

校内用 RSS 電子掲示板システムの構築

技術専門職員 加島 倫

本システムは、汎用 Web 掲示板ソフトウェアを利用した、LAN 接続の複数掲示器へ情報を自動表示させる校内用電子掲示システムである。

キーワード：掲示板, 遠隔, 携帯電話, Linux, 出欠席

概要

本システムは、汎用 Web 掲示板ソフトウェアを利用した、LAN 接続の複数掲示器へ情報を自動表示させる校内用電子掲示システムである。教職員は机上の PC、携帯電話などから掲示システムへアクセスし、遠隔から掲示器へ情報を表示することが可能である。

CD-ROM ブートで稼働する Linux ベースの掲示器クライアントを特定の Web ページ表示をするのみに特化し、Microsoft SharePoint サービスの RSS フィード情報を IIS Web サーバーと .net ASP アプリケーションにより制御し、SharePoint サービス上の掲示板情報を掲示器に表示させる。

掲示器は安価で設置することができるため、低コストでの運営が可能である。

掲示器は、朝 7 時に、その日の掲示器の稼働が必要かの有無をサーバーに問い合わせる。

URL アドレス `http://myserver/notice/pwr.aspx` にアクセスすると、稼働日の場合 "PWR=1"、休日では "PWR=0" の文字列が返答される。pwr.aspx は、.net ASP のスクリプトで、WSS3.0 のデーターが蓄積されている SQLServer 2005 上の記事テーブルから当日の電源情報を抽出するようになっている。一連のデーターフローを図 2 に示す。

1. システム構成

図 1 に本校における機器構成を示す。サーバーパソコン 1 台に Windows Server 2003 R2、Windows Sharepoint Services 3.0(WSS3.0)、SQLServer 2005、IIS(Web Server)を稼働させ、掲示器は小型 PC と大画面ディスプレイを 1 組とし、3 組設置した(現在は 2 台のみ稼働)

2. 動作フロー

教職員は個人端末から WSS3.0 の汎用掲示板にアクセスし必要な情報を書き込む。掲示器は 10 分おきにサーバーの URL アドレスを表示・更新する。サーバーは掲示器から特定の URL アドレスでアクセスを受けると、WSS3.0 の掲示板 RSS フィードを読み出し、html へ展開後、掲示器へ Web ページとして掲示情報を返信する。

