value>{message}</value>
            </message>
        );
        File.write(LOGFILE, logXML.toXMLString());
    }

    return view();
}
```

(一部抜粋)

view メソッド

```
private function view():IWebResponse
{
    var html:XML =
        <html>
            <head>
                <title>Guestbook</title>
            </head>
            <body/>
        </html>;

    html.body.appendChild(viewLog());
    html.body.appendChild(viewForm());

    return new HTMLResponse(html);
}
```

(一部抜粋)

E4X でhtmlを作る

```
private function view():IWebResponse
{
    var html:XML =
        <html>
            <head>
                <title>Guestbook</title>
            </head>
            <body/>
        </html>;

    html.body.appendChild(viewLog());
    html.body.appendChild(viewForm());

    return new HTMLResponse(html);
}
```

(一部抜粋)

body にログの内容を追加

```
private function view():IWebResponse
{
    var html:XML =
        <html>
            <head>
                <title>Guestbook</title>
            </head>
            <body/>
        </html>;

    html.body.appendChild(viewLog());
    html.body.appendChild(viewForm());

    return new HTMLResponse(html);
}
```

(一部抜粋)

body にフォームを追加

```
private function view():IWebResponse
{
    var html:XML =
        <html>
            <head>
                <title>Guestbook</title>
            </head>
            <body/>
        </html>;

    html.body.appendChild(viewLog());
    html.body.appendChild(viewForm());

    return new HTMLResponse(html);
}
```

(一部抜粋)

HTMLResponse を返す

```
private function view():IWebResponse
{
    var html:XML =
        <html>
            <head>
                <title>Guestbook</title>
            </head>
            <body/>
        </html>;

    html.body.appendChild(viewLog());
    html.body.appendChild(viewForm());

    return new HTMLResponse(html);
}
```

(一部抜粋)

viewLog メソッド

```
private function viewLog():XML
{
    var logXML:XML = XML(File.read(LOGFILE));
    var html:XML = <ul/>;

    for each (var message:XML in logXML.message) {
        html.appendChild(
            <li>{message.name.toString(): {message.value.toString()}</li>
        );
    }

    return html;
}
```

(一部抜粋)

ログファイル をXMLとして読み込む

```
private function viewLog():XML
{
    var logXML:XML = XML(File.read(LOGFILE));
    var html:XML = <ul/>;

    for each (var message:XML in logXML.message) {
        html.appendChild(
            <li>{message.name.toString()}: {message.value.toString()}</li>
        );
    }

    return html;
}
```

(一部抜粋)

UL 要素を作って

```
private function viewLog():XML
{
    var logXML:XML = XML(File.read(LOGFILE));
    var html:XML = <ul/>;

    for each (var message:XML in logXML.message) {
        html.appendChild(
            <li>{message.name.toString()}: {message.value.toString()}</li>
        );
    }

    return html;
}
```

(一部抜粋)

LI 要素を追加する

```
private function viewLog():XML
{
    var logXML:XML = XML(File.read(LOGFILE));
    var html:XML = <ul/>;

    for each (var message:XML in logXML.message) {
        html.appendChild(
            <li>{message.name.toString()}: {message.value.toString()}</li>
        );
    }

