

プログラミングの学習環境に関する研究 (既存の授業システムとの比較)

目次

1. はじめに
2. (再)構築したシステムについて
3. 学習効果の測定について
4. アンケートについて
5. 新システムの性能評価について
6. まとめ

1. はじめに

小畠一泰, 笠井聖二: WebAssembly を用いたブラウザ上で完結する C 言語の TDD 式学習システムの構築と提案, 令和3年度 呉工業高等専門学校 電気情報工学科 卒業研究報告会予稿集, 2022 より

高専・大学における工学系

→ C言語が必要言語の1つ

呉高専でも5年間プログラムを書く機会がある

しかし学生によって習得度合いの開きが感じられた

1. はじめに

小畠一泰, 笠井聖二: WebAssembly を用いたブラウザ上で完結する C 言語の TDD 式学習システムの構築と提案, 令和3年度 呉工業高等専門学校 電気情報工学科 卒業研究報告会予稿集, 2022 より

要因は何か？

- ・ 授業方法や実習方法に問題がある → 方法の問題
- ・ 授業時間や実習時間が少ない → 時間の問題
- ・ 授業時間外でする機会が少ない → 機会の問題
- ・ コードを写すだけで理解していない → 意欲, 方法の問題



授業や課題で,

“自分が何をしているか, 何を作っているのか想像がつかない”
という問題が生まれてしまっている

1. はじめに

小畠一泰, 笠井聖二: WebAssembly を用いたブラウザ上で完結する C 言語の TDD 式学習システムの構築と提案, 令和3年度 呉工業高等専門学校 電気情報工学科 卒業研究報告会予稿集, 2022 より

限られた授業時間内で
効率的 & 効果的
に学習できるシステムがあれば...

ブラウザ上で完結し、
軽量かつテスト可能な
学習環境を提案

1. はじめに

小畠一泰, 笠井聖二: WebAssembly を用いたブラウザ上で完結する C 言語の TDD 式学習システムの構築と提案, 令和3年度 呉工業高等専門学校 電気情報工学科 卒業研究報告会予稿集, 2022 より

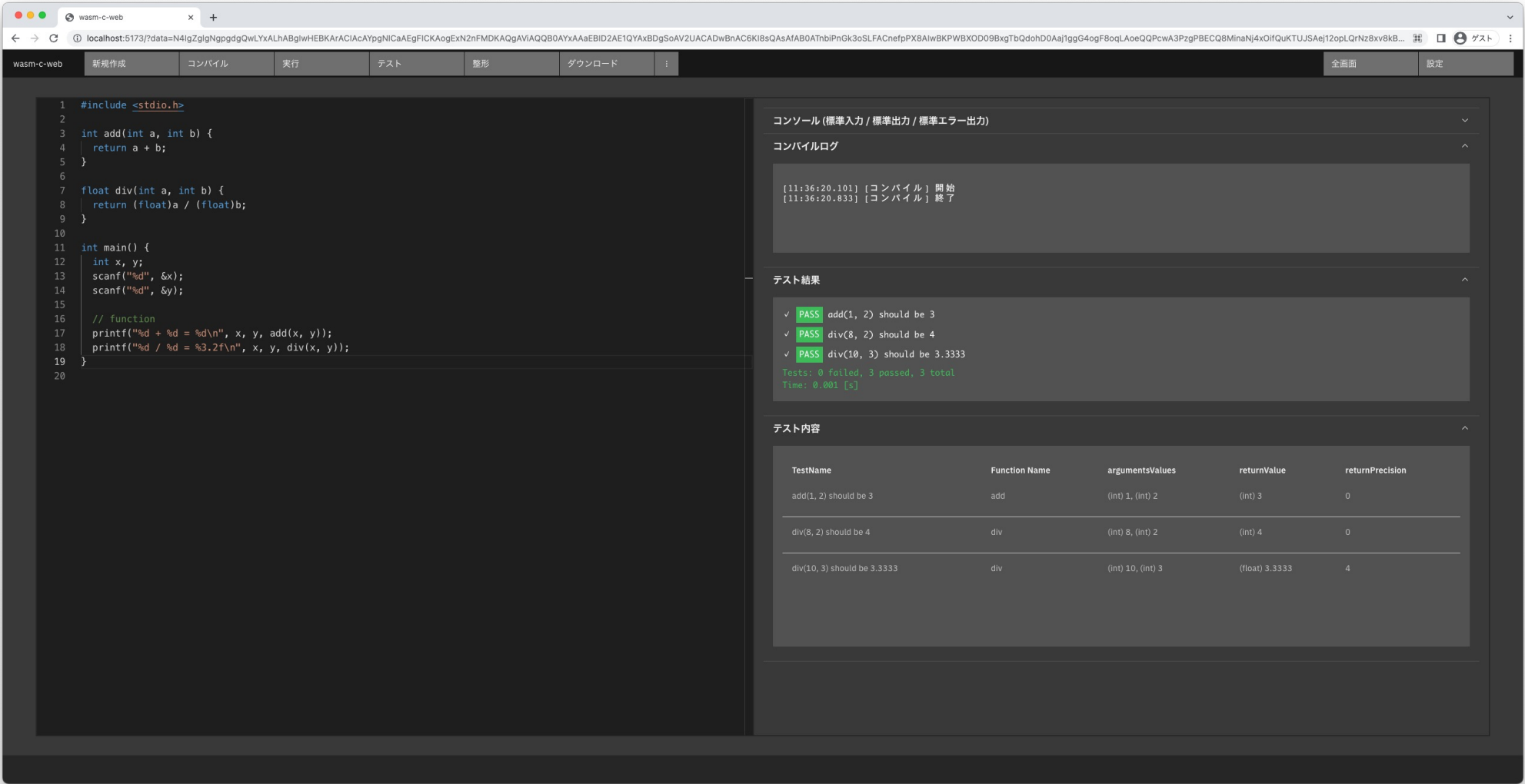
わかったこと

- 多くの学生が現在の授業方法に問題を感じている
- 構築したシステムでその問題は大きく改善されること

実際の学習効果について

- 授業などで使い, 使用状況や使い勝手などを調査する

2. (再)構築したシステムについて



2. (再)構築したシステムについて

旧システム

```
1  #include <stdio.h>
2
3  int add(int a, int b) {
4      return a + b;
5  }
```

テスト結果

✓ **PASS** add(1, 2) should be 3

Tests: 0 failed, 1 passed, 1 total

Time: 0.001 [s]

```
1  #include <stdio.h>
2
3  int add(int a, int b) {
4      return a - b;
5  }
```