図 1. システム構成図

SYSTEM EQUIPMENTS OVERVIEW

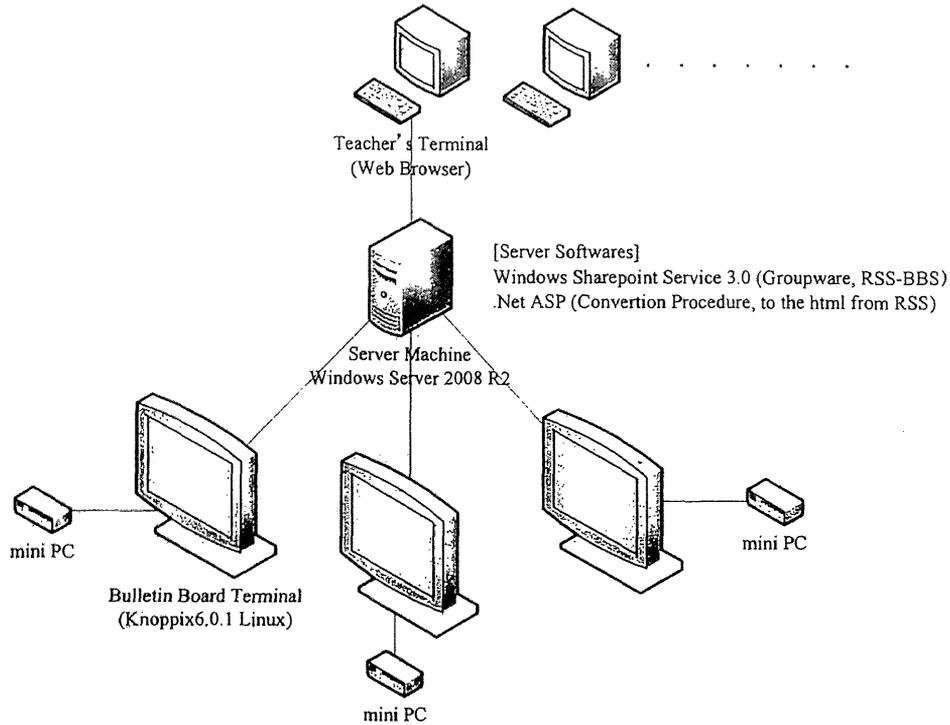
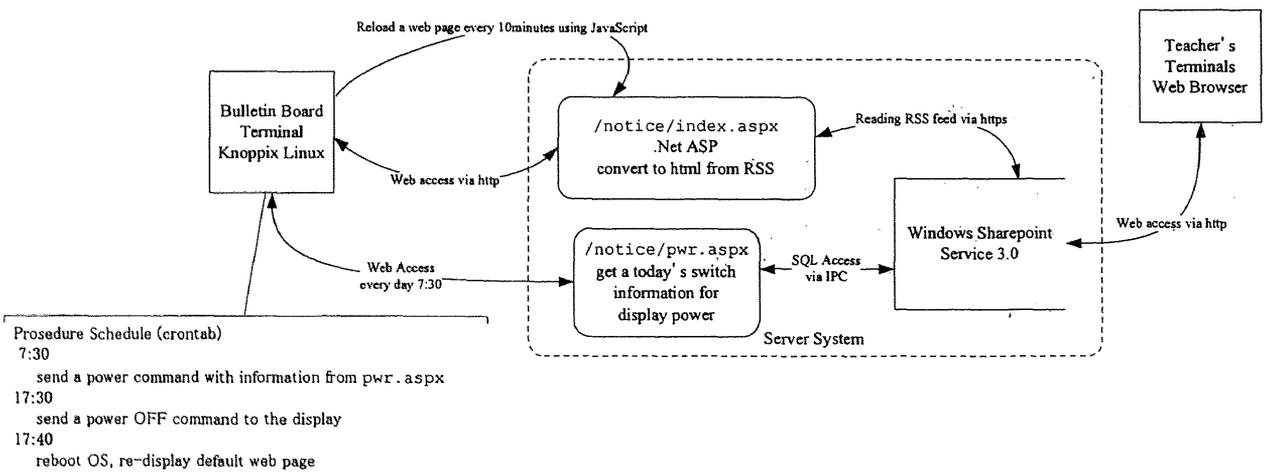


図 2. データフロー

SYSTEM DATA FLOW



3. 掲示器部の構成

掲示器は、HDD レス小型 PC とデジタルテレビで構成される。掲示器の仕様を下記に示す。

(タイプ 1, 教務室設置)

本体: Shuttle XH61, CPU: Celeron 1.8GHz,
RAM: 2GB, USB flash memory, HDMI, RS232C
モニター: Sharp 42 インチ液晶地デジTV

(タイプ 2, 昇降口設置)

本体: NEC Mate, CPU: Celeron 333MHz,
RAM: 128MB, HDD: 40GB, RS232C
モニター: Pioneer 業務用 42 インチプラズマ

(タイプ 3, B館設置)

NEC Mate, CPU: Celeron 333MHz,
RAM: 128MB, HDD: 40GB, RS232C
モニター: NEC PN-42DP3 42インチプラズマ

写真 昇降口掲示器



掲示器 PC には、OS に無償の Knoppix Linux(日本語版 6.0.1)を使い、電源を投入すると自動的に既定の Web ページを表示するようなカスタマイズをした。Knoppix は、CD-ROM、USB メモリーから起動でき、電源断処理(シャットダウン)が不要であり、不用意な電源断にも強い。また、Pentium-II 333MHz、128MB 程度の旧型パソコンでも稼働させることが可能である。

