Vol.2012-DPS-150 No.22
Vol.2012-CSEC-56 No.22
2012/2/29

Web ベースプログラミング環境の開発

佐内修平†　　栁澤秀明†

これから情報分野の勉強を始める人の中には，プログラミングを難しいと感じている人が多い．また，これから情報分野の勉強を始めようとする人にとって，自宅のパソコンにプログラミング環境を構築する作業は困難である．近年は，クラウドコンピューティングと呼ばれる Web サービスに注目が集まっている．クラウドコンピューティングの利点は，データの一元管理ができることや，ユーザに環境構築の手間がかからないことである．これらの点から，本研究ではプログラミング環境を構築する際の問題を取り除くために Web 上にプログラミング環境を構築した．これにより，ユーザはインターネットに接続できる環境と Web ブラウザさえあれば，環境構築の作業なしで，プログラミングを行うことができる．本システムの評価として，本校の１年生に本システムを使用してもらい，システムの有用性を検証した．

Development of a Web-based Programming Environment

Shuhei Sanai†　and　Hideaki Yanagisawa†

In the people who are going to begin study of field of information, there are many people who think that programming is very difficult. However it is difficult to structure programming environment to home PC for people who began the study of the field of information. Additionally in late years, web service to be called cloud computing attracts attention. Advantage of cloud computing is to be able to unify the management of data and to have no use for environmental formulation. So in this research, I structured programming environment on the Web to straighten out problem when we structure programming environment .Therefore we could program if there are even environment to connect to the Internet and Web browser. As evaluation of this system, I had first-year student in this school use this system.

1. はじめに

これから情報分野の勉強を始めようとする人の中には，プログラミングを難しいと感じている人が多い．その理由として考えられるのは，プログラミング環境の整っている学校のパソコンを使用できる時間に限りがあり，学習できる時間が少ないということである．プログラミング学習において大切なのは，実際にソースプログラムを書いて，動かしてみることである．しかし，自宅のパソコンにプログラミング環境の整っている学生は少なく,自宅でのプログラミング学習ができていないのが現状である．さらに自宅のパソコンにプログラミング環境を構築する作業は，これからプログラミング学習を始めようとする学生にとって，コンパイラをダウンロードするためのユーザ登録やパソコンの環境変数の設定などが難しく，学習を始める前の段階で諦めてしまう学生もいる．

近年はクラウドコンピューティングと呼ばれる Web サービスに注目が集まっている．これはデータやアプリケーションを全て Web 上のサーバに置き，ユーザはクライアント端末から，Web を介してサービスを利用するものである．このクラウドコンピューティングの利点としては，データの一元管理ができることや，ユーザに環境構築や設定の負荷がかからないことが挙げられる．また，教育機関などにおいては，サーバにシステムを構築することで，全てのクライアントパソコンに環境を構築する必要がなくなり，システム管理者の負担を減らすことができる．

本研究では，プログラミング環境を構築する際の問題を取り除くために，Web 上にプログラミング環境を構築した．これにより，インターネットに接続できる環境とブラウザさえあれば，プログラミングが可能となる．また，Web 上でファイルを一元管理することで， 作成したソースファイルを時間，場所を選ばずに閲覧・編集することができる．

2. 関連研究

Web 上でプログラミングが行える環境として，WebTTY[1]というシステムがある．WebTTY はターミナルプロセスを HTML 要素経由で扱うための Web アプリケーションであり，Ajax，DHTML，C，シェルスクリプトなどを組み合わせたアプリケーションで，Web ブラウザ経由でターミナルプロセスの操作を可能にする．デモページ[2]で動かすことのできる WebTTY は，キーを押してから画面に反映されるまでに１秒以上の時間がかかるなど，レスポンス面に問題点がある．WebTTY では CUI ベースで操作を行うのに対し，本システムでは GUI によるプログラミングが可能である[3].