テスト結果

× **FAIL** add(1, 2) should be 3

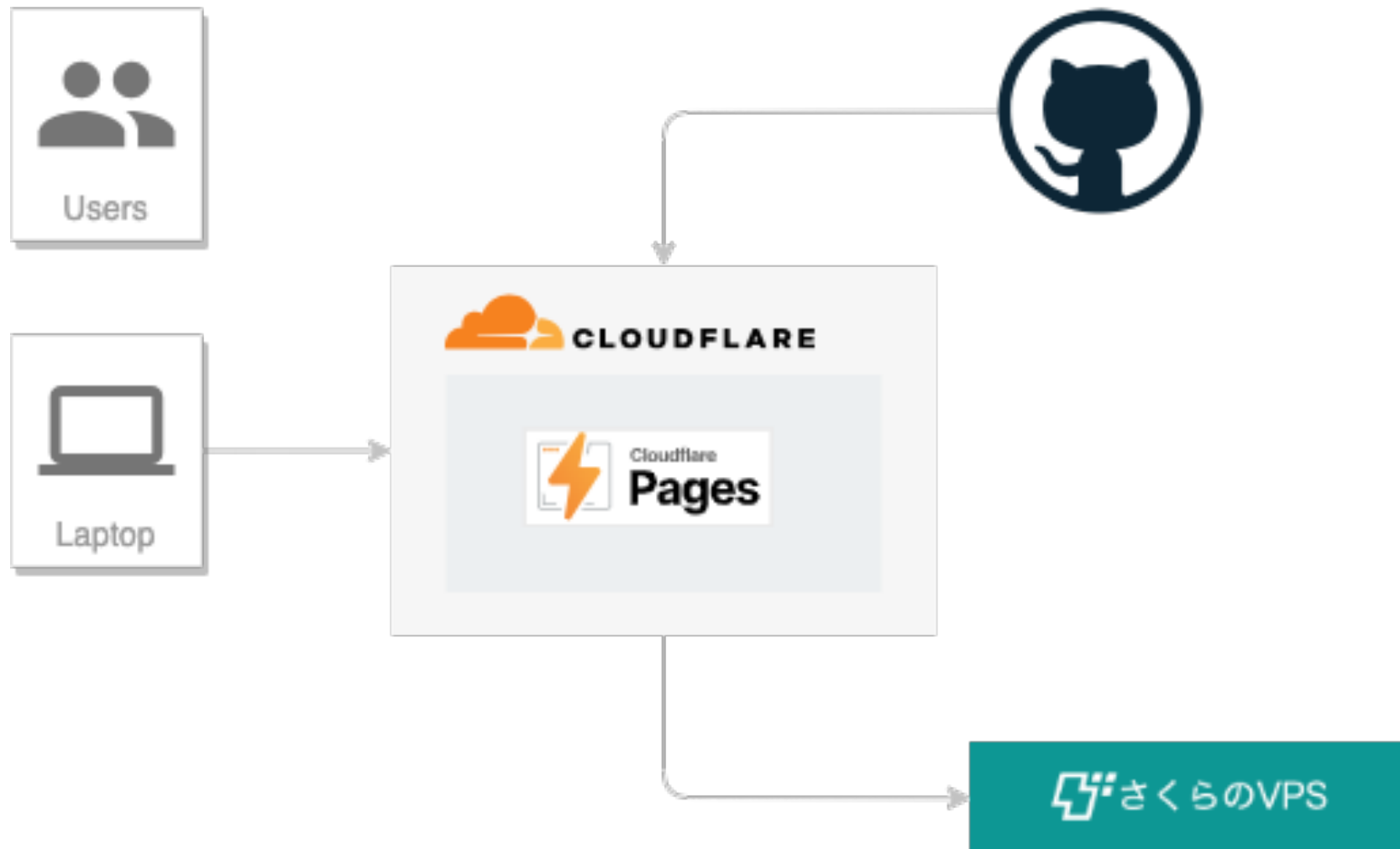
expected: 3, received: -1

Tests: 1 failed, 0 passed, 1 total

Time: 0.001 [s]

2. (再)構築したシステムについて

旧システム



2. (再)構築したシステムについて

旧システム

旧システムでは後に行う評価において
同時接続・同時利用に耐えられない



より効率的かつ大規模なアクセスにも耐えうるシステムへ

2. (再)構築したシステムについて

パターン1) オンプレミス

Nginxをリバースプロキシ&ロードバランサーとして利用

```
upstream app {  
    server 192.168.1.1 weight=2; # compiler + nginx  
    server 192.168.1.2 weight=1; # compiler + lsp  
}
```

- 学内ネットワークに依存するのでトラブル時に対応が難しい
- スケールしない
- メンテナンスが大変

The NGINX logo is displayed in a bold, green, sans-serif font. The letters are stylized, with the 'N' and 'G' having a unique geometric design.