    return html;
}
```

(一部抜粋)

viewForm メソッド

```
private function viewForm():XML
{
    return <form action="?" method="POST">
        <input type="text" size="8" name="name"/>
        <input type="text" size="24" name="message"/>
        <input type="submit" value=" Post "/>
    </form>;
}
```

(一部抜粋)

enter メソッドを呼び出し

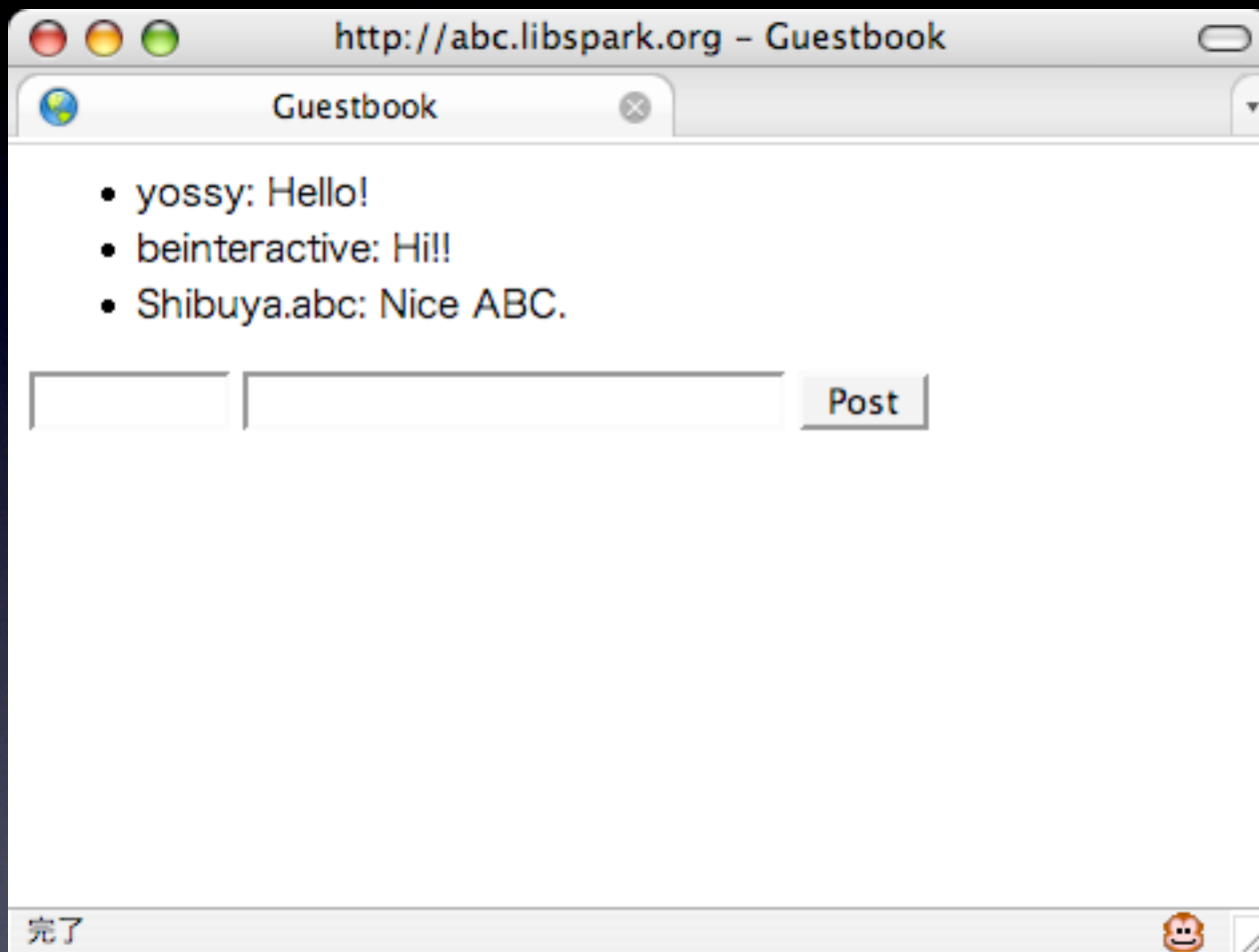
```
new GuestbookGateway().enter();
```

(一部抜粋)

コンパイルする際には**AS3フラグ**をつける

```
$ java -jar asc.jar -strict -warnings  
-import builtin.abc -import toplevel.abc  
-AS3 guestbook.as
```

<http://abc.libspark.org/guestbook.abc>
にアクセス



Nice Guestbook.

ちなみに、E4Xで扱っておくと
勝手に**サニタイズ**されるという嬉しいオマケ付き

最終章 Flash と連携

サーバーサイドはFlashと連携してこそ真価を発揮する

どうせならAMFで通信したい

そんなあなたにBlazeDS

AMFサーバーのJavaによる実装

しかもオープンソース

<http://opensource.adobe.com/wiki/display/blazeds/>

がっつり移植したのでそれを使う

amf.as (計算サービス) を書いてみる

```
import org.libspark.web.AMFGateway;

class CalculatorService
{
    public function add(a:int, b:int):int { return a + b; }
    public function sub(a:int, b:int):int { return a - b; }
}

var gateway:AMFGateway = new AMFGateway();

gateway.addRemotingService('CalculatorService', CalculatorService);

gateway.enter();
```

(一部抜粋)

サービスクラス を定義(普通にクラス定義)

