

体験学習モデルによる OS の学習システム

米田 孝文^{*1}

早川 栄一^{*2}

^{*1}拓殖大学大学院 工学研究科 電子情報工学専攻

^{*2}拓殖大学 工学部 情報工学科

^{*1}yoneda@os.cs.takushoku-u.ac.jp, ^{*2}hayakawa@cs.takushoku-u.ac.jp

本稿では、OS の資源管理方式を可視化した、体験学習モデルによる OS の学習システムについて述べる。OS を学ぶ学生を対象に、OS の振舞いを体験しながら資源管理方式を学習できる仮想環境を提供する。従来の OS 学習は、その全体像を捉えるのが困難であった。本システムは、OS の全体像や概念と実装の対応関係などを明確にし、それを学習に結びつける。これらによって、学習者は実ハードウェアの制約を受けずに、資源管理の全体構成や各アルゴリズムなどを理解することが可能になる。今回はプロトタイプとして、ファイルシステムを中心に実現を行った。

A Learning System for Operating System modeled on learning by experienced

Takafumi Yoneda^{*1}

Eiichi Hayakawa^{*2}

^{*1}Graduate School of Engineering, Takushoku University

^{*2}Takushoku University

^{*1}yoneda@os.cs.takushoku-u.ac.jp, ^{*2}hayakawa@cs.takushoku-u.ac.jp

This paper describes the design of an Operating System (OS) learning system that visualizes its resource management based on experience. Our system supports a virtual environment that makes learners manage hardware resources as a real operating system. On experiencing the behavior of an OS, learners learn to understand the algorithm of resource management in operating system course. Traditional learners get into difficulties in realizing the overall structure. Our system supports the understanding of the structure of OS and the relationship between concept and implementation without the limitation of hardware resources. This prototype system implements that mainly of the file system facilities.

1 はじめに

現在の学習者は、OSの資源管理方式を学習するときに戸惑う問題がある。例えば学習者が、それを学習する場合、第一に教科書、参考書を読んだり、人に教わるなどの概念から学習する手段がある。第二には、実際のOSのソースコードを読んで、実装レベルから学習する手段がある。しかし、このどちらもOSの全体像が見えづらい。また、概念と実装の「繋がり」が見えないまま学習が終わってしまう。「繋がり」とは、概念はどのようにコーディングに結びつき、またコードはどのような概念を元に記述されていくのかを導く思考の道筋である。つまり学習者は、OSの全体像や概念と実装の繋がりを学習できる教材を持っていない問題を抱えているのである。

本報告は、体験学習モデルによるOSの学習システムについて述べる。

2 問題分析

OSを熟知する一番の手段は、自分でOSを実装すること、もしくは既存のOSのソースコードを解析することである。しかしOSのソースコードは巨大である。そして、その動作は動的であるため、コードからコードへ頻繁に制御が移る。そのため学習者は、その流れを追うのが困難で、OS全体を把握するためには長い期間を要してしまう。最悪の場合、学習者は学習意欲をなくしてしまう。これは問題である。我々はOS学習において、概念と実装の繋がりとデータ構造の関係を見せることによって、この問題を解決できると考えた。

3 本研究の目的

本研究の目的は、学習者がOSの役割を引き受け、OSの振舞いを体験することによって、資源管理方式を学習できる環境を実現することである。その際、OSの概念と実装を可視化してシステムを実現する。これによって学習者は、資源管理方式を自分の目と手で体験することができるようになる。そして学習者は、多くの資源管理の関係とその仕組み

を視覚的、体験的に学習し、1と2で挙げた問題を解決できると考えた。本来、OSのない時代、その管理は人間が行っていた。その管理者は、資源管理について一から十まで知っていたのである。本研究における人間がOSの役割を引き受ける環境の実現は、その時代の再現である。これは、教材として扱うに十分で、妥当な選択であると考ええる。

4 設計方針

システムの設計方針について述べる。

(1) OSを学習者に置き換える

学習者がOSの立場を担い、仮想OSとして資源管理を行うことによって、その学習効果を高める。そのため多くの資源管理方式を可視化し、それらを仮想ハードウェアとして学習者に提供する。また実際のユーザの立場は仮想化し、仮想ユーザとしてシステムに組み込む。これらによって、コンピュータ利用における自然な環境が実現できると考えた。そして学習者は、OSの側の視点から資源管理方式に触れることができるようになる。