3.1. Knoppix 起動スクリプトの変更

CD-ROM ブートで稼働する OS を Live CD と呼び、一般的な構成では KDE などのデスクトップ環境が起動するように設定されている。この起動スクリプトを修正し、自動的に Web ページが表示されるようにする。knoppix 6.0.1 の起動スクリプトのシーケンスと呼出し図を図 4 に示す。

Web ページが表示されるようにカスタマイズするには、xinitrc で実行されるセッションマネージャー Xsession の代わりに Web ブラウザーの firefox が起動するように変更する。変更した xinitrc を図 5 に示す。スケジュール起動のプロシジャ、pwrquery.sh、ntpdate は crontab に登録し時限起動をする。

3.2. 掲示器電源の制御

掲示器は大画面モニターを採用する必要があるが、安価な家電用デジタルTVを流用することでコストを抑えることができる。但し、掲示器が稼働しない時は電源を落とす必要があるため、本システムに接続できるディスプレイとしては、同期信号がなくなると自動的に電源が切れるオートパワーオフ機能のあるものか、RS232C リモートコントロール機能が付いていなければならない。

ほとんどの業務用液晶ディスプレイと、中上級グレードの家庭用デジタルTVにはリモートコントロール端子が付いており、電源を PC から制御することが可能である。

本掲示器では、電源制御プログラムを毎朝 7:00 に cron スケジューラーを使って起動させ、サーバーに電源情報の問い合わせ後、RS232C シリアル出力ポートにコマンドを送信し、ディスプレイ電源を制御するようになっている。制御プログラムを図 6 に示す。

図 4. knoppix 起動スクリプトのシーケンス

KNOPPIX INIT SCRIPT SEQUENCE AND CALL TREE

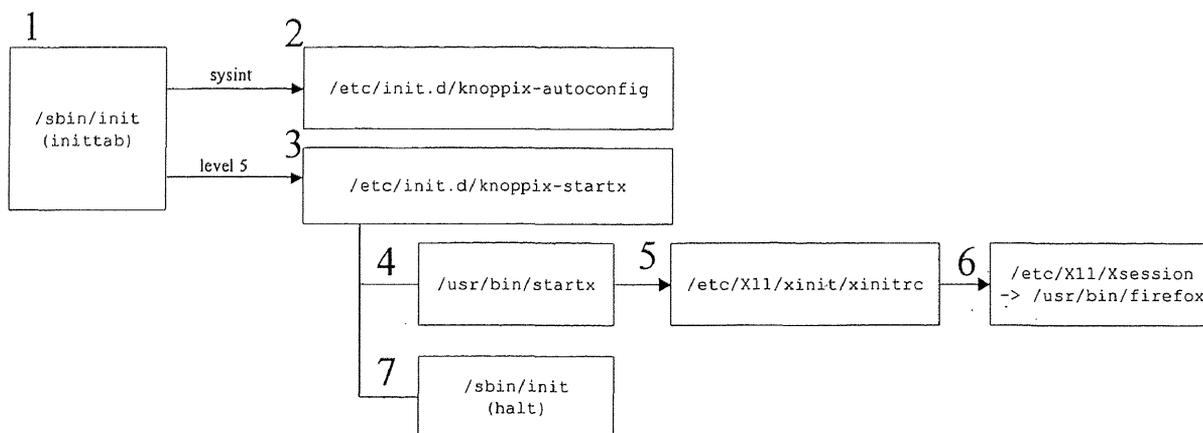


図 5. xinitrc

```

#!/bin/bash
# /etc/X11/xinit/xinitrc
#
# global xinitrc file, used by all X sessions started by xinit (startx)

xscreensaver -no-splash & # disable screensaver
unclutter -idle 2 & # auto hide a mouse pointer
exec firefox `cat /etc/sysconfig/starturl` # execute a firefox
  
