

WEBによる教育支援システム”NOBASU”での課題プログラム評価

馬場 章 太[†] 長尾 憲 暁[†]
中西 透^{††} 船 曳 信 生^{††}

本稿では、本研究グループで開発中の WEB ベースの教育支援システム NOBASU (NetwOrk-Based Assistant System for University education) における課題プログラム評価機能の実装結果について報告する。NOBASU は、大学での学生への教育サービスの充実と、教員負担の軽減を目的としている。プログラミング教育では、学生数が多いために課題プログラムの評価に時間がかかり過ぎる、プログラムの構成の正しさの判断が困難である、といった問題が挙げられる。そこで本稿では、課題プログラムの出力結果が一意に決められる課題については、出力結果を自動で比較する機能を実装した。また、プログラム出力の検証プログラムを実行可能とすることで、プログラムの正当性をより細かく判断できる機能を実装した。2つのプログラミング課題を通じて、本提案機能の有効性を検証した。

Evaluating programing assignment by a WEB-based educationalsupport system “NOBASU”

SHOTA BABA,[†] NORIAKI NAGAO,[†] TORU NAKANISHI^{††}
and NOBUO FUNABIKI^{††}

This paper presents our assignment program evaluation functions for programming education on a Web-based education supporting system for lectures and exercises in universities, named NOBASU (NetwOrk-based Assistant System for University education). NOBASU has been developed to improve the quality of educational services while reducing loads of instructors. The programming education has difficulty for instructors, because accurate evaluations of programs submitted from many students usually take very long time. In this paper, we present two functions of helping program evaluations. The first function automatically compares the output of the program with the corresponding correct output prepared by the instructor. The second function verifies the correctness of the program by checking the CPU time and by applying the checking program made by the instructor. We have implemented them on NOBASU and verified the effectiveness through two programming assignments.

1. はじめに

大学でのプログラミング教育では、教員において、学生数が多いために学生から提出された課題プログラムの正しさを確認する作業に時間がかかり過ぎる、結果が正しくてもプログラム構成が正しいかを判断することが困難である、などといった問題がある。そのため、学生が正しいプログラムを作成できる様に、個別

に十分な指導ができていないのが現状である。

そこで本稿では、教員に対して、まずプログラミング課題の評価作業の負担を軽減するために、課題プログラムの出力が一意に決められる課題については、出力結果をその正しい結果と自動で比較する、出力結果の正当性判定機能を提供する。また、プログラム構成が正しいことを判断できるようにするために、プログラム構成の正当性判定機能を提供する。

出力結果の正当性判定機能では、diff コマンドを利用し、出力結果をその正しい結果と自動で比較する。プログラム構成の正当性確認機能では、time コマンドにより、プログラム実行に要したユーザ CPU 時間を参照可能にする。また教員が作成した検証プログラムを実行可能にする。これは担当教員が学生によって提出されたプログラムに関して検証したい項目 (生成ファ

[†] 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology,
Okayama University, E-mail: {baba,nagao}@sec.cne.
okayama-u.ac.jp

^{††} 岡山大学工学部通信ネットワーク工学科
Department of Communication Network Engineering,
Faculty of Engineering, Okayama University, E-
mail: {nakanishi,funabiki}@cne.okayama-u.ac.jp

イルのサイズが適正であるか、プログラム要求を満たしているかなど)を確認したり、担当教員が用意した入力に対して正しくプログラムが動作することを自動で検証する。これにより、プログラム構成の正当性をより細かく判断できるようにしている。

NOBASU(Network-Based Assistant System Of University)は、本研究グループで独自に開発と運用を進めている、講義・演習を対象とするWEBベースの教育支援システムである。本システムは、対面講義の存在を前提としており、それを補完・拡張し、学生の教育サービスの改善や一層の充実を図り、同時に教員負担を軽減することを目的としたものである。

NOBASUの機能には、学生サービス機能として、就職活動支援機能や成績一覧表示機能等がある。また、プログラミング課題の実行・参照には安全で効率の良い実行環境を構築するためにUser-Mode-Linux(以下UML)と呼ばれる仮想OSを用いている。UMLを利用することで、新たなホストを用意すること無く、現在使用中のサーバ用ホスト上に独立したサーバを構築している。