2. (再)構築したシステムについて

パターン2) AWS

CaaSとFaaSで構築

- lspをECS(Fargate)で実装
- compilerをlambdaで実装



結果

lambdaでコンパイルできなかった

2. (再)構築したシステムについて

パターン3) GCP

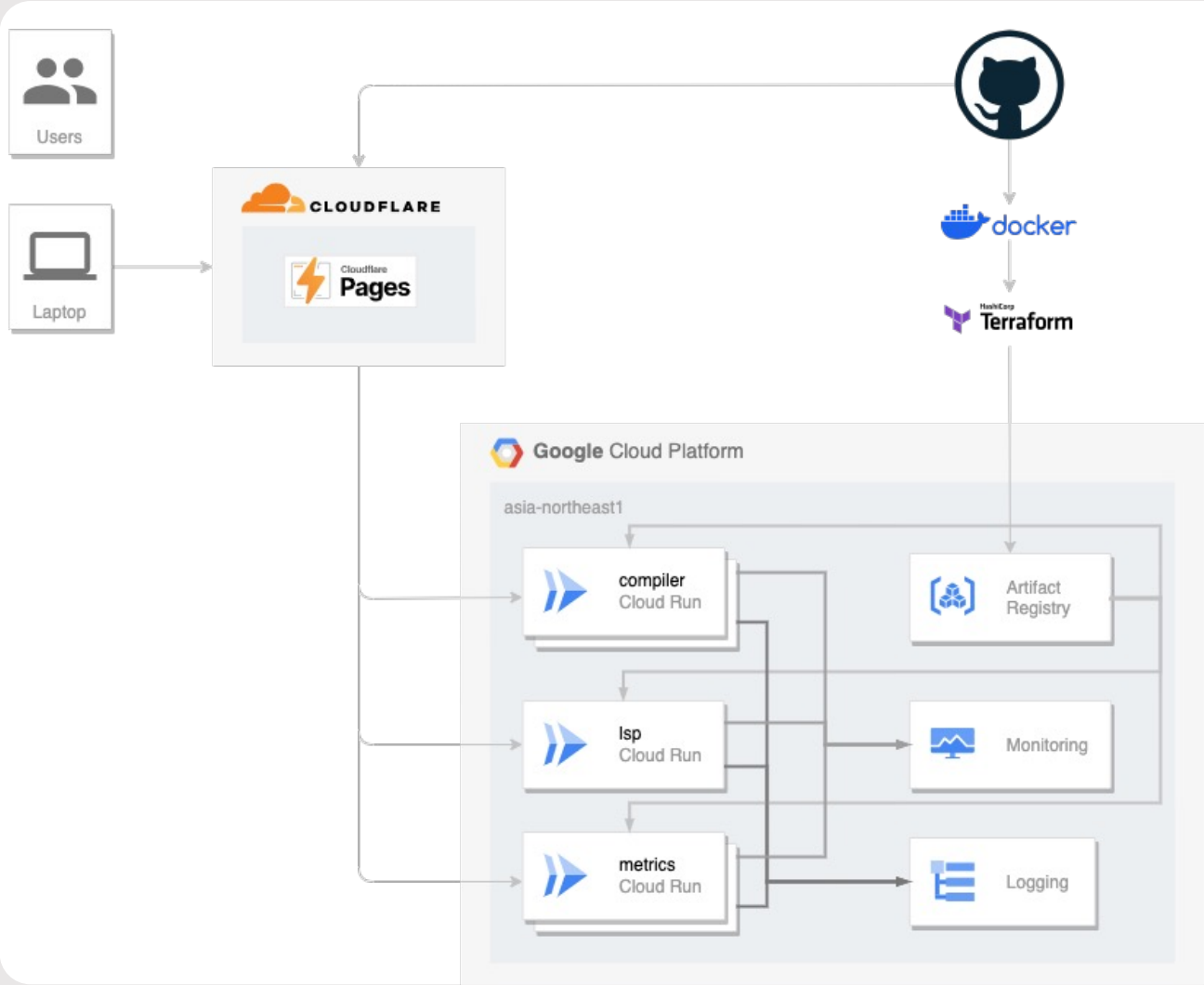


表 1 Cloud Run の性能

サーバー	CPU コア	メモリ	オートスケール
compiler	2	1GiB	0～10
lsp	4	4GiB	0～1
metrics	2	1GiB	0～2

3. 学習効果の測定について

被験者について

呉工業高等専門学校電気情報工学科2年に在籍する学生43名

表2 C言語を使い始めてどのくらい経ちますか？			表3 C言語以外を含むプログラミングには 授業時間以外で普段どのくらい触れていますか？			表4 プログラミングは得意ですか？		
選択肢	人数	割合	選択肢	人数	割合	選択肢	人数	割合
1年未満	2	5.9%	毎日触れている	0	0.0%	得意	1	2.9%
1年以上2年未満	29	85.3%	週に5～6日ほど触れている	1	2.9%	やや得意	5	14.7%
2年以上3年未満	3	8.8%	週に3～5日ほど触れている	1	2.9%	普通	12	35.3%
3年以上5年未満	0	0.0%	週に1～2日ほど触れている	16	47.1%	やや不得意	9	26.5%
5年以上	0	0.0%	月に1日程度触れている	4	11.8%	不得意	7	20.6%
			ほとんど触れない	10	29.4%			
			触れない	2	5.9%			

3. 学習効果の測定について

※ 被験者自身のパソコンを使用

調査の流れ

1. 各被験者に，符号を割り当てた
2. キー入力数が保存されるようにした
3. 授業システムではコンパイル・実行の際に，コンパイル時間などが保存されるようにした
4. 新システムでもコンパイル時間などが保存されるようにした
5. 集計・アンケート・調査を行なった

3. 学習効果の測定について

表 5 実施時間

項目	開始～終了	時間
研究説明・測定準備	08:50～09:45	55 分間
既存の環境での演習	09:45～10:00	15 分間
システムの詳しい説明	10:00～10:08	8 分間
システムでの演習	10:08～10:18	10 分間

3. 学習効果の測定について

付録1 演習1.c

```
#include <stdio.h>
```

```
/* 注意
```

- 関数名は変更しないでください。
- main 関数の中身は自由に変更して構いません。

```
*/
```

```
int add(int a, int b) {
```

```
// 引数を足し合わせた値を返す関数を作ってください。
```

```
}
```

```
float div(int a, int b) {  
// 引数 a を引数 b で割った値(型は float)を  
返す関数を作ってください。
```

```
}
```

```
int total(int max) {
```

```
// 0 から max まで(max を含む)を足し合わ  
せた値を返す関数を作ってください。
```

```
}
```

```
int main() {}
```

3. 学習効果の測定について

既存の環境について

2021
メモ帳
Cygwin

2022
GitHub Codespaces



The screenshot displays the GitHub Codespaces interface. On the left, the 'EXPLORER' sidebar shows a file tree with folders like '.github', '.vscode', 'app', and 'models', and files like 'emotion.rb' and 'interaction.rb'. The main editor area shows the 'devcontainer.json' file with Ruby code for setting up the environment. The bottom panel contains a 'TERMINAL' window showing logs for starting and finishing various extensions and the Monaco editor.

```
monaco-github-github-g59jq2w5w7.github.dev
```

EXPLORER

- GITHUB
- .devcontainer
 - build-devcontainer.sh
 - devcontainer.json
 - on-create-command.sh
 - post-attach-command.sh
 - setup-devcontainer.sh
 - ssrh
 - welcome_message.txt
- .github
- .vscode
- app
 - api
 - assets
 - components
 - controllers
 - helpers
 - jobs
 - mailers
 - models
 - archived.rb
 - emotion.rb
 - interaction.rb

emotion.rb x {} devcontainer.json

```
73 attr_reader :label
74
75 attr_reader :pronounceable_label
76
77 # Public: Get the Emoji that this reaction's content represents.
78 #
79 # Returns an Emoji.
80 attr_reader :emoji_character
81
82 def initialize(content:, label: nil, pronounceable_label: nil, emoji_character: nil)
83   @content = content
84   @label = label || @content
85   @pronounceable_label = pronounceable_label || @label
86   @emoji_character = emoji_character || Emoji.find_by_alias(@content)
87   @platform_enum = @pronounceable_label.gsub(" ", "_").upcase
88 end
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: zsh