```

図 6. ディスプレイ制御プログラム

```

[/etc/tvpwr/pwrquery.sh]
#!/bin/sh
/bin/rm -f /tmp/query
wget -q -O /tmp/query http://kaorin/notice/pwr.aspx # request a power query
read QUERY < /tmp/query
if [ $QUERY == "PWR=1" ]; then # parse a response
  /etc/tvpwr/pon
else
  /etc/tvpwr/pof # CRT power off
  /sbin/poweroff # self power off, wait BIOS scheduled powerup
fi

[/etc/tvpwr/pof]
#!/bin/sh
stty 9600 < /dev/ttyS0 # set a baudrate
cat /etc/tvpwr/pof.sharp > /dev/ttyS0 # send a command
sleep 1 # wait 1 sec.
cat /etc/tvpwr/pof.sharp > /dev/ttyS0
sleep 1
cat /etc/tvpwr/pof.sharp > /dev/ttyS0
  
```

今回対応した、PC 制御端子付きTVモニタの制御コマンドを下記に示す。

[SHARP 液晶 TV]

ボーレート: 9,600bps
ON コマンド: 50 4F 57 52 31 20 20 20 0D
OFF コマンド: 50 4F 57 52 30 20 20 20 0D

[Pioneer 業務用プラズマ]

ボーレート: 9,600bps
ON コマンド: 02 50 4F 4E 03 0A
OFF コマンド: 02 50 4F 46 03 0A

[NEC 業務用プラズマ]

ボーレート: 4,800bps
ON コマンド: 30 30 21 0D
OFF コマンド: 30 30 22 0D

3.3. Knoppix のリマスタリング

Live CD 動作では、1つの圧縮ファイルに収納されたシステム全体のファイルを仮想的に展開して利用される。ファイルの変更点はメモリーディスクに一時記憶する形で保管され、再起動時にすべての変更は元の状態に戻るようになっている。設定・構成を変更する場合は、システムファイルを再圧縮し、再びCDROMに書き戻す必要がある。この作業をリマスタリングと言う。リマスタリングの手順を図7に示す。

以上、生成した iso 形式ファイルを CD-R ドライブのあるパソコンへ ftp、USB メモリー経由でコピーし CD へ書き込む。

図 7. リマスタリング手順

```
1. 作業領域のためのHDDを搭載したPCで Live CDを起動する
2. 作業領域を作成

TEMPDIR=/mnt/hda1
mount /dev/hda1 $TEMPDIR
mkdir -p $TEMPDIR/source/KNOPPIX
mkdir -p $TEMPDIR/master/KNOPPIX
cp -Rp /KNOPPIX/* $TEMPDIR/source/KNOPPIX/
cp -r /mnt-system/boot $TEMPDIR/master/boot
cd /mnt-system
find . -size -10000k -type f -exec cp -p --parents
    "{}" $TEMPDIR/master ¥;
mount --bind /dev $TEMPDIR/source/KNOPPIX/dev
mount --bind /proc $TEMPDIR/source/KNOPPIX/proc
mount --bind /sys $TEMPDIR/source/KNOPPIX/sys
chroot $TEMPDIR/source/KNOPPIX

(/etc/resolv.confに nameserverを登録)
(リポジトリを更新)
apt-get update
(リポジトリが無効な場合は /etc/apt/sources.listを編集)
apt-get --purge remove cpus xpdf adrine
apt-get install xscreensaver unclutter
deborphan | xargs apt-get -y remove
COLUMNS=200 dpkg -l | grep ^rc | awk '{print $2}'
    | xargs dpkg -P
apt-get clean

3. 主要ファイルの編集
/etc/crontab
/etc/X11/xinitrc
/etc/tvprw/pwrquery.sh
/etc/tvprw/pon
/etc/tvprw/poff
~knoppix/.mozilla/firefox/knoppix/prefs.js
~knoppix/.xscreensaver

4. CDROM用ISOファイルの生成
umount /sys
umount /proc
umount /dev
exit

TEMPDIR="/mnt/hda1"
mkisofs -R -U -V "KNOPPIX_KIOSK filesystem"
    -publisher "H.Kashima Univ.of Tsukuba"
    -hide-rr-moved -cache-inodes -no-bak -pad
    $TEMPDIR/source/KNOPPIX | nice -5
    /usr/bin/create_compressed_fs - 65536 >
    $TEMPDIR/master/KNOPPIX/KNOPPIX
cd $TEMPDIR/master
find -type f -not -name shalsums -not -name boot.cat
    -not -name isolinux.bin -exec shalsum '{}' ¥; >
    KNOPPIX/shalsums
cd /
mkisofs -pad -l -r -J -v -V "KNOPPIX" -no-emul-boot
    -boot-load-size 4 -boot-info-table -b
    boot/isolinux/isolinux.bin -c
    boot/isolinux/boot.cat -hide-rr-moved -o
    $TEMPDIR/knoppix_kiosk.iso $TEMPDIR/master
```