† 徳山工業高等専門学校　専攻科　情報電子工学専攻
Computer Science and Electronics Engineering , Advanced Course , Tokuyama College of Technology

Vol.2012-DPS-150 No.22
Vol.2012-CSEC-56 No.22
2012/2/29

ブラウザ上で Linux カーネルを動かすことのできる環境に, JSLinux[4]がある. JSLinux は，JavaScript で x86 のエミュレータを実装することで，Linux カーネルを動作させている．Web ページにアクセスすると，コマンドプロンプトが表示され，コマンドを入力することで，Linux カーネルを利用することができる．JSLinux も Webtty 同様 CUI ベースで操作を行うため，GUI によるプログラミングはできない．さらに，現時点では本システムでのターゲット言語である Java 言語に対応していない．

クライアントパソコンから，コンピュータを遠隔操作するアプリケーションとして VNC（Virtual Network Computing） [5]がある．VNC を使用すれば，ネットワーク内に存在するプログラミング環境の整ったコンピュータを遠隔操作することで，プログラミングが可能となる．VNC にはリモートアクセス用のプロトコルである RFB プロトコルが使用されている．VNC はすでに開発が終了しており，現在は様々な派生アプリケーションが公開されている．RealVNC[6]は VNC の後継アプリケーションとして開発されているアプリケーションであり，機能・操作方法なども VNC とほぼ同等である． RealVNC に商用サポートを付けている TridiaVNC[7]は，RealVNC に比べてより効果的に圧縮可能な Zlib を用いて圧縮を行っている．TridiaVNC に比べてさらに効率的にデータを転送できるのが TightVNC[8]である．TightVNC では，オリジナルの Tight Encoding を用いて，圧縮効率を上げているため，低速回線においても動作する．このような様々なアプリケーションが派生アプリケーションとして公開されている．しかし，これらのアプリケーションは，遠隔操作する側とされる側の両方のコンピュータに導入する必要がある．そのため，コンピュータの知識の少ない人が，遠隔操作する側のパソコンとなる自宅のパソコンにアプリケーションを導入する作業が必要となる．一方で本システムでは，ユーザが環境構築する必要はないため，スムーズに本システムを使用することができる．

クライアントコンピュータのデスクトップ環境をネットワーク経由で利用することのできるサービスとして，DaaS（Desktop-as-a-Service） がある．DaaS では，クライアント環境を全てサーバに集中させることで，ユーザは作業場所が変わっても，同一の環境で作業を行うことができる．DaaS には富士通の「ワークプレイス-LCM サービス」[9]，日本 IBM の「IBM Smart Business デスクトップ・クラウド構築サービス」[10]などがある．これらのサービスでは，従来のデスクトップ環境やアプリケーションの実行環境をサーバに移し，ユーザはネットワーク経由で，クラウド基板上のデスクトップ環境を使用できる．この時，サーバから転送されるのは，デスクトップ画面だけであり，ファイルはクライアントに転送されない．そのため，情報漏えい防止にも繋がる．また，ネットワーク経由で使用するため，ネットワーク環境さえあれば，時間・場所を選ばずに同一の環境を使用することができる．

Windows Azure[11]は Microsoft が提供するクラウド OS であり，ユーザはネットワークを経由して提供される開発環境を利用することができる. Windows Azure では Java, PHP，Ruby などの複数の言語を使用してクラウドシステムを構築し，そのシステムは Windows Azure 上で動作させることができる．

これらの DaaS や Windows Azure において，全ての処理はサーバで行われているため，ネットワークに繋がっていない状況で実装したアプリケーションを実行することはできない．一方で本システムでは，GUI アプリケーションの実行はクライアント側で行われる．これによって，ネットワークに繋がっていない状況でも実装したアプリケーションを実行することができる．

Google が提供しているサービスに Google Web Toolkit(GWT) [12]と Google App Engine[13]がある．GWT は Java 言語で書いたソースプログラムを JavaScript に変換してブラウザ上で実行してくれる．このツールを使用すれば，GWT コンパイラが Java クラスをブラウザ対応の HTML と JavaScript に変換してくれる. GWT 用いることで，クライアント側の UI を実装することができる．サーバ側の機能を実装するには，Google App Engine を用いる．Google App Engine では，サーブレットや JSP などの標準の Java 技術を使用して，アプリケーションを構築できる．これらのツールを利用するためには専用のアプリケーションをダウンロードする必要がある．それに比べて，本システムでは，アプリケーションのダウンロードは必要ないため，コンピュータの知識が少ない人でもすぐにシステムを使用することができる．また，Google App Engine では，使用できるクラスに限りがある[14].