```
[09:43:36] Starting 'watch-extension:vscode-api-tests'...
[09:43:36] Finished 'clean-extension:typescript-language-features' after 248 ms
[09:43:36] Starting 'watch-extension:typescript-language-features'...
[09:43:36] Finished 'clean-extension:php-language-features' after 384 ms
[09:43:36] Starting 'watch-extension:php-language-features'...
[09:43:40] Finished 'clean-extension:html-language-features-server' after 4.66 s
[09:43:40] Starting 'watch-extension:html-language-features-server'...
[09:43:43] Finished 'clean-client' after 7.33 s
[09:43:43] Starting 'watch-client'...
[09:44:50] [monaco.d.ts] Starting monaco.d.ts generation
[09:44:56] [monaco.d.ts] Finished monaco.d.ts generation
```

Codespaces main 0 0 Ln 29, Col 7 Tab Size: 4 UTF-8 LF Ruby

3. 学習効果の測定について

新システムについて

表 6 テスト内容

関数呼び出し	返値	精度
add(1, 3)	4	0
add(-5, -3)	-8	0
div(10, 3)	3.333	3

3. 学習効果の測定について

キー入力数の変化について

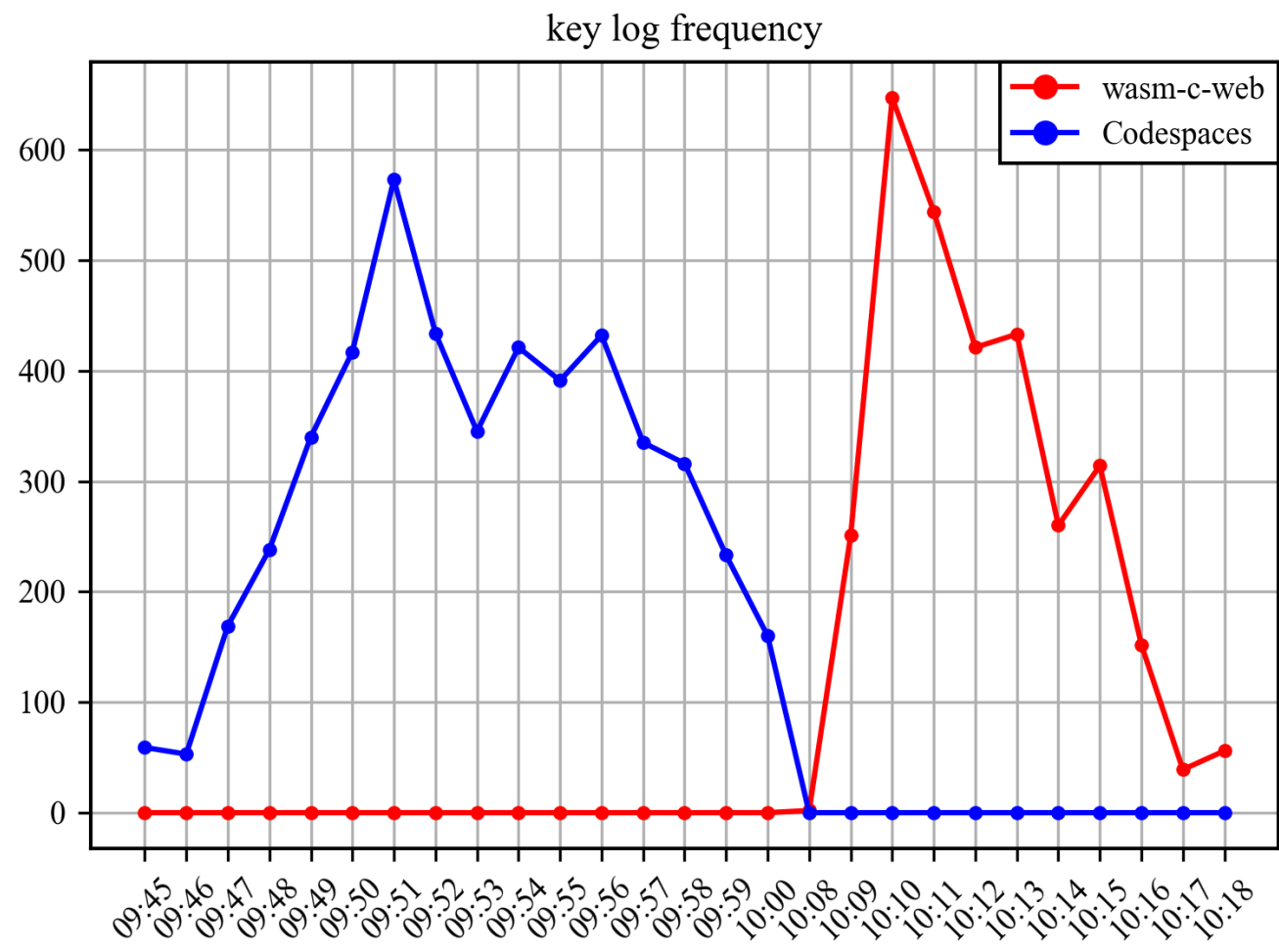
表 7 キー入力数の変化

	授業システム	新システム
サンプル数	26	26
最小値	0	0
最大値	592	321
平均値	189.1	120.0
中央値	182	98
最頻値	0, 228	93
上位 3 つ	592, 478, 288	321, 292, 235
下位 3 つ	0, 0, 1	0, 2, 8

3. 学習効果の測定について

キー入力数の変化について

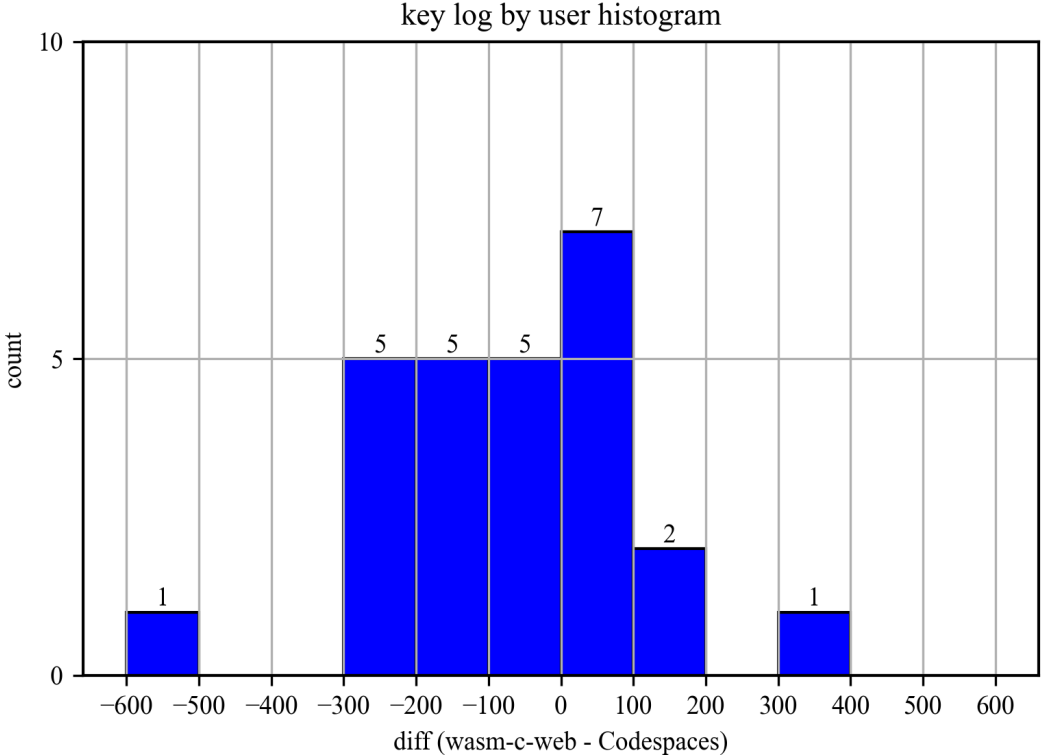
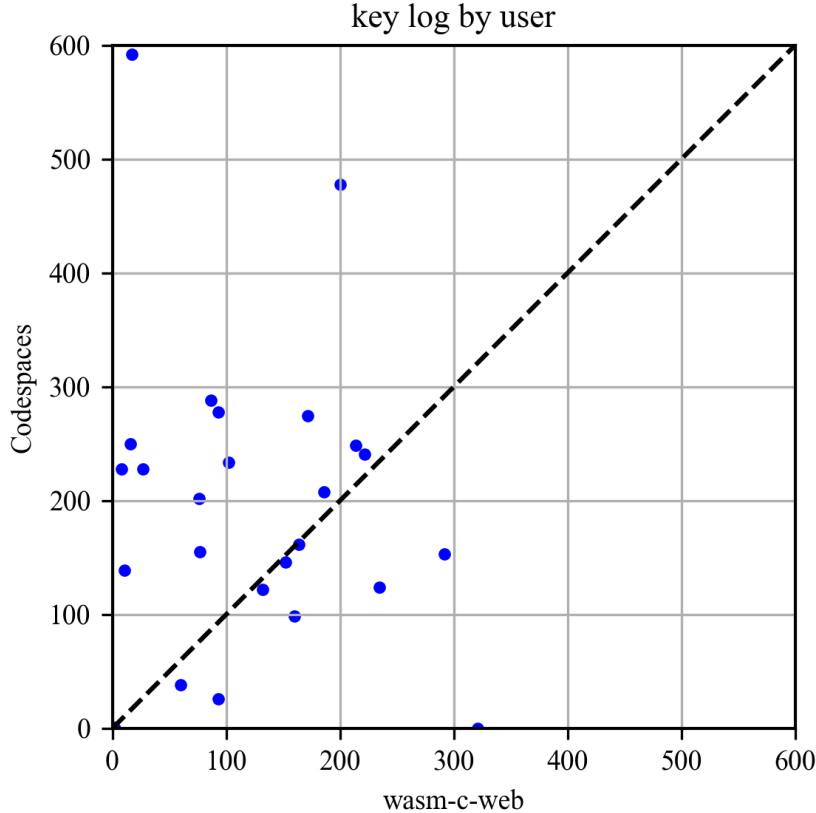
N=26,26



3. 学習効果の測定について

キー入力数の変化について

N=26,26



3. 学習効果の測定について

コンパイル数の変化について

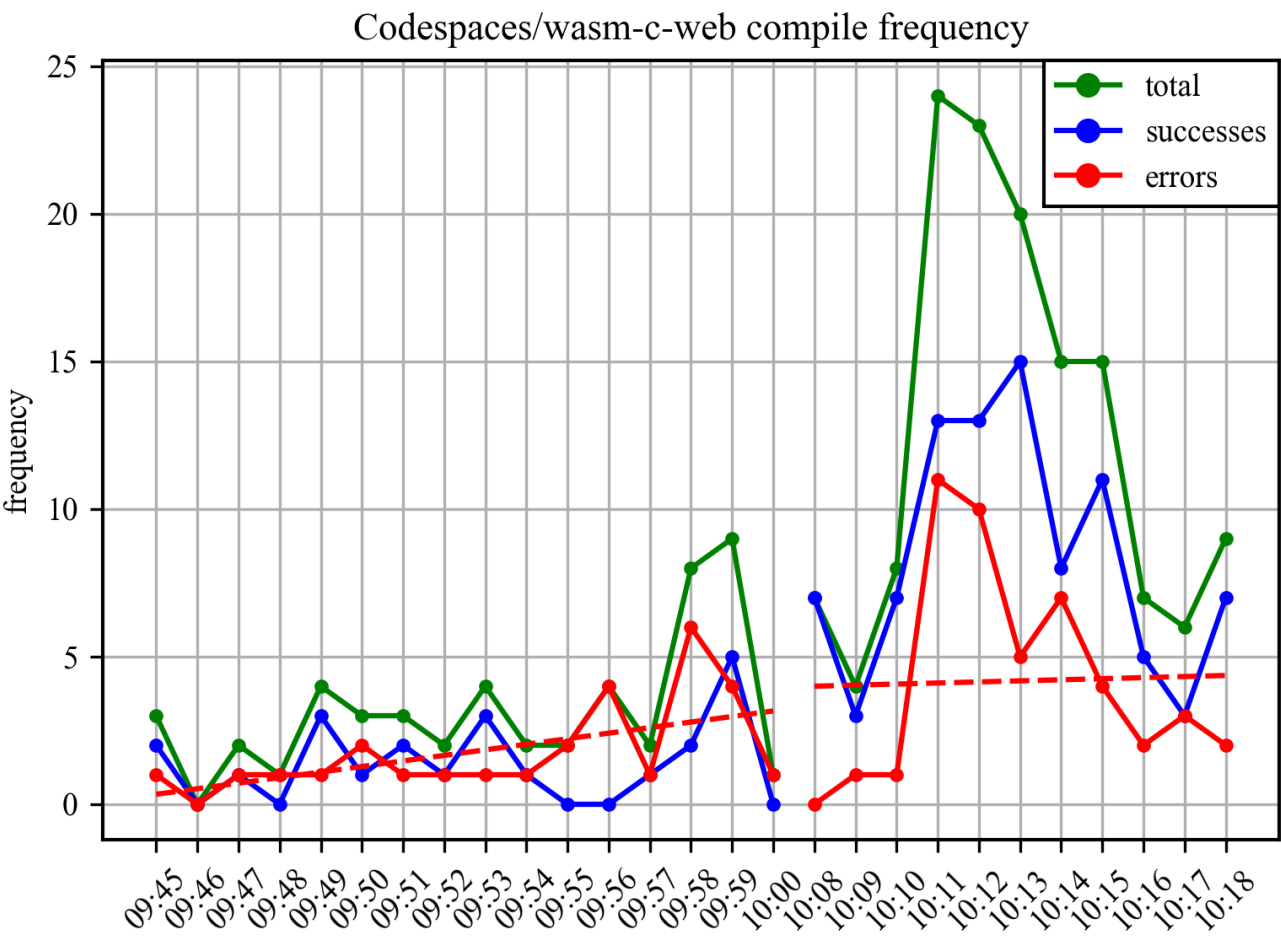
表 8 コンパイル数の変化

	授業システム	新システム
サンプル数	10	39
最小値	1	1
最大値	14	9
平均値	5.0	3.5
中央値	3	3
最頻値	2	3
上位 3 つ	14, 11, 7	9, 8, 8
下位 3 つ	1, 2, 2	1, 1, 1

3. 学習効果の測定について

コンパイル数の変化について

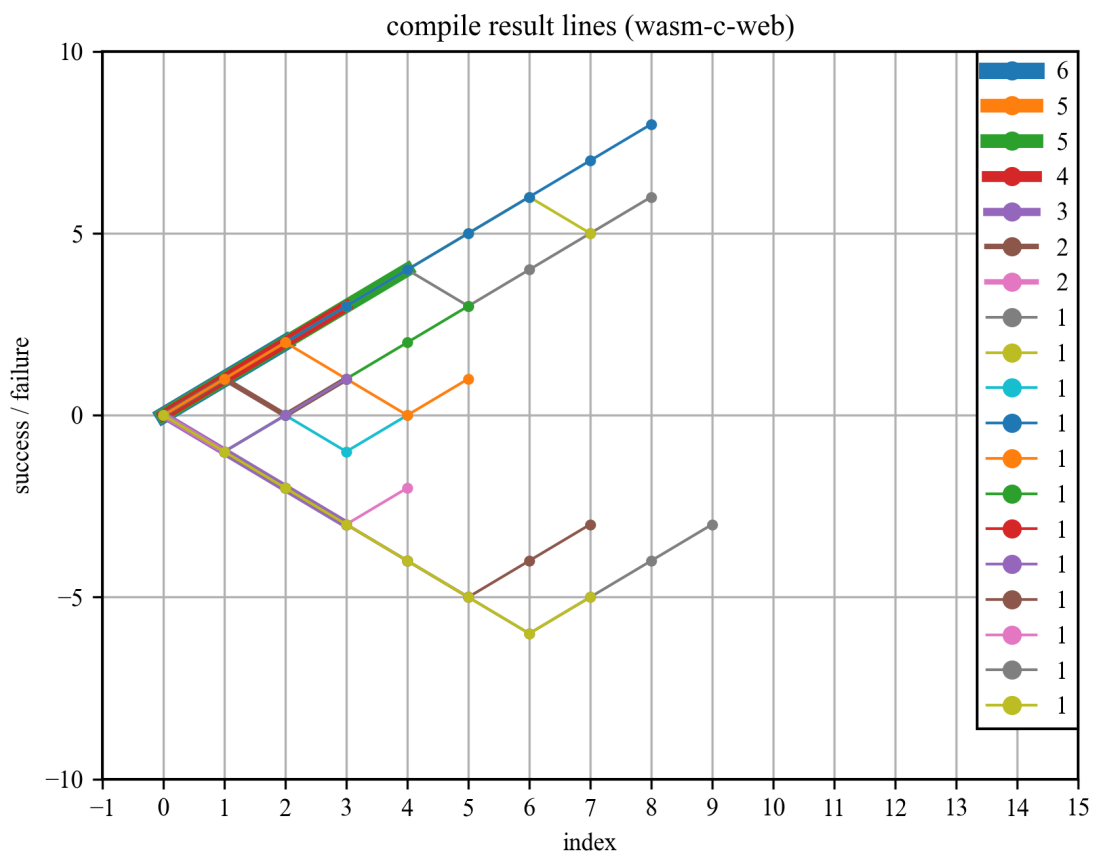
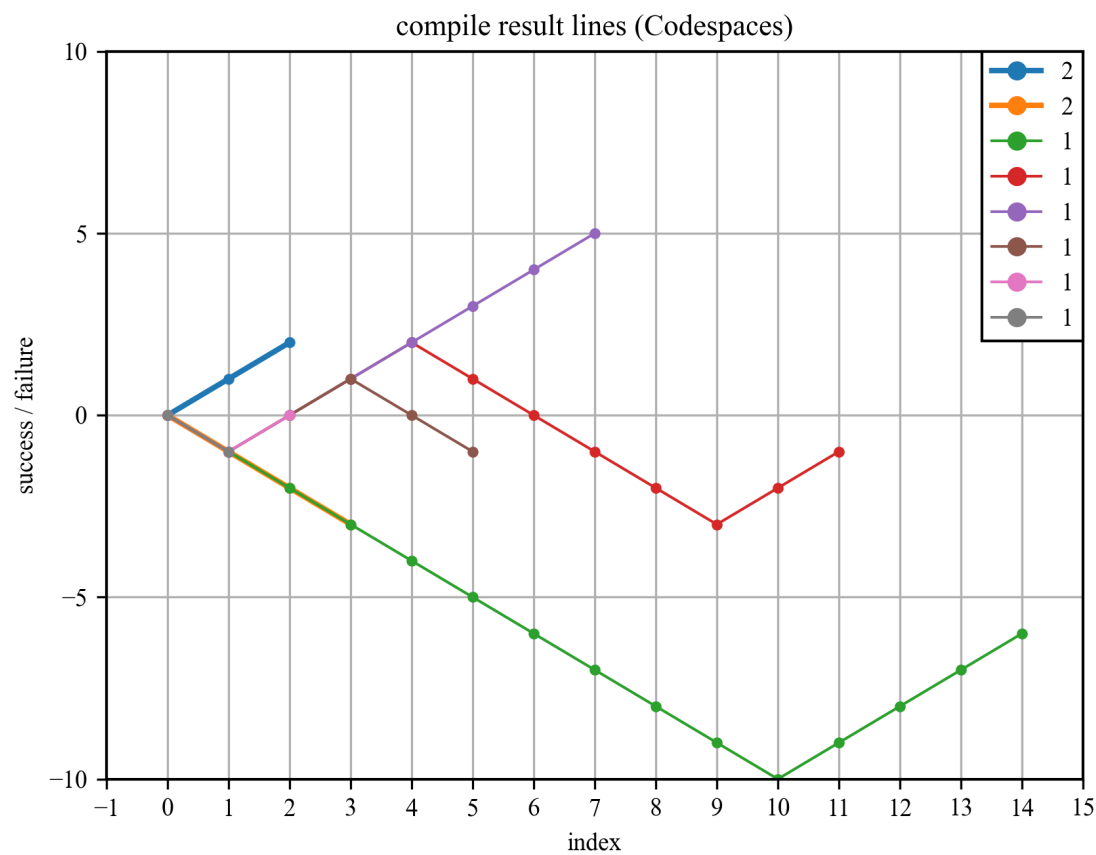
N=10,39



3. 学習効果の測定について

コンパイル数の変化について

N=10,39



4. アンケートについて

通常の授業の良いところ/悪いところについて

N=34

良いところ

- 全て手打ちだったので構文などを覚えられた：21票