4. サーバーの構成

サーバー機は Windows Server 2003 R2 の稼働するサーバーPC を使用する。DBMS に SQL Server 2005、グループウェア Windows Sharepoint Server 3.0(WSS3.0)を搭載する。

教職員端末、掲示器へのインターフェースはすべて Web アクセス(http)で行い、教職員は個人用 PC から Web ブラウザーで、掲示器は専用の URL アドレスにアクセスをする。設定した URL アドレスを下記に列挙する。(サーバー機のホスト名は説明の便宜上 “myserver” とする。)

[URL アドレス一覧]

WSS3.0 のトップページ

<http://myserver:81/>

生徒向け掲示器

<http://myserver/notice/Student.aspx>

電源クエリ

<http://myserver/notice/pwr.aspx>

教務室内掲示器

<http://myserver/notice/RollReports.asp>

x

教職員用掲示器

<http://myserver/notice/Faculty.aspx>

本体のスペックを下記に示す。

[サーバーPC仕様]

DELL PowerEdge2600

CPU: Xeon 5130 2GHz

RAM: 2GB

HDD: 250GB RAID

4.1. Windows Sharepoint Services 3.0 (WSS3.0)

Windows Sharepoint Services は、Windows Server に無償付属するグループウェアである。Web ブラウザー上で、掲示板、文書、スケジュール情報などを共有することが可能である。掲示板機能には RSS 配信機能が備わっており、RSS 対応ブラウザで記事閲覧が可能である。RSS は、掲示板や Web コンテンツなどの更新情報を要約形式で配信するための技術で、閲覧者は RSS へ 1

度アクセスするだけで時系列の更新情報を得ることができるものである。SharepointService の掲示板ページの例を図 8-1、掲示器出力を図 8-2 に示す。

図 8-1. 掲示板ページ画面

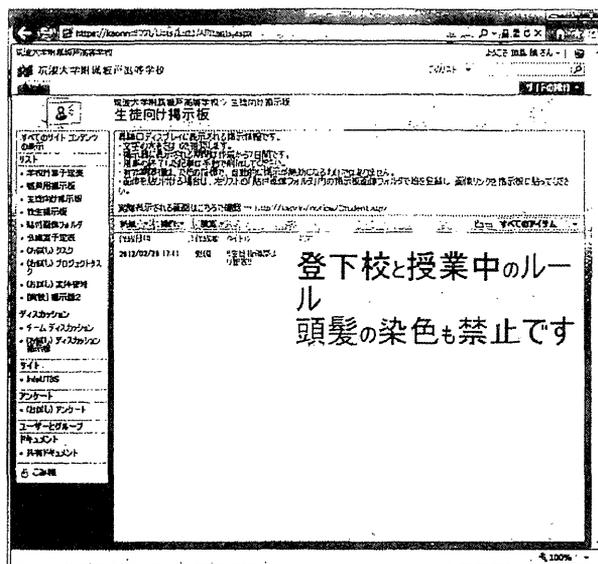
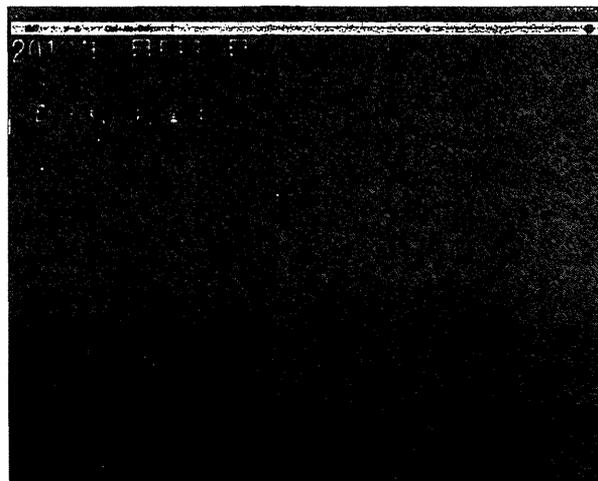