3. Web ベースプログラミング環境

本節では，構築した Web ベース Java プログラミング環境の概要，構成，コマンドの実行方法，プロセス管理について説明する．なお，本論文では，CUI ベースのプログラミング環境構築について述べる．

3.1 システムの概要

本システムでは，Web 上にプログラミング可能な環境を構築するために，UNIX の様な UI を実装した．ユーザはインターネットに接続できる環境とブラウザさえあれば，プログラミング学習が可能になる．これにより環境構築や設定などユーザにかかる負担を減らすことができる．ユーザは，Web ブラウザを使用して本システムにアクセスして，ソースプログラムの作成や保存・コマンドの入力を行うことができる．システムへのアクセスでは ID とパスワードを用いてユーザ認証を行うことで，ユーザに個人の環境を提供し，さらに作成したプログラムはサーバ上に保存されるので，ユーザは，時間・場所を選ばずに作成したプログラムを閲覧・編集することができる．

サーバとの通信には，非同期 HTTP 通信を用いることで，ページ遷移をなくし，ユーザに不快感を与えない仕様になっている．図 1 に本システム全体のイメージ図を示

Vol.2012-DPS-150 No.22
Vol.2012-CSEC-56 No.22
2012/2/29

す．ユーザは，Web 上のサーバに構築されていることを意識することなく，ブラウザから本システムを使用することができる．

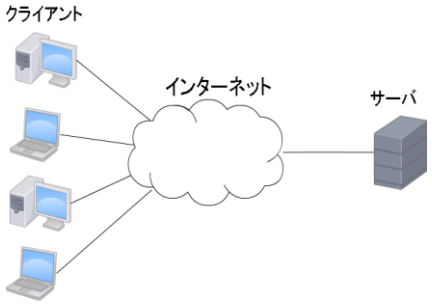

図 1 本システムイメージ図

図 2 に Web ブラウザ上にアイコン，ターミナル，テキストエディタを表示した本システムの UI を示す．ユーザは，アイコンをダブルクリックすることで，ターミナルを開き，ターミナル上でコマンド入力を行うことで，テキストエディタの起動や，プログラムのコンパイルを行う．

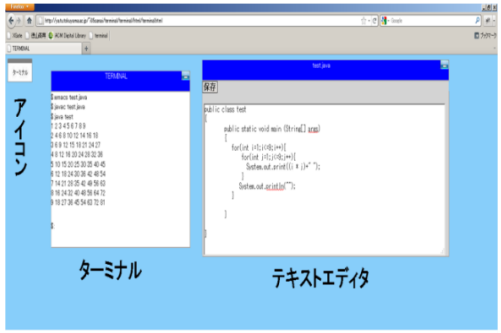

図 2 本システムの UI

3.2 開発環境及び動作環境

本システムの開発環境および動作環境を表 1 に示す．本システムは，クライアント側を JavaScript，HTML，サーバ側は PHP を用いて実装を行った．

表 1 開発環境および動作環境

	クライアント	サーバ
OS	Windows7	Sun OS 5.10
使用言語	JavaScript,HTML	PHP
使用 ライブラリ	Yahoo UI prototype.js	
動作 ブラウザ	Internet Explorer 7 以降 Firefox3.5.7 以降	

3.3 システムの構成

図 3 に本システムの処理の流れ図を示す．

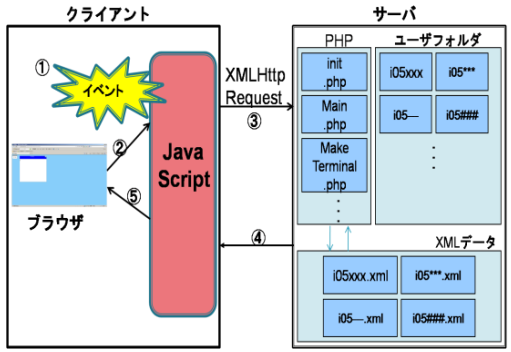

図 3 本システムの処理の流れ

ブラウザ上でマウスのクリックや，キー入力などのイベントが発生すると（①），イベントに応じた処理が JavaScript によって行われる．この時のイベントは「サーバと通信を必要とするイベント」，「サーバと通信を必要としないイベント」の 2 つに分けることができる．

コマンドの入力や，ファイルの保存など「サーバと通信を必要とするイベント」が発生した場合は，JavaScript の XMLHttpRequest を用いてサーバ側にデータを送り（③），サーバ側で PHP を用いてイベントに応じた処理を行い，その結果をクライアントに返す（④）．クライアント側では，受け取ったデータを JavaScript を用いて表示できる形に加工し，DOM を介して，ブラウザに表示する（⑤）．またサーバ側には，PHP ファ