(2) OSの振る舞いを体験しながら学習できる

(1)の仕組みは体験の仕組みが中心を占めており、それを実現するだけで学習の仕組みと連携できない。そこで本システムは、仮想ユーザが仮想OSに要求を出すという行動を、学習者にとっては問題が与えられたということにする。そして、この問題を解釈して対処するための仕組みを組み込み、学習者が個々に学習できる環境を構築する。仮想ユーザは、学習者の問題の答えへの正誤判断も行う。これらによって本システムは、体験学習における学習の仕組みを実現できる。

また本システムは、扱う情報に関して学習内容に関係ない部分はできるだけ抽象化したい。学習内容の真意を明確にできる環境を実現するためである。抽象化するものは、例えばポインタがある。ポインタは、その情報を忠実に数値で示さずとも、その仕組みを使うことができる。例えばC言語などは、配列という形で抽象化されている。本システムでは、

GUI コンポーネントなどを使って抽象化できる。また本システムは、学習者に、体験、発見的学習のできる環境を提供したい。従来の学習にある想像に頼る面を減らし、より深い理解の得られる環境を構築するためである。そのために本システムは、設計方針の (1) を実現する。これらによって本システムは、体験学習における体験の仕組みを実現できる。

(3) OS の基本機能について学習させる

次の三点を本システムの学習内容の中心とする。

- ・ プロセス管理
- ・ メモリ管理
- ・ ファイルシステム

この 3 点を組み込むことによって、OS の基本機能の学習を確保できると考えた。目標は、本システムを使えば、参考書などの補足資料を必要とせず、OS の中の限定した部分ではあるが、その学習内容を完結することである。これによって、一つの学習教材として、システムを実現できる。

(4) 情報工学系の大学二年生以上を対象とする

学習させる内容は、情報処理の専門家が欲しい知識であること。また、その理解に予備知識が必要になることを考えた。予備知識は、本システムで見せるものが、本来の OS の中では何を示しているものなのかを想像できる程度の知識である。よって対象者は、基本ソフトウェアの制御プログラム (スワッピング方式や仮想記憶の記憶管理など) を学んだ経験のある学習者であり、情報工学系の大学二年生以上と想定する。

(5) 学習者の操作内容を補足する

本システムは、学習内容と関係ない情報を抽象化することに決めた。つまり学習者は、本システムの中で本来の OS に忠実でない振舞いを体験することがある。そこで、そういった部分は補足する必要がある。

5 設計

システムの設計について述べる。

5.1 システムの全体構成

(1) 仮想ユーザと仮想ハードウェア

本システムは、設計方針の (1)、(2) を満足するために、仮想ユーザと仮想ハードウェアの二つで構成する (図 1 参照)。仮想ユーザは、学習者に課題を提供する。課題は、資源管理操作の体験学習における学習内容のことである。仮想ハードウェアは、学習者に GUI コンポーネントとグラフィックスを使って可視化した資源管理モジュールを提供する。これは学習者が、OS のデータ構造を操作し、学習するための GUI インタフェースである。これらによって本システムは、体験学習モデルによる OS の学習環境を構築できる。

(2) 仮想 OS

設計方針の (1) を実現するために、仮想 OS は UNIX を想定し、学習者は UNIX の資源管理アルゴリズムを守らないといけない。UNIX を想定する理由は、そのシステムが長い歴史を経て完成されたシステムだからである。また、Windows のような過剰なサービスを持っていない。そのため OS の仕事に必要な、限定した機能だけで実現されている。この特徴は、学習環境を実現したい本システムにおいて最適である。なぜなら学習において、分かりやすさは学習効果を一層高める。また仮想 OS の具体的な部分は、学習者側で実装できるようにする。そして仮想 OS のデータ構造は、学習内容に必要な最低限のものを用意する。これらによって本システムは、OS 学習に必要な限定した仕組みを組み込んだ、学習者が自分で考えることのできる学習環境を構築できる。

(3) 補足説明の表示

設計方針の (5) を実現するために、学習者の操作に対して補足説明を見せる機能を組み込む。ただし本システムは、設計方針の (2) で示したように、学習内容に関係ない情報を抽象化したい。ここで文字情報を使って補足説明を示すと、結局その実現を妨げる恐れがある。そのため、この補足の内容はでき