図 8-2. 昇降口掲示器画面



4.2. 生徒向け掲示器用 URL

掲示器が URL <http://myserver/notice/Student.aspx> をアクセスすることにより、掲示情報が送出される。このページには、掲示情報の他、自動画面スクロールと10分後にリロードする JavaScript が含まれており自動的に更新される。

Student.aspx がアクセスされると、C#プロシジャで日付の表示、RSS データー読込、漢字コードの UTF8 への変換、本文と時刻フォーマットの修正を行った後、xslt 変換が実行される。

xslt は、xml によって記述された文書を変換するための簡易言語である。RSS は xml 規格に則っているため、xslt スクリプトは記事情報を html 形式に変換するのに適している。

Student.aspx 内に xslt ファイルを指定したタグを記述することで、変換が実行される。xslt で変換表示する画面レイアウトを図 9 に示す。

図 9. xslt 画面レイアウト

yyyy 年 MM 月 DD 日 (gg)

タイトル	投稿者名 MM/DD hh:mm
本文	
タイトル	

4.3. 電源クエリ URL

pwr.aspx は、アクセス日で WSS3.0 のスケジュールDBを検索し、その日に掲示器の電源を ON するかどうかのクエリを返す。

電源情報は、WSS3.0 の Calendar(予定表)機能を利用しスケジュールをする。WSS のカレンダー機能には項目をカスタマイズすることができ、ユーザーは独自の管理項目を作ることができる。本校の計画表入力画面の例を図 10 に、管理項目のフィールド表を図 11 に示す。

図 10. 入力画面例

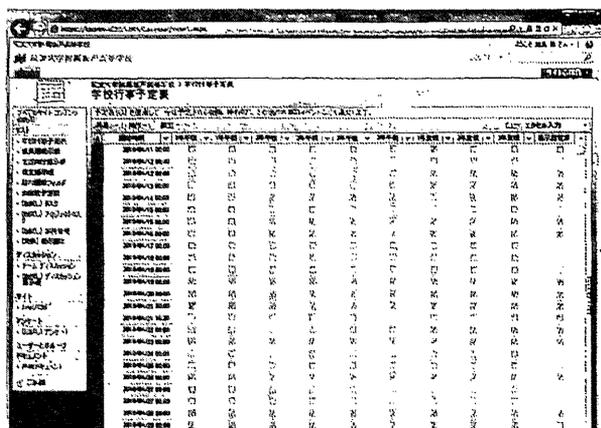


図 11. 管理項目フィールド表

Field Name	Type
タイトル	text
場所	text
開始時刻	datetime
終了時刻	datetime
説明	text
管理情報フラグ	boolean
1年午前	boolean
1年午後	boolean
2年午前	boolean
2年午後	boolean
3年午前	boolean
3年午後	boolean
1年昼校	boolean
2年昼校	boolean
3年昼校	boolean
掲示器電源	boolean
予定の対象	list
管理月	int

これらのデーターは、SQL Server の WSS_Content データーベース内の AllUserData テーブルにイベント毎で登録されている。pwr.aspx ページでは、asp スクリプトを使いテーブルから当日の電源計画を抽出する。pwr.aspx を図 12 に示す。

