



# تطبيق ويب

(tt'byq wyb)

2012 年度 計算数学 II  
12班

保坂 和宏 ・ 若月 駿



# グループ・メンバー

- tt'byq wyb
  - Web アプリケーション
- 保坂 和宏 (数学科 B3)
- 若月 駿 (数学科 B3)

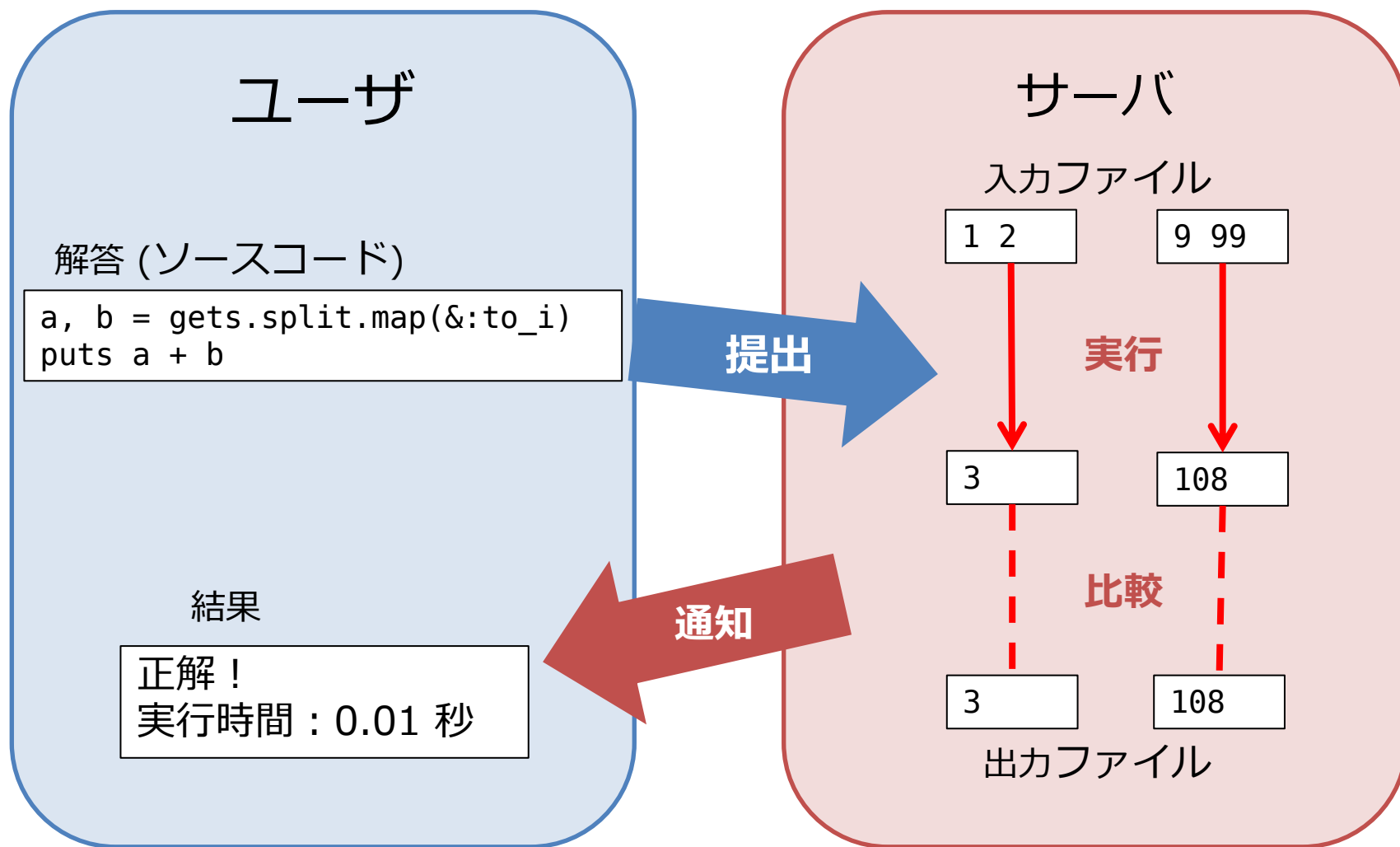
# テーマ・動機

- Web アプリケーションを学んでみたい！
  - ログインとかログアウトとか, どうなっているの??
- 題材として, プログラミングコンテストのジャッジサーバを作ってみよう, ということに
  - 受け取ったプログラムを走らせるには??