Vol.2012-DPS-150 No.22
Vol.2012-CSEC-56 No.22
2012/2/29

イルの他に，ユーザの作成したファイルを保存するユーザフォルダ，ユーザが開いたテキストエディタや，ターミナルの情報を保存する XML データがある．XML データにより，複数のターミナルやテキストエディタを同時に使用可能となっている．
ウィンドウのドラッグ&ドロップなどの「サーバと通信を必要としないイベント」が発生した場合は，JavaScript を用いて動的に DOM を操作することで，機能を実現している（②と⑤）．
次に図 4 に本システムのディレクトリ構造を示す．本システムは，主に "html" ，"js" ，"php" ， "user" ， "u_data" の 5 つのディレクトリから構成されている．html には，システムのメインとなる html ファイルが格納されている．js ディレクトリには，クライアント側の処理を行う JavaScript ファイルが格納されている．php ディレクトリには，サーバ側で行う処理が書かれた php ファイルがある．user ディレクトリは，下位層がユーザごとのディレクトリに分かれており，ユーザが本システム上で作成したファイルはここに保存される．またユーザごとのディレクトリは，ユーザが初めて本システムにログインした際に生成される．u_data フォルダは，3.5 節で述べるユーザごとのプロセスを管理するための xml ファイルが格納されるディレクトリである．u_data ディレクトリも user ディレクトリと同様にユーザごとのディレクトリに分かれており，その中にターミナルでのプロセスを管理する xml ファイルとエディタでのプロセスを管理する xml ファイルの 2 つが存在する．

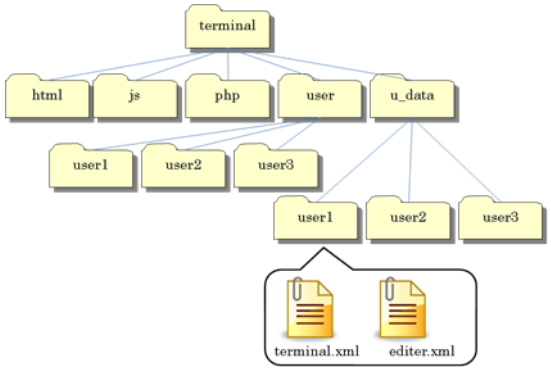

図 4 本システムのディレクトリ構造

3.4 コマンドの実行

図 5 にターミナルにおけるコマンド実行時の画面を示す．

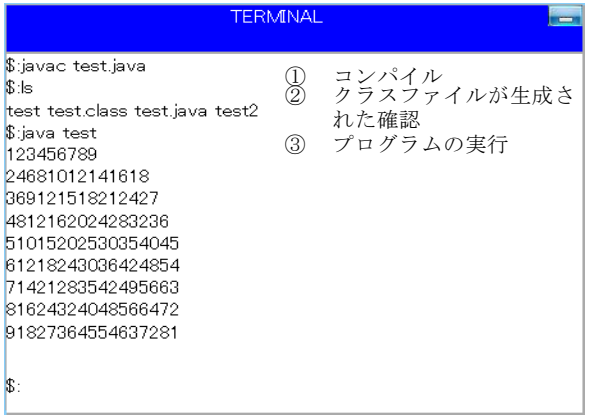

図 5 コマンド実行画面

ユーザは，CUI ベースでディレクトリ内のファイルの確認やコピーなどの簡単なファイル操作，コンパイル，プログラムの実行などを行うことができる．図 5 では，①java ファイルのコンパイルを行い，②ls コマンドでクラスファイルが生成されたことを確認．③そして，プログラムの実行を行っている．
また，本システムにおけるコマンド実行の実現方法について説明する．以下の図 6 にクライアント側のソースコードの一部を，図 7 にサーバ側のソースコードの一部を記載し，手順を説明する．

```
new Ajax.Request   ①
  'command.php',{   ②
                  method :'post',③
                  parameters:
                    Form.serialize(フォームのid),④
                  onComplete:function(result){
                  処理結果を表示⑧
                  }
});
```

図 6 command.js（クライアント側）

Vol.2012-DPS-150 No.22
Vol.2012-CSEC-56 No.22
2012/2/29

```
$command = $_POST['command']; ⑤
$result= shell_exec($comannd. . ' 2>&1'); ⑥
echo $result; ⑦
```