るかぎり短く、簡潔に示す必要がある。

(4) 学習者の操作をログに取得

学習者が、ある課題に対して一通り学習して、その間違いを確認したい場合、資源管理操作の履歴情報が必要になると考えた。そこで、タイムスタンプと共に、学習者の操作をログに残す。内容は、学習者の選択した資源管理モジュールの名前や設定した値とする。これによって本システムは、学習者が自分の学習の履歴を確認する仕組みを組み込むことができる。

(5) 学習者の基本的な操作の流れ

学習者の基本的な操作の流れは、次の四つとする。

1. 課題プールから課題を選択する
2. 資源管理モジュールを選択する
3. 課題の内容に沿って資源管理操作を行う
4. 資源管理操作の結果を仮想ユーザに渡す

学習者は、これらの1~4の操作を繰り返し行い、体験的にOSの資源管理方式を学習する(図1参照)。1で課題の選択、2で解答手段の選択、3で試行錯誤、4で解答の四つの手順を実現している。

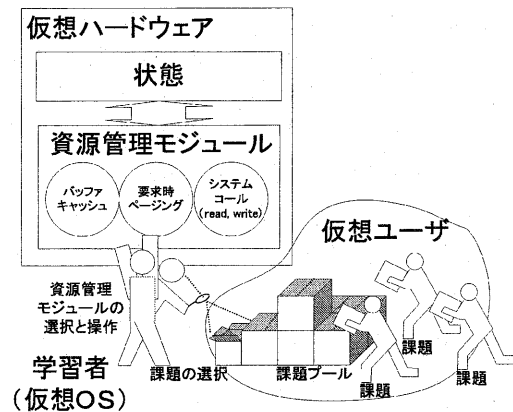


図1: システムの全体構成

5.2 仮想ユーザの全体構成

仮想ユーザは、仮想OSに、自由に、しかし意図を持って要求を出している。本システムは、この仕組みをGUIコンポーネントを使って、問題の提供と

正誤判断の機能を持たせることで実現する(図2参照)。目標は、実際のユーザが、実際のOSに要求を伝えるような仕組みを表現することである。

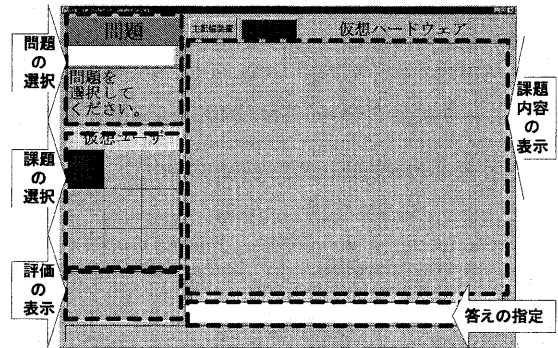


図2: 仮想ユーザ

5.2.1 機能

仮想ユーザは具体的に、次に示す機能を使って、問題の提供、正誤判断の仕組みを実現する。

(1) 課題の選択

学習者が課題プールの中から、任意に課題を選択できるようにボタンを提供する。ボタンが押されたら、そのボタンと対応する課題内容をテキストエリアに、またその課題の答えを入力するためのテキストフィールドとボタンを表示する。これらによって学習者に、課題の選択と解答の手段を提供できる。

(2) 課題の提供

資源管理操作の課題は、課題一つに対してテキストファイル一つを用意し、その内容はシステムコールで構成する(図3参照)。システムコールを採用した理由は、それ一つで、その目的を明確に示すことができるからである。

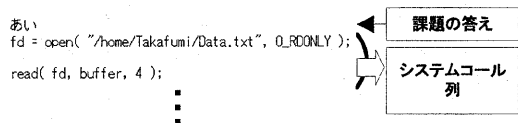


図3: 課題ファイルのフォーマット

(3) 問題の提供

資源管理操作の問題は、問題一つに対してディレクトリ一つを用意し、その中には複数の課題ファイルを用意する。問題は例えば「バッファの役割」などの抽象的な主題をディレクトリに設ける。主題は、学習内容の絞込みを行うための名前付けであり、仮想ユーザの意図を示すものではない。仮想ユーザの意図は、課題をいくつもこなしていきながら自然と明確になる。また、そのディレクトリ内には、課題を提供する順序を決めるテキストファイルの一つを用意し、課題を提供する流れを作る(図4参照)。このファイルを順序ファイルと定義する。