本論文の構成は以下の通りである。まず、2.において教育支援システムNOBASUの構成と機能について紹介する。次に、3.において課題プログラム評価のための拡張機能について述べる。4.では、提案機能の評価について考察する。最後に、5.で、本論文のまとめを行なう。

2. NOBASUの従来機能

教育支援システムNOBASUでは、教育活動である講義、プログラミング演習での利用に加え、就職活動支援機能を実現している。以下にそのための、学生サービス機能、教員サービス機能、システム管理機能を紹介する。特に、NOBASUでは安全で効率のよい課題プログラムの検証機能を実現するためにプログラミング実行・検証支援機能は、Linux上で動作する仮想OSであるUML(User Mode Linux)上でサービスを提供している。図1に、NOBASUのシステム構成を示す。本章では、NOBASU本体での機能、UML上での機能、システム構成について述べる。

2.1 NOBASU 本体での機能

2.1.1 学生サービス機能

学生への教育サービス向上の一環として、学習しやすい環境を提供するために、NOBASUでは以下の機能を提供している。

(1) 講義資料閲覧

学生は、講義で使用される資料や過去の試験問

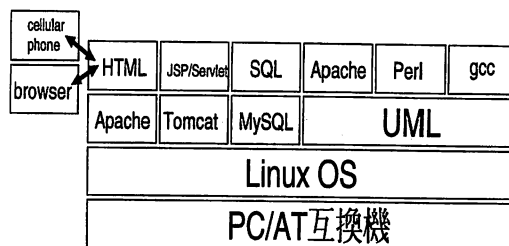


図1 NOBASU システムの構成

題等を、ダウンロードすることが可能である。

(2) 成績照会

学生は、受講している講義で課されたレポート課題や各種試験採点結果を、本機能を用いて閲覧することが可能である。

(3) レポート提出

学生は、講義、演習、講演会などで課されたレポート課題や、プログラミング課題のソースコードを提出することが可能である。その際、受け付けるファイル形式やアップロード後のファイル名を統一することで、多数の学生から提出されるレポートの管理を容易としている。

(4) 掲示板

学生が講義内容などに関する質問や意見を掲示板に書き込み、学生または教員が回答する機能を提供する。掲示板を用いて意見交換を行うことで、学生同士や学生と教員間でのコミュニケーションを促進する。

(5) キーワード提出

学生の各講義での復習を目的として、講義中に現れたキーワードの提出を行なう。この機能は学生の出席チェックを兼ねることも可能である。

(6) 就職活動支援

就職活動をしている学生は、就職希望調査書の提出や求人情報・会社説明会情報を閲覧可能である。

2.1.2 教員サービス機能

教員の教育活動における負担軽減のために、NOBASUでは、以下の機能を提供している。

(1) 受講生登録

教員が、担当する講義・演習を受講している学生の登録を行う。これは、Excel ファイルを CSV 形式でアップロードすることで可能である。

(2) 講義成績登録

教員が、講義で課したレポート課題や各種試験の成績をデータベースに登録する。この登録結

果により、学生の学習状況の把握や、各学生への成績開示が可能となる。

- (3) メール送信
教員が、受講生への連絡事項（休講予定・試験日程など）を、学生の登録アドレスに対して一斉に、または個別に送信することが可能となる。
- (4) 講義資料アップロード
教員が講義で使用する資料や過去の試験問題などをアップロードし、学生への提供を行う。
- (5) キーワード表示
学生が提出した講義のキーワードを一覧表示し、学生の理解度の確認等を行う。またキーワード提出状況に関する、講義全体での一覧表を作成し、学生の出席確認に代えることも可能とする。
- (6) 就職活動支援
教員は、学生の就職希望調査書を閲覧する。また、会社説明会の日程などをアップロードし、学生へ提供することが可能である。

2.1.3 システム管理機能

システム管理者による、学生・教員利用のための前準備やシステム管理を行なうために、NOBASU では以下の機能を提供する。

- (1) ユーザ登録
NOBASU では、事前に利用者全員のユーザ登録を行なう必要がある。そのため、本システムでは、学生・教員・管理者の全てのユーザをその利用前に、各ユーザに関する情報（学生番号、氏名、パスワード、メールアドレス、ロール）を記述したファイルを、WEB ブラウザからサーバへアップロードすることにより、登録する。
- (2) 講義情報登録
本システムを利用する講義の情報（講義番号、講義名、担当教員、開講区分）を登録する。登録方法は、ユーザ登録の場合と同じである。
- (3) データベース管理機能
本システムでは、様々なデータを MySQL[4] で管理している。WEB ブラウザ上からそれらのデータ管理を行なえるようにすることで、システム管理者の負担を軽減する。