```
import org.libspark.web.AMFGateway;
```

```
class CalculatorService
```

```
{
```

```
    public function add(a:int, b:int):int { return a + b; }
```

```
    public function sub(a:int, b:int):int { return a - b; }
```

```
}
```

```
var gateway:AMFGateway = new AMFGateway();
```

```
gateway.addRemotingService('CalculatorService', CalculatorService);
```

```
gateway.enter();
```

(一部抜粋)

AMFGateway を作成

```
import org.libspark.web.AMFGateway;

class CalculatorService
{
    public function add(a:int, b:int):int { return a + b; }
    public function sub(a:int, b:int):int { return a - b; }
}

var gateway:AMFGateway = new AMFGateway();

gateway.addRemotingService('CalculatorService', CalculatorService);

gateway.enter();
```

(一部抜粋)

サービスクラス を登録

```
import org.libspark.web.AMFGateway;

class CalculatorService
{
    public function add(a:int, b:int):int { return a + b; }
    public function sub(a:int, b:int):int { return a - b; }
}

var gateway:AMFGateway = new AMFGateway();

gateway.addRemotingService('CalculatorService', CalculatorService);

gateway.enter();
```

(一部抜粋)

Flash側 を書いてみる

```
var nc:NetConnection = new NetConnection();  
  
nc.connect('http://abc.libspark.org/amf.abc');  
  
var r:Function = function(res:Object):void  
{  
    trace('result<' + res + '>');  
};  
  
nc.call('CalculatorService.add', new Responder(r, null), 3, 4);
```

NetConnection を作成

```
var nc:NetConnection = new NetConnection();
```

```
nc.connect('http://abc.libspark.org/amf.abc');
```

```
var r:Function = function(res:Object):void
```

```
{
```

```
    trace('result<' + res + '>');
```

```
};
```

```
nc.call('CalculatorService.add', new Responder(r, null), 3, 4);
```

AMFGateway に接続

```
var nc:NetConnection = new NetConnection();
```

```
nc.connect('http://abc.libspark.org/amf.abc');
```

```
var r:Function = function(res:Object):void
```

```
{
```

```
    trace('result<' + res + '>');
```

```
};
```

```
nc.call('CalculatorService.add', new Responder(r, null), 3, 4);
```

リスナー関数を作る

```
var nc:NetConnection = new NetConnection();
```

```
nc.connect('http://abc.libspark.org/amf.abc');
```

```
var r:Function = function(res:Object):void
```

```
{
```

```
    trace('result<' + res + '>');
```

```
};
```

```
nc.call('CalculatorService.add', new Responder(r, null), 3, 4);
```

CalculatorService.add を引数 3, 4 で呼び出す

```
var nc:NetConnection = new NetConnection();
```

```
nc.connect('http://abc.libspark.org/amf.abc');
```

```
var r:Function = function(res:Object):void
```

```
{
```

```
    trace('result<' + res + '>');
```

```
};
```

```
nc.call('CalculatorService.add', new Responder(r, null), 3, 4);
```

Flash を実行してみる

出力:

AMF通信 キタコレ!!

出力:

result<7>

もちろん**完全型互換**

QueryService を書いてみる

```
class QueryService
{
    public function getPeopleList(query:Object):Array
    {
        var result:Array = [];
        for (var i:uint = 0; i < query.limit; ++i) {
            result.push(createPeople(query));
        }
        return result;
    }
    private function createPeople(query:Object):Object
    {
        return {
            name: createName(query),
            age: createAge(query)
        };
    }
}
```

...

Flash側 を書いてみる

```
var r:Function = function(res:Object):void
{
    for each (var p:Object in res) {
        trace('<' + p.name + '><' + p.age + '>');
    }
};
```

```
var query:Object = {limit: 5, nameLen: 3, minAge: 30, maxAge: 40};

nc.call('QueryService.getPeopleList', new Responder(r, s), query);
```

Flash を実行してみる

出力:

AMF通信 キタコレ!!

出力:

<OJK><34>

<LEV><38>

<RRY><33>

<BIH><35>

<EYJ><39>

非常に**楽**にやりとり可能

SSAS++

クライアントもサーバーも
両方ASで書く日も近いかもしれません

本日の資料及びソースコードは
全て公開しますので興味ある人はどうぞ

www.be-interactive.org

おまけ

<http://abc.libspark.org/swfassist.abc>

swfassistも動いた

まとめ

Tamarin と Flex SDK で遊べ!!

ご清聴ありがとうございました