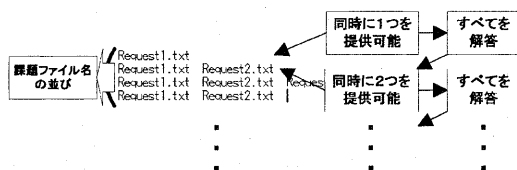


図4: 順序ファイルのフォーマット

(4) 答えの指定

学習者は、課題内容に沿って資源管理操作を終えて結果が分かったら、その結果が正しいかどうかを判定してもらうために仮想ユーザに伝える。(1)で述べた解答の手段で実現できる。

(5) 評価

学習者から資源管理操作の結果を受け取ったら、それが課題の答えと一致しているかどうかを判定し、学習者の結果を評価する。評価は、学習者の結果が課題の答えを満たしたか、満たしていないかによって「YES」、「NO」の二通りのメッセージを示す。本システムは、学習者の操作の過程を評価しない。設計方針の(2)で示したように、発見的学習のできる環境を実現したいからである。これらによって本システムは、正誤判断の仕組みを実現し、学習者に、課題の答えの正解、不正解を伝える。

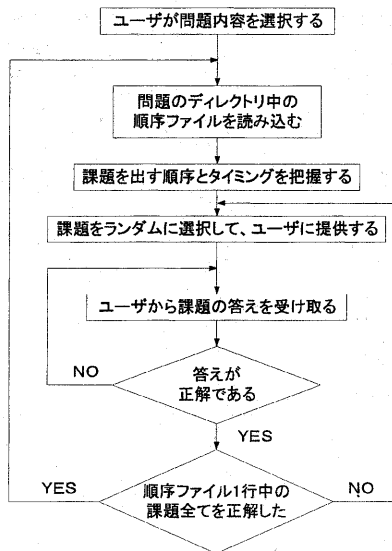


図5: 仮想ユーザのアルゴリズム

仮想ユーザは、これらの五つの機能を実現し、図5のアルゴリズムを守ることにより、問題の提供、正誤判断の仕組みを実現する。

5.3 仮想ハードウェアの全体構成

仮想ハードウェアは仮想OSに資源管理モジュールを提供し、その管理が行われる。具体的には、GUIコンポーネントや概念図を模したグラフィックスなどを使って、そのデータ構造を可視化した仮想主記憶装置と仮想補助記憶装置の資源管理モジュールで構成される。

仮想主記憶装置は、本来の主記憶装置を可視化したものであり、資源管理モジュール一つに対して一つのボタンが割り当てられている。ボタンが押されると、GUIインタフェースを表示する(図6参照)。具体的には、4.3.2の(2)、(3)、(4)の仕組みを持ち、学習者に資源管理操作をしてもらう。

仮想補助記憶装置は、本来の補助記憶装置を可視化したものであり、グラフィックスを使った同心円一つに対して一つの資源管理モジュールが割り当てられている。それが選択されると、GUIインタフェースを表示する(図6参照)。具体的には、4.3.2の(1)の仕組みを持ち、学習者に資源管理操作をし

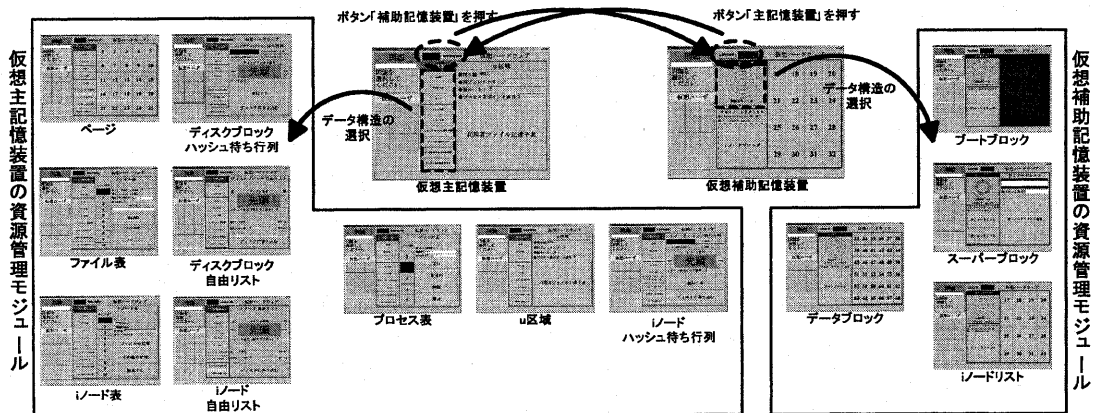


図 6: 仮想ハードウェアと資源管理モジュール

てもらう。

5.3.1 機能

仮想主記憶装置のアドレス構造中の移動や、仮想補助記憶装置のブロック構造中の移動はボタンの選択により行う。仮想ハードウェアの状態の設定には、GUI コンポーネントを使う。これらによって、設計方針の (2) で示した、簡略に操作のできる環境を実現できる。