図 7 command.php（サーバ側）

① 非同期通信のリクエストを送信
② 送信先は comannd.php
③ 通信方法は POST メソッド
④ 送信パラメータは，コマンドライン文字列
⑤ サーバ側でデータを受信
⑥ 受信したコマンドを実行
⑦ 実行結果をクライアントへ返す
⑧ 結果を受け取り，表示

3.5 プロセスの管理

本システムでは，複数のターミナル・テキストエディタを同時に開き，使用することができる．複数のターミナル・テキストエディタを同時に管理するために，本システムでは，XML を用いて管理を行っている．ターミナルやテキストエディタを開くごとに, 3.3 節で述べた, u_data ディレクトリ下のそれぞれのユーザディレクトリの XML ファイルの中にデータを追加し，閉じるごとにデータを削除する．例として，ターミナルを一つ開いた場合に追加される XML データを以下に示す.

```
<terminal    id=123456789>
      <terminalID>terminal123456789</terminalID>
      <terminalPath    path="./user/i05sanai/"/>
</terminal>
```

複数のターミナルやテキストエディタを識別するための ID は，ターミナルやテキストエディタを開いた時の時刻をマイクロ秒単位で取得し，ID としている．ターミナルでは, terminalPath の初期値としてユーザのホームディレクトリのパスが設定されており，ユーザが cd コマンドを実行する度に，変更したディレクトリのパスが設定される仕組みになっている．ターミナルでイベントが発生する度に，この XML データにアクセスして, ターミナルごとの ID とカレントディレクトリのパスを確認するため，複数のターミナルを同時に使用することができる．

テキストエディタにおいてもターミナルと同様の仕組みを実装しており，複数のテキストエディタを同時に使用できる．

4. 評価

本節では，本システムの評価方法と結果及び考察について述べる．なお，システムのユーザ認証は，外部サーバへの SSH 認証で行った．

4.1 性能評価

本システムの性能評価として，平成 23 年度本校 1 年生 43 名に本システムを同時に使用してもらい，システム使用中のサーバの CPU への負荷，メモリ使用割合，ネットワークトラフィックを計測した．今回の評価では，システムの使用手順を記載したマニュアルをあらかじめ配布し，43 名に一斉にシステムを使用してもらった．評価に用いたサーバのスペックを表 2 にクライアントパソコンのスペックを表 3 に示す．評価の結果として，図 8 にシステム使用時の CPU への負荷の遷移の様子，図 9 にメモリ使用量の遷移の様子，図 10 に受信パケットの遷移の様子，図 11 に送信パケットの遷移の様子を示す．また図 12 にこれらのグラフを重ねたグラフを示す．

表 2 サーバスペック

サーバ名	Sun Fire T1000
CPU	UltraSPARC-T1
CPU 周波数	1GHz
メモリサイズ	8184M バイト

表 3 クライアントパソコンスペック

OS	Windows XP
CPU	Intel Core2 Duo
CPU 周波数	2.66GH
メモリサイズ	2GB
ブラウザ	Internet Explorer 7