- 様々なソフトに触れられたコンパイルコマンドなどに詳しくなれた：14票

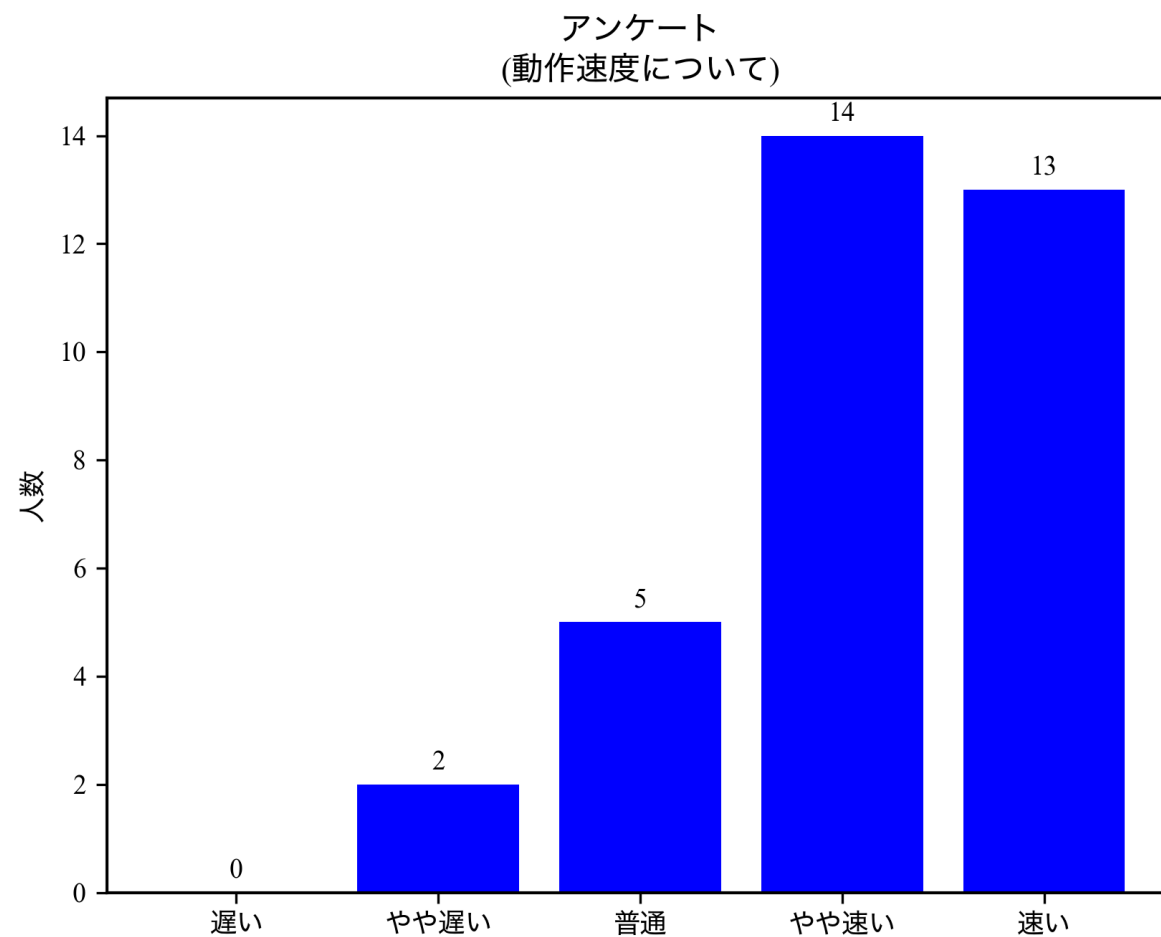
悪いところ

- 全て手打ちなので時間がかかってしまう：14票
- 動作が正しいか確認するのが大変：13票

4. アンケートについて

新システムの動作速度について

N=34



4. アンケートについて

新システムの良いところ/悪いところについて

N=34

良いところ

- テスト : 25票
- コンパイル : 17票
- 実行 : 16票
- 整形(ボタンをクリックするとソースコードが自動的に整えられる機能) : 10票
- 自動保存(500ms毎に自動的にブラウザ上に保存される機能) : 6票

悪いところ

- コンパイル, テスト, 整形 : 2票,
- 実行, 補完 : 1票

4. アンケートについて

授業で現在の学習方法から新システムに変更された場合の影響について

N=34

動作面と操作面から効率について問うたところ

- 操作が手軽なので学習効率が上がる : 19票
- 動作が軽快なので学習効率が上がる : 13票
- 操作が難しいので学習効率が落ちる : 4票
- 動作が遅いので学習効率が落ちる : 2票

4. アンケートについて

本システム・本研究自体へ総括して意見や感想があれば教えてください

- テスト機能があって便利だった
- 個人的にはCodespacesよりも使いやすかった
- Codespacesより接続が安定していて、ストレスもなかった
- このシステムがあればプログラミングが劇的に捗ると思います。世界的に正式にこのシステムのサービスが開始されてほしいほどです。
- このシステムのように初学者にとって使いやすく、最小限必要な機能だけなわかりやすいシステムがもっと拡充されることを願います。

5. 新システムの性能評価について

付録1 演習1.c