.net ASP において、フォームの無い単純な SQL クエリは asp サーバースクリプトのみの記述で得ることができる。また、年度更新における登録作業では、本校教務部で計画された年次計画 Excel シートから必要な事項を複写することにより行う。Internet Explorer に表示された計画表のイベントリスト画面からは、直接 Excel で記録した表をペースト可能である。

図 12. pwr.aspx

```
<asp:Repeater ID="RepQuery" runat="server" DataSourceID="dsPowerQuery">
  <ItemTemplate><%# Eval("Query") %></ItemTemplate>
</asp:Repeater>
<asp:SqlDataSource ID="dsPowerQuery" runat="server"
  ConnectionString="<%$ ConnectionStrings:学務 ConnectionString %>"
  SelectCommand="
    SELECT 'PWR='+CASE WHEN COUNT(*) = 0 THEN '0' ELSE '1' END AS 'Query'
    FROM WSS_Content.dbo.AllUserData
    WHERE tp_DirName = 'Lists/Calendar'
    AND bit5 = 1 AND bit15 = 1          <-- bit5:管理フラグ, bit15:電源フラグ -->
    AND datetimed1 = CONVERT( NVARCHAR, CURRENT_TIMESTAMP, 111)">
</asp:SqlDataSource>
```

5. 応用

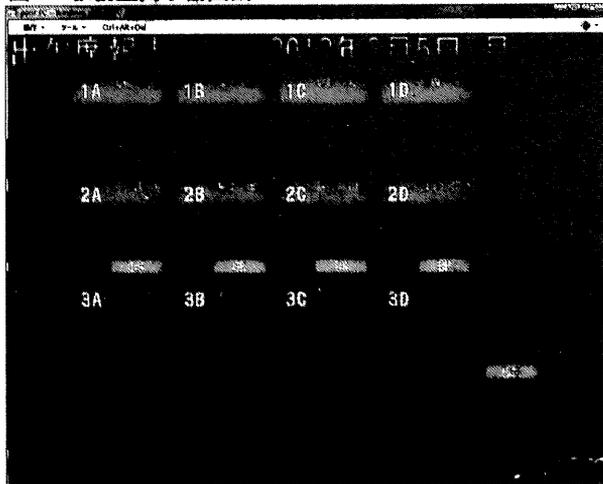
本システムの掲示器セットは、既定の URL の Web ページを表示するのみの装置であるため、html で表現できる画面であれば、集計報告等さまざまな情報掲示に応用が可能である。

5.1. 教務室情報掲示器

高等学校では、ホームルーム出欠席情報を職員に掲示しなければならない。本システムでは、.net ASP と C# プロシジャで SQL Server への集計処理を行い、出欠席数掲示ページを作成した (図 13 参照)。

校内掲示器同様、10 分毎に情報更新をし、電源計画通りに表示器の電源が制御されるようになっている。

図 13. 教務室掲示器画面



RollReport.aspx は、生徒の出欠数の集計であ

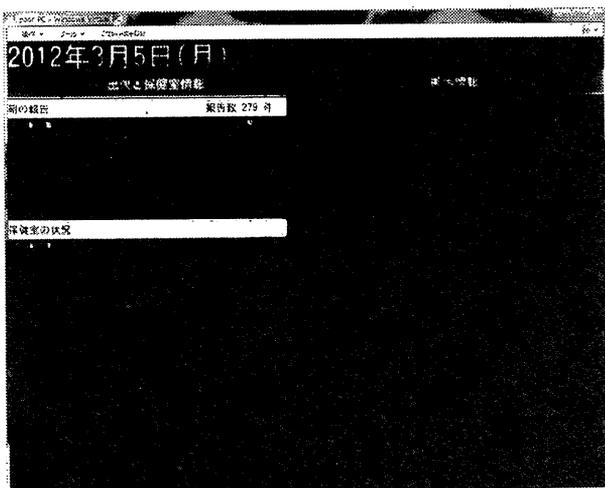
る。各ホームルームの在籍数、1 時間目科目の欠課、遅刻、有効報告数が表示される。