Vol.2012-DPS-150 No.22
Vol.2012-CSEC-56 No.22
2012/2/29

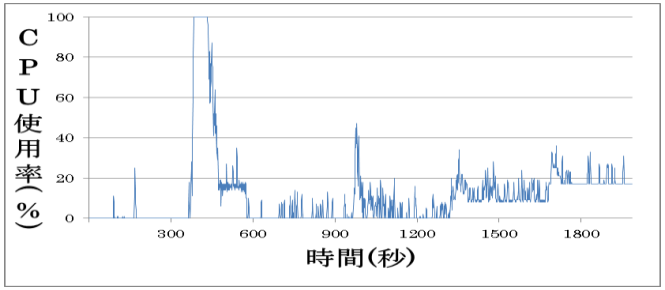

図 8 CPU 使用率の遷移

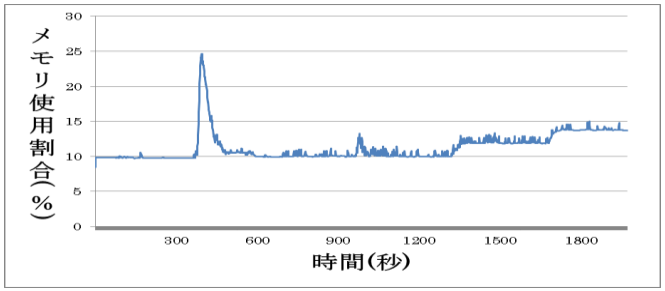

図 9 メモリ使用率の遷移

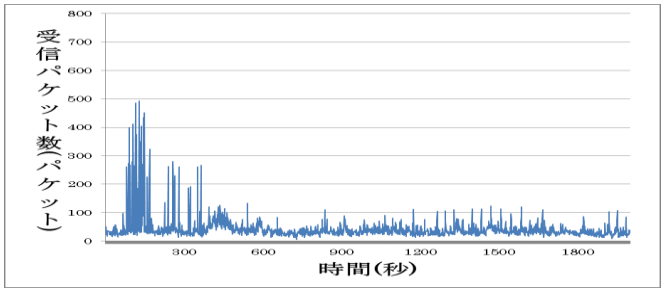

図 10 受信パケットの遷移

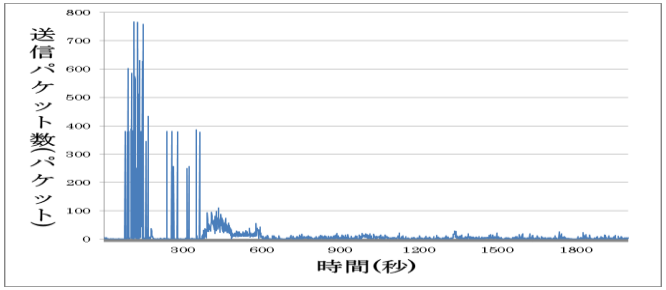

図 11 送信パケットの遷移

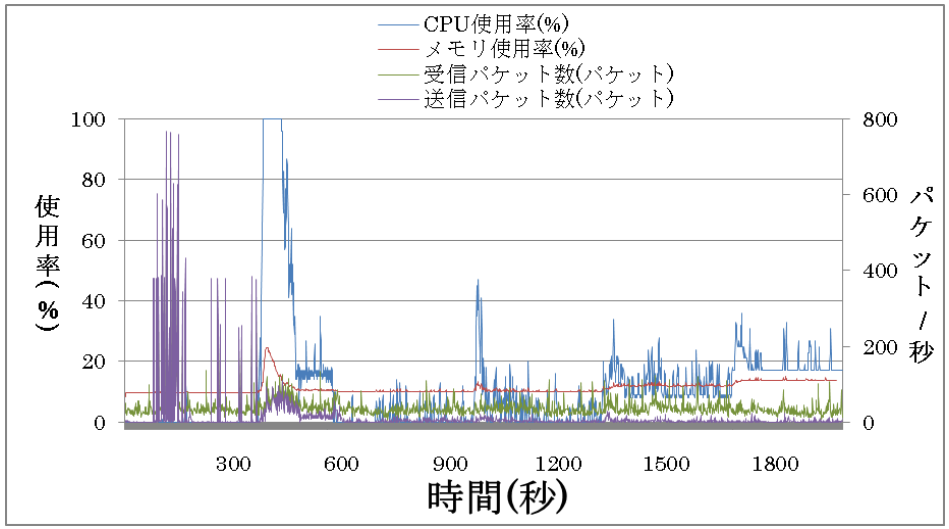

図 12 各グラフの比較

4.2 ユーザレビュー

本システムのユーザレビューは，「反応速度」，「操作性」，「画面の見やすさ」，の 3 項目を 5：とても満足 4：満足 3：普通 2：不満 1： とても不満の 5 段階で，「学校のシステムとの類似性」を 5：とても似ている 4：似ている 3：普通 2：似ていない 1：全く似ていない，の 5 段階で，「家庭で使用したいか」を 5：とても使用したい 4：使用したい 3：わからない 2：使用したくない 1：全く使用したくない，の 5 段階で評価してもらった．図 13 に評価結果を示す．