5.3.2 可視化と実装

本来の OS が持つデータ構造の中で何を選択し、どのように可視化するかを次に述べる。

(1) ファイルシステム

このデータ構造は、ブートブロック、スーパーブロック、i ノードリスト、データブロックの四つを採用する。これらによって本システムは、電源起動時の仕組み、ファイルシステムのキャッシュの仕組み、i ノードとデータのブロックによる管理の仕組みを学習させる。

(2) バッファキャッシュ

このデータ構造は、自由リストとハッシュ待ち行列の二つを採用する。これらによって本システムは、バッファへのデータの割り当て、解放、ディスクへの書き込みのタイミングと、その重要性を学習

させる。

(3) プロセス

本来のプロセスは、レジスタやテキストなどの多くの情報を持っている。しかし本システムはプロトタイプとして、資源管理操作の内容をファイルシステムとバッファとの間のデータのやり取りに限定した。よってプロセスは、数値演算を行わない。そのため、CPU の仕組みは必要なく、例外やコンテキストの仕組みも必要ない。これらにより本システムのプロセスは、必要な情報をプロセスの状態に限定することができる。

データ構造は、u 区域とプロセス表を採用した。これらによって本システムは、プロセスの状態を管理する仕組みを学習させる。

(4) メモリ管理

メモリの管理方式は、要求時ページングを採用し、データ構造は、i ノード表、ファイル表を採用した。これらによって本システムは、ファイルに対して、open、close、read、write を実行できる。そして、ファイルへのデータ読み書き時のロックと参照数の関係の重要性とタイミングを学習させる。

6 実現

システムの実現について述べる。

6.1 実現

今回、実現したことを次に述べる。

本システムは、カーネルのできることは全て、学習者のできることに等しいと決めた。今回はプロトタイプとして、本来の OS が持つ仕組みの中で実現した資源管理内容は 4.3.2 で挙げた項目である。仮想ユーザの問題の提供、正誤判断の仕組みと、仮想ハードウェアのデータ構造を操作するための仕組みを実現し、一つの体験学習環境を実現した。

6.2 予備実験

本システムを使って予備実験を行い、現時点でのシステムの有効性を検証した。結果を次に示す。

実験規模は、OS に関する授業を受けた経験のある本研究室内の学部四年生 5 人を被験者に設定した。実験手順は、最初にシステムの仕様と使い方を口頭で説明し、その後試験者全員が同時にテストを開始する。二時間後、すぐにアンケートを記述してもらう。アンケートは、五段階評価による質問を五つ、自由記入欄による質問を三つ用意した。学習内容は、open、read、write、close などのファイル操作のシステムコールを中心に学習内容を構成し、七つの課題を用意した。

実験結果は、まず被験者はほぼ全員が一時間で一間以上の課題を解答することができた。そしてほぼ全員が、二時間後、最後の課題まで到達することができた。アンケートから、良い点は「GUI インタフェースが見やすい」、「試行錯誤の仕組みが達成感を与える」などの意見が挙げられ、悪い点は「最初、システムの使い方が分からなかった」、「学習後に結局何が学習できたのかははっきりしない」などの意見が挙げられていた。考察すると、「体験の仕組み」については好評であるが、「学習の仕組み」については学習者側への配慮が足りないということが分かった。これは、今回のシステムが、資源管理モジュールの仕組みを可視化することに重点を置きすぎて、

それを使う学習者への配慮が足りなかったことが原因である。今後の課題は、資源管理モジュールが増えていく中で、学習のしやすさを高めるようにシステムを拡張することが必要である。