なお、本校の学務情報処理システム(InfoUTSS)では、出欠席の報告は、科目担当教員が任意の時間に報告する「投票方式」であるため、有効な報告が未集録の場合は集計に反映されない。

5.2. 教職員情報掲示器

Faculty.html では、朝 SHR 集計、保健室の利用状況、教職員向けの掲示情報が表示される。(図 14 参照)

図 14. 教職員向け掲示器画面

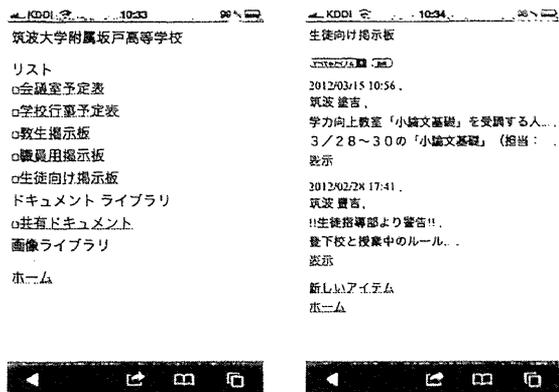


5.3. 携帯電話からのアクセス

WSS には、文字数を簡略化した Web ページを表示するモバイル端末対応機能が備わっている。インターネット回線へサービスを公開することにより、教職員は携帯電話(i モード、ezweb など)、スマートフォンを使用し、学外から掲示情報を更新すること

が可能である (図 15・16 参照)。

図 15・16. 携帯端末の画面例



6. あとがき

現状には旧式 PC(PC98NX Cerelon333MHz 128MB)を利用している場所があるが、本体の容積や、処理速度、故障率、メーカー保証、導入時の手間を勘案すると、無理に旧型を流用せずとも、新型の PC を利用する方がはるかに良い。特定の PC 本体に必要なソフトウェアや問題解決情報をインターネットで調査したり、必要な部品が不足して購入する場面があった時など、古い機種ではリソースが乏しい場合が多く、手間と時間がかかるためである。

本校昇降口の掲示エリアに従来のホワイトボードと、本機を 1 台設置している。電子掲示表示器は 1 画面のみであるのと、複数の記事がある場合は自動スクロールし全てを表示し終えるまで待たないといけないため、生徒が表示器の前で登校時しばらく立ち止まり、情報の頒布効率が良くない。これらの解決には、チャンネルの増設と他の場所への併設が有効であると考えられる。

管理の問題がある。古来のホワイトボードによる掲示は、掲示者が目の前で手書き記入をしなければならないが、特に難しい管理は不要であり、たまにしなければならないことと言えば、雑巾で水拭きしてやることくらいである。

一方、電子掲示板は遠隔書込みや、多くの掲示チャンネルを一括制御が可能な大変便利なツールであるが、コンピュータ機器を使う性質上、維持・管理をする技術者(人員)が必要となってくる。プロジェクトを立ち上げ、臨時の予算で開発導入できても、

それを活用・維持して行くためには、人員と管理費用を次年度以降も確保しなければならない。

また、あらゆる業務サービスにはバックアップ体制が必要である。例えば、電子掲示板程度であれば、“誰でも管理できる” ホワイトボードを併設しておくのが最も有効かつコストのかからない手段だ。機械が動かなくなったら、手で書きに行けばよいことだ。

ところが、コンピュータにしかできないような基幹業務系サービスともなると、“じゃあ、電卓と表計算ソフトで・・・” という訳にはなかなか行かないので、業務を一旦停止して復旧するのを待つほか手立てがなくなる。

幸い、人命がかかった、病院の生命維持装置の故障とは訳が違うから、要は、有事の際に不具合の妥当性を説明できれば良いのだが、それにしても、おおよそ考えられる有効な継続計画を準備しておくのは最低限必要である。

維持管理リソースの確保は難題であるが、今後のエレガントな指揮・運営計画に期待しつつ、後書きとする。