2.2 UML 上での機能

UML 上において、本論文での拡張機能を実装する前に実装されている機能を示す。

- (1) プログラムソース参照
教員は、学生が提出した課題に対するプログラムソースを WEB 上で参照することが可能である。

- (2) プログラムのコンパイル・実行
学生から提出されたプログラムソースを本システム上で実際にコンパイル・実行して、その動作を確認する。まずプログラムのコンパイルと実行の指示は WEB ブラウザから行い、その実行結果はログに保存される。各プログラムで出力ファイルが作成される場合には、ログと共に WEB 上より参照可能である。さらに、個別に実行するだけでなく、課題毎に一括して実行させることで、実行の手間を軽減している。

2.3 NOBASU の実装環境

現在稼働している NOBASU 本体のサーバ環境は、表 1 の通りである。

CPU	Pentium4 3.20GHz
メモリ	1GB
OS	VineLinux3.1
kernel	2.4.27-0v17
java	1.5.0
WEB サーバ	Apache2.0.50
サーブレットコンテナ	Tomcat5.0.28
データベース	MySQL4.0.25

また、UML 環境は、表 2 の通りである。UML のフォルダ構成は図 2 のように科目、課題、学生毎のフォルダに別けられている。

OS	User-Mode Linux
kernel	2.6.13.1
distribution	Gentoo Linux

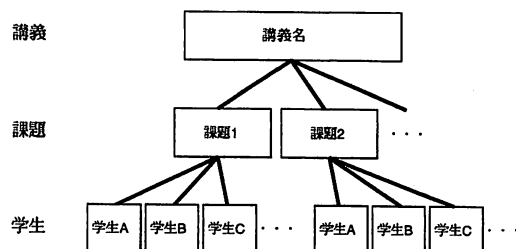


図 2 UML のフォルダ構造

UML の CGI は本研究で機能拡張しており、それらを図 3 に示す。UML 上での CGI は『講義の表示と選択』、『課題の表示と選択』、『提出者の表示と選択』が行なえるようになっている。そして、各提出

者ごとに『一括実行』、『個別実行』、『ソース・ログの表示』、『出力ファイルの参照』が可能となっている。ここで、『一括実行』と『個別実行』を行なう際にはプログラムを実行するプログラムが呼び出さようになっている。

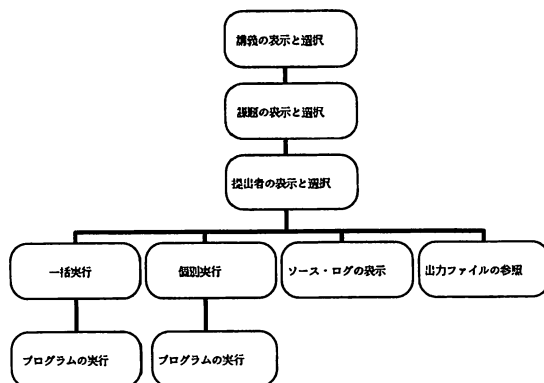


図 3 UML の CGI 構造

3. 課題プログラム評価のための拡張機能

本章では、まず本論文で提案する課題プログラム評価のための NOBASU の機能拡張の背景と目的について述べる。次に、拡張した機能の構成と実装について述べる。

3.1 従来機能の問題点

従来の NOBASU では、教員に対して、プログラミング課題の確認のためにプログラムソースの参照、実行した際の標準出力とエラー出力結果の参照のみが可能である。これより以下のような問題が考えられる。

- (1) エラーのあるプログラムソースのチェックに時間がかかる。
- (2) 出力結果が一意となる課題に対する出力結果の確認にも手作業のため、時間がかかる。
- (3) 出力結果が正しくても、プログラム構成が正しいとは限らず、課題プログラムソースのより正確な評価が困難である。
- (4) プログラム構成の正しさの評価には、ソース解読などにより判断しなければならず、さらに時間がかかる。
- (5) 以上の結果、問題のある学生に対しても十分な指導の時間が取れない。