6.3 実現規模

本システムは Windows 上で、Java 言語を使って構築した。ソースコードの実現規模を表 1 に示す。

表 1: 構成ごとのソースコード行数

仮想ユーザ	350行
仮想主記憶装置	3500行
仮想補助記憶装置	1500行

6.4 操作例

学習者が仮想 OS としてできることは図 7 に示す。

7 おわりに

本報告は、体験学習モデルによる OS の学習システムの実現を行うことによって、学習者に OS の仕組みを学習できる環境を提供することについて述べた。これによって本システムは、従来の OS 学習において、問題のある部分を補うことができるということが分かった。今後の課題は、次の三点である。

- ・より忠実な OS の仕組みを組み込んだ学習環境の実現
- ・実際に学習者に使ってもらって、評価を取る
- ・学習者の操作の過程を評価する

参考文献

- [1] Maurice J.Bach : UNIX カーネルの設計, 共立出版株式会社, 1991
- [2] Joseph O'Neil : 独習 Java, 翔泳社, 1999
- [3] 結城 浩 : Java 言語で学ぶデザインパターン入門, ソフトバンクパブリッシング株式会社, 2001

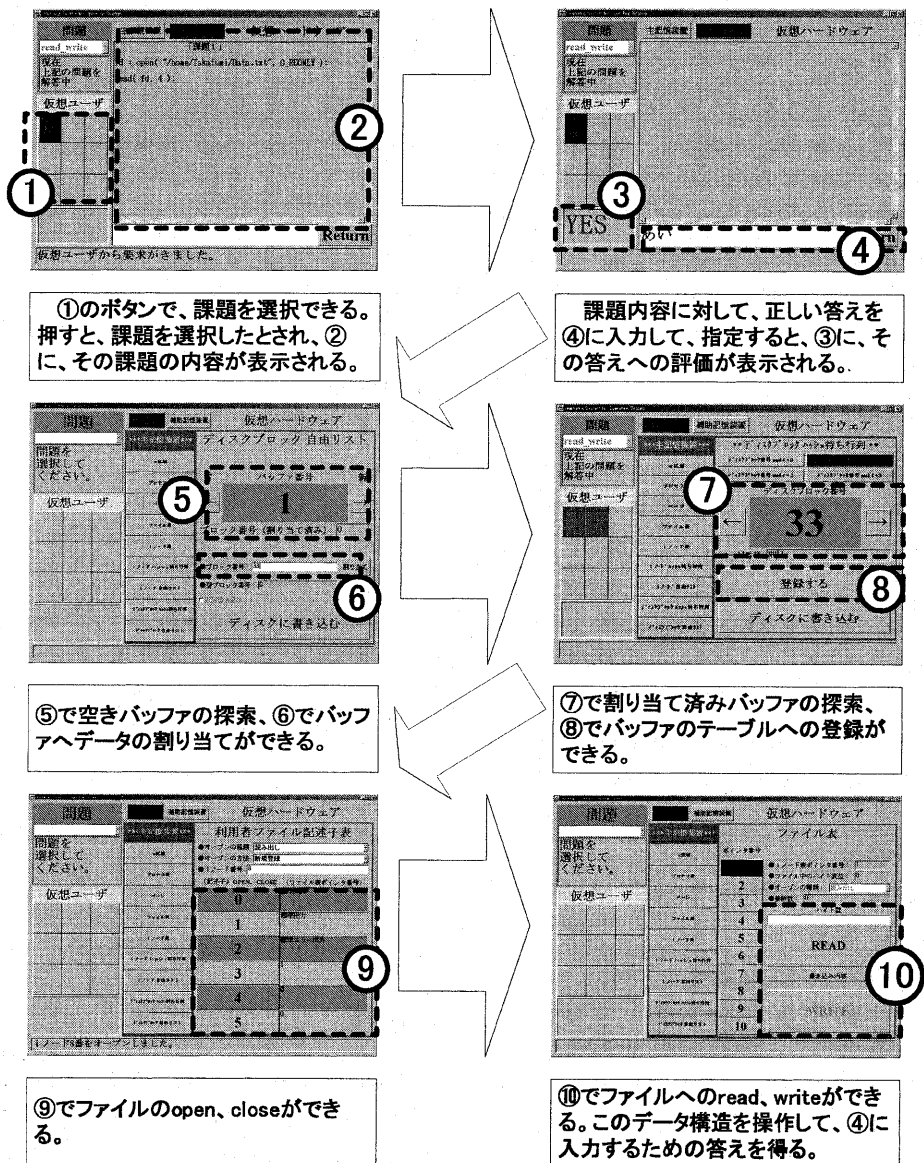


図 7: 操作例