```
#include <stdio.h>
int add(int a, int b) {}
float div(int a, int b) {}
int total(int max) {}
int main() {}
```

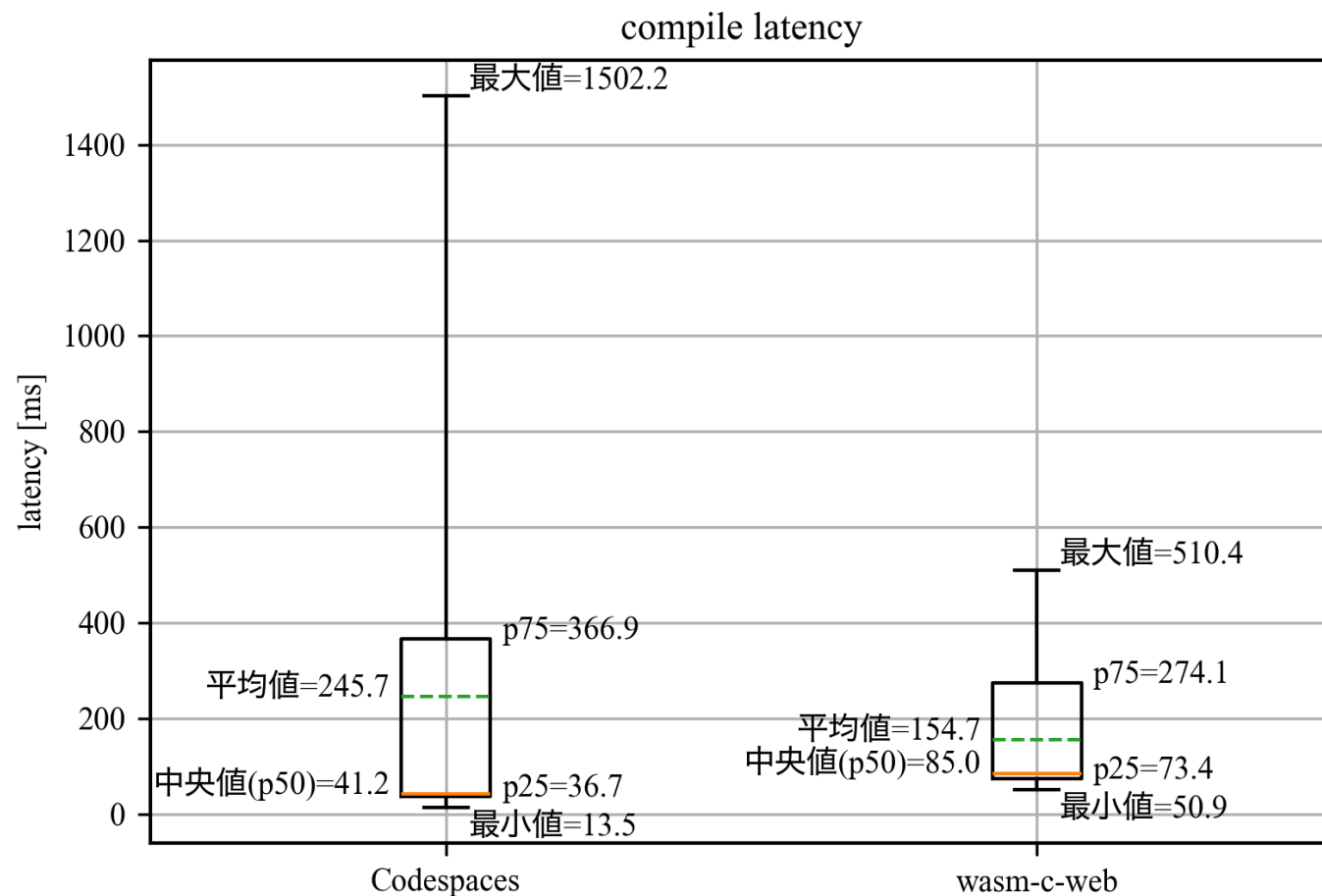
付録2 日本語.c