3.2 要求機能

前節で述べた問題点を解決するために、教員に対し

てまず、出力結果が一意に決められている課題については出力結果の正しさをシステムが自動で判定する機能を提供する。次に、プログラム構成の正しさの評価するために、その実行時間が参照可能となる機能を提供する。また、担当教員が作成した検証プログラムを実行可能にする。これにより、プログラムの正当性をより細かく判断できるようにしている。

3.3 出力結果の正当性判定機能

出力結果の正当性判定機能では、プログラムの出力結果が一意に決められる課題について、学生の提出したプログラムの出力結果と、事前に与えるその正しい結果との比較を自動で行う。本機能は、プログラムが実行され、結果が出力される際に、diff コマンドで判定することにより実現している。ここで、diff コマンドとは2つのファイルを比較し、それらの違いを出力するコマンドである。この機能により教員はプログラムのソースコードの確認を行う前に、その出力結果が正しいか否かを知ることが可能である。

3.4 プログラム構成の正当性判定機能

出力結果の正当性判定機能により、プログラムの出力結果が正しいと判断された場合でも、無駄なループや無駄な処理を含み、プログラム構成が正しくない場合がある。そのようなプログラムを発見するために、NOBASU において、プログラムを実行する際に time コマンドを利用し、プログラム実行時に消費されたユーザ CPU 時間を参照可能としている。次に、実行時間の参照だけではプログラム構成の正当性を判断することが困難な場合もあるため、教員が独自に作成した検証プログラムを各学生からの課題プログラムソースに關して実行できるようにしている。そのため、図 2 の各課題フォルダに、担当教員の作成した検証プログラムを課題ごとに配置する。そして学生プログラムを NOBASU により、学生プログラム実行後に検証プログラムを実行するようにしている。この機能により、教員は自分が確認したい項目に關して自由に検証することが可能となる。これによりプログラム構成の正当性判定の信頼性が大幅に上がるものと考えられる。

4. 評価

本章では、第 3 章で提案した NOBASU の課題プログラム評価のための拡張機能を実装し、その検証を行う。今回、検証を行なったプログラミング課題は 2005 年度後期に 2・3 年次生向けに開講された「ネットワークセキュリティ」と、2006 年度 3 月に研究室に配属された 4 年生を対象として行なわれた、「プログラミングテスト」である。

4.1 「ネットワークセキュリティ」での検証

4.1.1 課 題

「ネットワークセキュリティ」の課題は、ある条件を満たす素数を生成し、その素数を用いて生成された鍵を用いてエルガマル暗号によりメッセージを暗号化し、暗号化されたメッセージを復号するプログラムの作成である。

4.1.2 出力結果の正当性判定機能

まず出力結果の正当性判定機能により、暗号する前のメッセージと復号した後のメッセージが同じか否かを自動で判定する。図4のように課題を提出した学生一覧を表示した時点で、出力結果が正しいかどうかが一目でわかるようになっている。(図では学生番号と名前の部分を隠してある。)

教育支援システム「NOBASU」

提出者	提出日時	提出内容	提出結果
001	001	課題 1: エルガマル暗号化/復号化	match
002	002	課題 1: エルガマル暗号化/復号化	match
003	003	課題 1: エルガマル暗号化/復号化	match
004	004	課題 1: エルガマル暗号化/復号化	match
005	005	課題 1: エルガマル暗号化/復号化	match
006	006	課題 1: エルガマル暗号化/復号化	match
007	007	課題 1: エルガマル暗号化/復号化	match
008	008	課題 1: エルガマル暗号化/復号化	match
009	009	課題 1: エルガマル暗号化/復号化	match
010	010	課題 1: エルガマル暗号化/復号化	match
011	011	課題 1: エルガマル暗号化/復号化	match
012	012	課題 1: エルガマル暗号化/復号化	match
013	013	課題 1: エルガマル暗号化/復号化	match
014	014	課題 1: エルガマル暗号化/復号化	match
015	015	課題 1: エルガマル暗号化/復号化	match
016	016	課題 1: エルガマル暗号化/復号化	match
017	017	課題 1: エルガマル暗号化/復号化	match
018	018	課題 1: エルガマル暗号化/復号化	match
019	019	課題 1: エルガマル暗号化/復号化	match
020	020	課題 1: エルガマル暗号化/復号化	match

図4 課題の提出者一覧表示

4.1.3 評 価

表3に、本課題で提出された4つのプログラムでの出力結果の正当性判定結果、プログラム実行時の消費時間、検証プログラム実行結果を示す。また、評価の基準として担当教員が作成したプログラムの結果も示す。検証プログラムでは、生成された鍵が素数になっているか、素数同士の関係が条件を満たしているかなどを検証した。