Vol.2012-DPS-150 No.22
Vol.2012-CSEC-56 No.22
2012/2/29

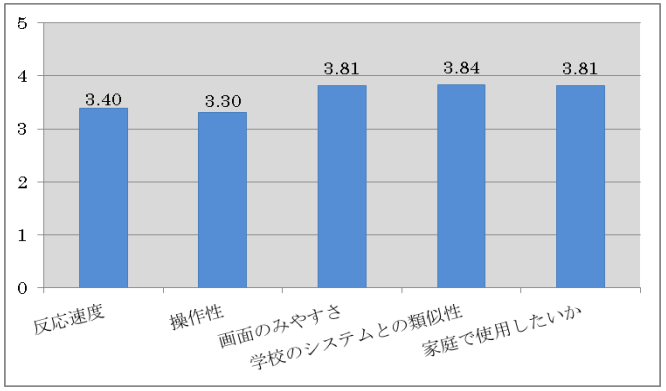

図 13 評価結果

4.3 考察

図 8 のグラフを見ると，1 か所だけ数十秒の間 CPU 使用率が 100%に達している．これは，43 名が同時にコンパイルを実行したことによる負荷がかかったことが原因と考えられる．しかし，システムを実用するとなると，40 人ほどの人数が同時にコンパイルを試みようとすることはないと考えられるので，これは問題ないものと考えられる．一方で，その後は 20%前後で遷移していることから，システム面では実用的であると考えることができる．次に図 9 のメモリ使用率を見ると，表 2 に示したサーバのメモリ量である 8184M バイトに対して，最大でも 25%程度の使用率であるため，メモリ不足によってシステムの性能が低下する可能性は低いと考えられる．図 10，図 11 のパケットの送受信量を見てみると，最も多い箇所でも，1000 パケット/秒には届いていないことから，ネットワークトラフィックがボトルネックになっているとも考えにくい．なお，受信パケット・送信パケット共に増加している箇所は，外部サーバへのユーザ認証によって増加したものであると考えられる．これらの点から，本システムは多人数による同時使用にも十分に耐えうる性能であると考えられる．

次に，図 13 のレビューの評価結果を見ると，他の項目に比べて反応速度と操作性の面での評価が少し悪かった．反応速度の面は，先に述べた同時にコンパイルを実行した時の負荷が原因でほかの項目に比べて低くなったと考えられる．操作性の面では，UI の実装が不十分であったことが原因と考えている．これは，評価のコメントにも UI の実装が不十分であるとの意見が多かったためである．今後は，ショートカットキー（保存:Ctrl+S，指定行への移動:Ctrl+G）や Tab キーを用いてのインデント機能，エディタに行番号を表示するなどして，UI の改善に取り組む．また，評価後に実行したプロセスが残ったままになる問題が発生した．この問題に関しては，タスクマネージャーのような仕組みを実装し，一定時間経過後も残ったままのプロセスは削除する仕組みにすることで，解決できると考えている．

5. おわりに

本研究では，プログラミング環境を構築する手間を省くために，Web 上に UNIX のような UI を構築した．これにより情報分野の勉強を始めようとする，パソコンの環境設定に関する知識の少ない人が，ブラウザとインターネットに接続できる環境さえあれば，プログラミングが可能となった．また Web 上に構築したことでデータの一元管理を行うことが可能となった．

評価として，本校平成 23 年度の 1 年生に同時に本システムを使用してもらうことで，システムの同時使用に対する有用性を確かめることができた．

今後の課題としてユーザから本システムに足りない点を指摘してもらうことで，より良いシステム開発につなげることができると考えている．

参考文献

1) WebTTY：http://journal.mycom.co.jp/articles/2006/11/30/webtty/menu.html
2) WebTTY デモページ：http://testape.com/webtty_page.php
3) 小寺勇司,柳澤秀明:"Web ベース Java プログラミング環境の構築",電気・情報関連学会中国支部連合大会論文集 pp.419-420(2010-10)
4) 高橋正和,「JavaScript で作った PC エミュレータ登場，Web ブラウザで Linux が起動するデモを公開」，日経 Linux，2011 年 7 月号，P.12
5) VNC：http://www.realvnc.com/
6) RealVNC：http://www.realvnc.com/index.html
7) TridiaVNC：http://www.tridiavnc.com/
8) TightVNC：http://www.tightvnc.com/
9) 富士通ワークプレイス-LCM サービス：
http://fenics.fujitsu.com/outsourcingservice/lcm/workplacelcm/
10) IBM Smart Business デスクトップ・クラウド構築サービス：
http://www-935.ibm.com/services/jp/ja/it-services/jp-so-its-virtual-infrastructure-access.html
11) Widows Azure：http://www.microsoft.com/japan/windowsazure/windowsazure/
12) Google Web Toolkit ：http://code.google.com/intl/ja/webtoolkit/
13) Google App Engine：http://code.google.com/intl/ja/appengine/
14) JRE クラスのホワイトリスト http://code.google.com/intl/ja/appengine/docs/java/jrewhitelist.html