```
#include <stdio.h>
int main() {
    printf("あ¥n");
}
```

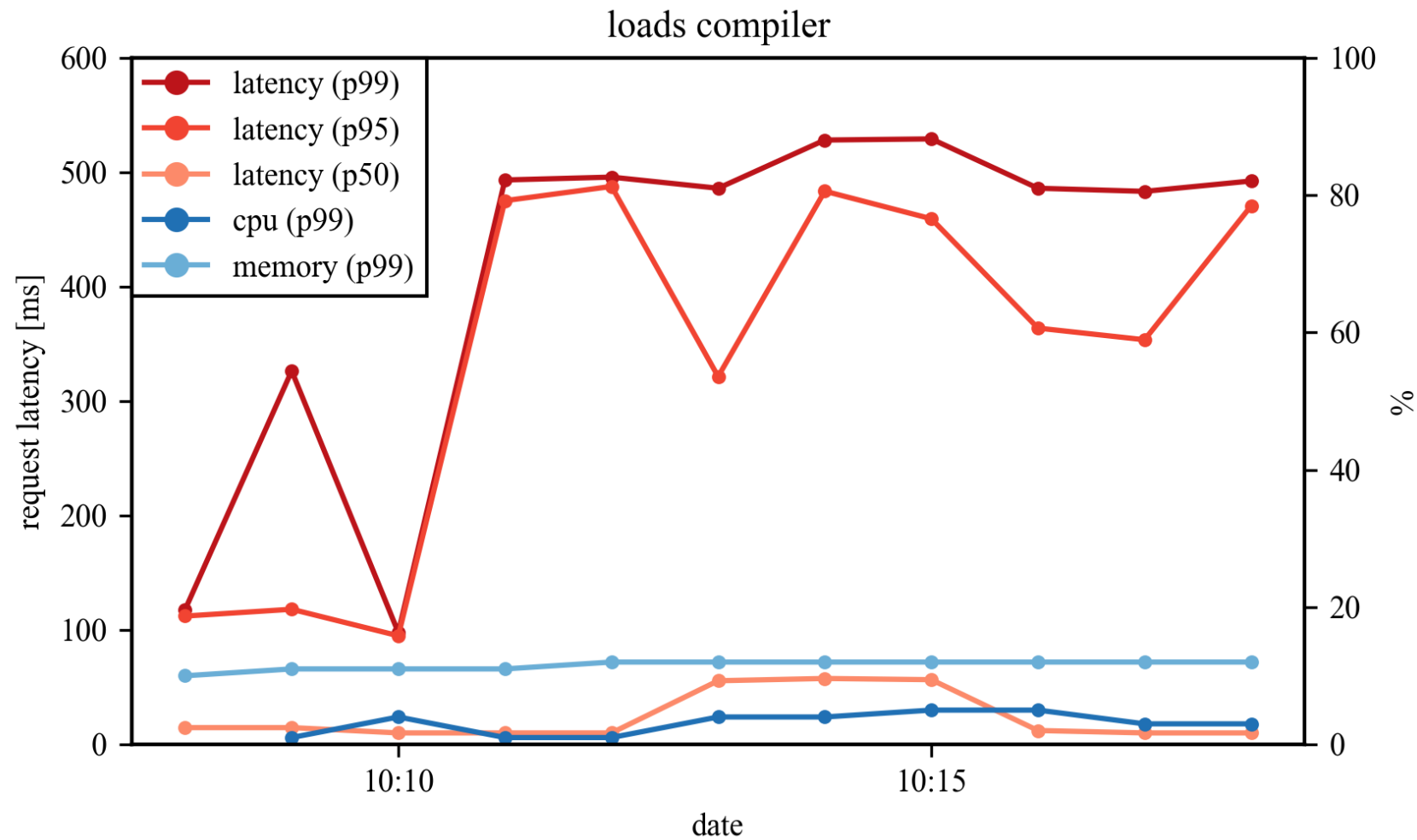
表 10 コンパイル後のファイルサイズ

対象	授業システム [KB]	新システム [KB]
付録 1	16.7	1.7
付録 2	33.1	61.2

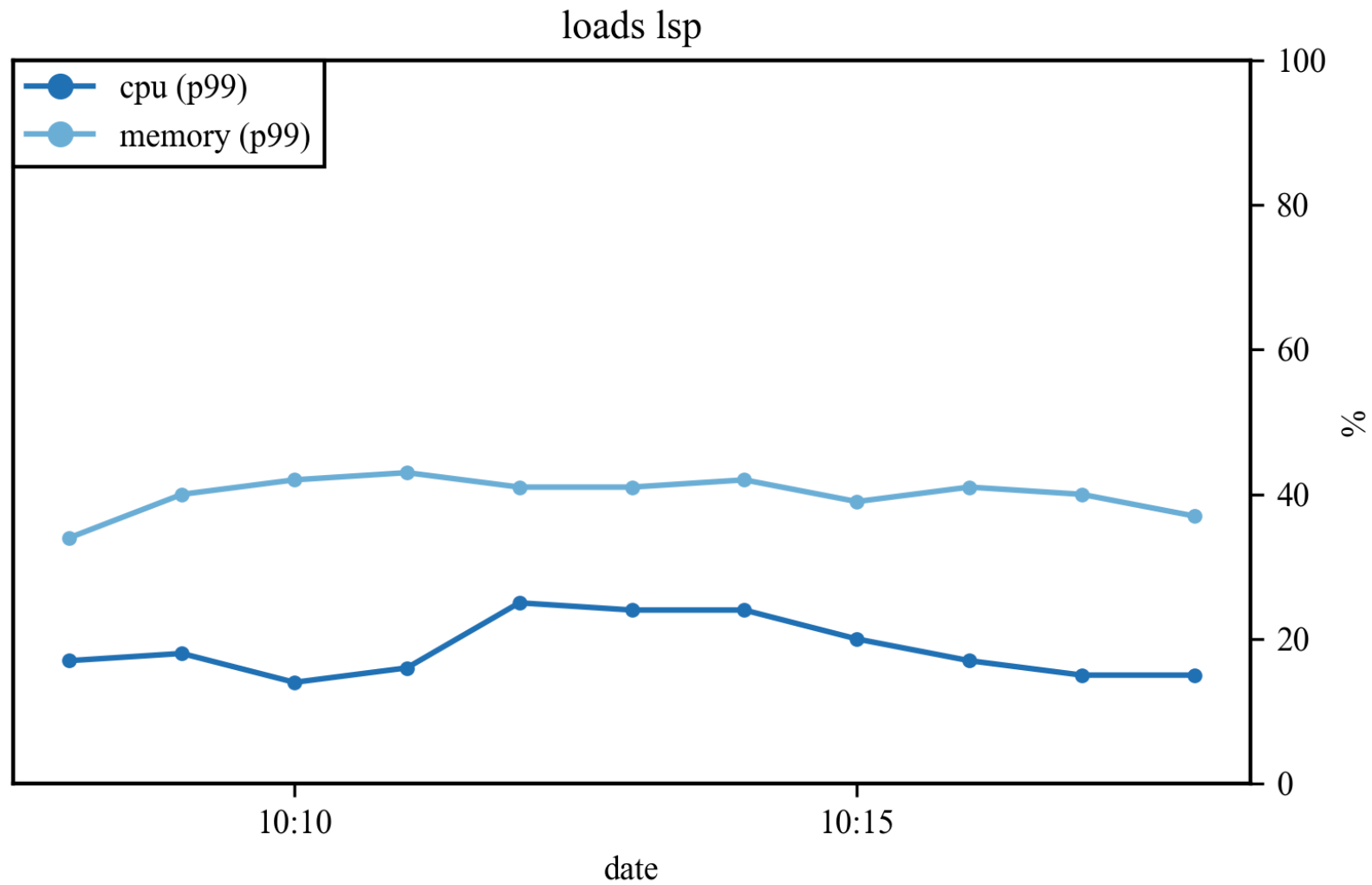
5. 新システムの性能評価について



5. 新システムの性能評価について



5. 新システムの性能評価について



6. まとめ

本研究では40名近い学生に対し、実際の授業と同様の環境で、統計調査並びにアンケートを実施することができた

調査の結果、次のことが明らかとなった

- 多くの学生が現在の授業方法に問題を感じていること
- 構築した新システムで大きく改善されること
- 学習効果があること
- 授業利用などにも耐えうるシステムであること