表3 ネットワークセキュリティ課題での結果

作成者	出力結果	ユーザ CPU 時間 (s)	検証 プログラム
教員	match	23.55	正常
学生 A	match	17.21	正常
学生 B	not match	23.77	異常
学生 C	match	0.1	異常
学生 D	match	12.75	異常

表3において、学生Aは出力結果が正しく、担当教員のプログラムよりも実行時間は少し短い上に、検証プログラムの結果も正しいということが明らかである。鍵を生成する際の素数生成アルゴリズムが優れている

ということが推測可能である。

学生Bでは出力結果が正しくないことより、プログラムが異常であることが明らかである。さらに、検証プログラムの結果より担当教員はプログラムの異常箇所を特定でき、参照すべきソースを絞って評価を行なうことが可能である。

学生Cでは出力結果は正しいが、実行時間が異常に短いことからプログラムが異常である可能性を推測可能である。また、検証プログラムの結果からもプログラムが異常であることが明らかである。実際、このプログラムは素数判定が省略されており、素数が生成されていなかった。

学生Dでは出力結果が正しく、実行時間も少し短い。しかし、検証プログラムの結果よりプログラムが異常であることが判明した。

4.2 「プログラミングテスト」での検証

4.2.1 課 題

「プログラミングテスト」の課題は、標準入力または入力ファイルにより指定されたIPアドレスのドメイン名をトップレベルから大文字/小文字を区別せずに辞書順にし、標準出力に出力するプログラムの作成である。

4.2.2 検証プログラム

図5に、「プログラミングテスト」の課題において提出された1つのプログラムに対する、検証プログラムの実行結果を示す。

教育支援システム「NOBASU」

提出者	提出日時	提出内容	提出結果
001	001	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
002	002	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
003	003	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
004	004	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
005	005	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
006	006	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
007	007	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
008	008	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
009	009	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
010	010	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
011	011	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
012	012	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
013	013	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
014	014	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
015	015	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
016	016	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
017	017	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
018	018	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
019	019	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match
020	020	課題 1: IPアドレスのドメイン名を辞書順に並べ替える	match

図5 プログラミングテスト課題での結果

図5では、課題プログラムが指定された仕様を満たしているかを否かを順次検証するため、ファイル指定と標準入力の両方またはいずれかに対応しているか、多数行(5万行以上など)をもつファイルを扱えるか、読み取り禁止ファイルを与えても異常をきたさないかなどを検査している。これらの結果を参照することに

より、一目で課題プログラムの評価を行なうことができる。また、ソースコードを見て評価する場合でも、教員はどの部分が正しく書かれていないかなどを推測した上で、プログラムを評価できたため評価時間を大幅に短縮可能である。また、50000 行を持つファイルの処理においては、効率の悪いソートアルゴリズムを採用してしまうと非常に時間がかかるため、ユーザ CPU 使用時間を測定し参照することにより、ソートアルゴリズムの評価も容易に行なうことが可能であった。

5. ま と め

本稿では、大学でのプログラミング教育において、教員に対して課題プログラム評価の支援機能の拡張結果を報告した。今後の課題として、より利用しやすいインターフェイスの実装、プログラムソースの自動解析機能があげられる。

参 考 文 献

- 1) Sun Microsystems,
<http://java.sun.com/applets/>
- 2) Japan Apache User Group,
<http://www.apache.jp/>
- 3) B.Laurie, P.Laurie(著), 田辺茂也 (監訳), 大川佳織 (訳), Apache ハンドブック, オライリージャパン, 東京, 2003.
- 4) MySQL.com, <http://www.mysql.com/>
- 5) The User-mode Linux Kernel Home Page,
<http://user-mode-linux.sourceforge.net>
- 6) 深沢千尋 (著), すぐわかる Perl, 技術評論社, 東京, 2003.
- 7) 高橋紀行, 船曳信生, 中西透, “講義・演習を対象とした WEB ベースの教育支援システム,” 信学技報, vol.ET2004-44, pp.45-50, 2004.
- 8) 長尾憲暁, 船曳信生, 中西透, “仮想 OS を用いたセキュアなプログラミング課題検証システム,” 信学技報, vol.ET2005-45, pp.41-46, 2005.