# プログラミングコンテスト？

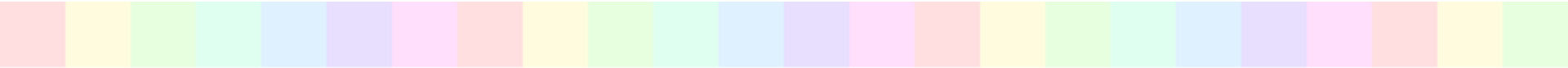
- 問題を解くプログラムを作成する
- 例：「整数  $a$  と  $b$  を入力として読み込んで、和  $a + b$  を出力せよ」

# プログラミングコンテスト？



# プログラミングコンテスト？

- 既存のサービスの例
  - PKU JudgeOnline (北京大学)
    - <http://poj.org/>
  - Aizu Online Judge (会津大学)
    - <http://judge.u-aizu.ac.jp/onlinejudge/>
- 出発点：Web アプリの知識ほぼゼロ
- 目標：ひとまず動くものを作る



# 作業・分担

- 学期前半：Web アプリの基礎の勉強
- 学期後半：ものづくり
  - 保坂：プログラム実行
  - 若月：ユーザインターフェース

# プログラムの実行

- ユーザから受け取ったソースコードを実行する
  - 今回は C++ と Ruby を対象
    - C++ は受け取ったソースコードをコンパイルする必要もあり



# プログラムの実行

## • 危険！！！！

- 無限ループするプログラムが送られたら？
- 途中で不正なメモリアクセスをするプログラムが送られたら？
- (所定の入出力以外の) ファイルを読み書きするプログラムが送られたら？
- その他，問題を解く上での不正行為，あるいはサーバへのハックがいろいろ考えられる

# プログラムの実行

- ファイルアクセスの防止
  - 提出を受け取るごとに, Linux のユーザを作成してしまう
    - 提出 ID でユニークに
  - そのユーザの読み書き権限を与えた実行用ディレクトリを作成 (後で削除)
  - 実行用ディレクトリ内で, そのユーザ名としてプログラムを実行
    - `sudo -u [ユーザ名]`
    - 実行用ディレクトリ外にはアクセスできない

# プログラムの実行

- 制限時間付き実行
  - timeout コマンドを使用
    - 指定した実行時間に達すると、実行中のコマンドにシグナル SIGKILL を送信して打ち切る、などができる

# プログラムの実行

- 実行結果の入手
  - 出力以外にもほしい情報がある
- 実行時間
  - time コマンドで測定
- 終了コード
  - 異常終了 (不正なメモリアクセスなどの実行時エラー) を検知できる
  - シェルの変数 \$? に代入される

# プログラムの実行

- sudo, timeout, time を適切に組み合わせつつ, 適切なタイミングで \$? を入手
  - シェルスクリプトの組合せ方がパズルのようでした
- Ruby からシェルコマンドを実行
  - 比較的低機能である sh が使われるので工夫が必要

# プログラムの実行

```
system("sudo -u [ユーザ名]  
  sh -c ¥"  
    echo ¥'  
      timeout [時間制限] [実行コマンド] < [入力] > [出力] 2> /dev/null;  
      echo ¥¥$? > [終了コード保存先]  
    ¥' > [実行用スクリプト]  
  ¥"  
")
```

```
system("sudo -u [ユーザ名]  
  sh -c ¥"  
    time -o [実行時間保存先] -f ¥¥¥"%S %U¥¥¥" sh [実行用スクリプト]  
  ¥"  
")
```

# 課題

- 実行用の Linux のユーザ作成のコストが重い？
  - 予めいくつかユーザを作っておいて使いまわせるとよさそう？
- セキュリティは大丈夫？
  - ファイルアクセス制限があるとはいえ、あらゆる攻撃に耐えきれているとは言い切れないと思います
  - ネットワークアクセスとか？

# 作ったもの

- ◆ プログラミングコンテストのシステム
  - ◆ ちなみに私はプログラミングコンテストはやったことがありません。
- ◆ 相方の保坂君と作業を分担しました。
  - ◆ 私がつくったのは「外側」の部分。
  - ◆ 「中核」となる、提出されたソースコードを実行する部分は保坂君が担当しました。



# 使ったもの

## ◆ Apache

- ◆ Webブラウザとshellとの間を仲介してくれるソフトウェア。初期設定さえ済ませておけば、決められたディレクトリにファイルを置いておくだけで、ブラウザからのアクセスに対応してそのファイルを実行してくれるすぐれもの。

## ◆ PostgreSQL

- ◆ データベース管理システム。名前、パスワードなどのユーザーデータを保存しておくために利用しました。

## ◆ Ruby

- ◆ さまざまな処理を記述するのに使用。

# 完成品の紹介

- ◆ サーバーを起動した状態にしておいたので、暇な人はアクセスしてみてください。
- ◆ ちゃんと動くものを作ることを優先したので、セキュリティについてはほとんど考えられていません。この中に悪意あるユーザーはいないと信じています。

# 完成品の紹介

## ◆ まずはログイン画面

Login - Mozilla Firefox

Login

localhost/2012ks2/ks/login\_page.cgi

Google ログイン 計算数学I/IIのページ [ThinkIT] 第2回:Apac... Login

### Login

username

password

Login

### Create new account

username

password

add

# 完成品の紹介

- ◆ ユーザー名とパスワードを入力してログインします。

Login - Mozilla Firefox

☐ Login

localhost/2012ks2/ks/login\_page.cgi

Google ログイン 計算数学I/IIのページ [ThinkIT] 第2回: Apac... Login

### Login

username

password

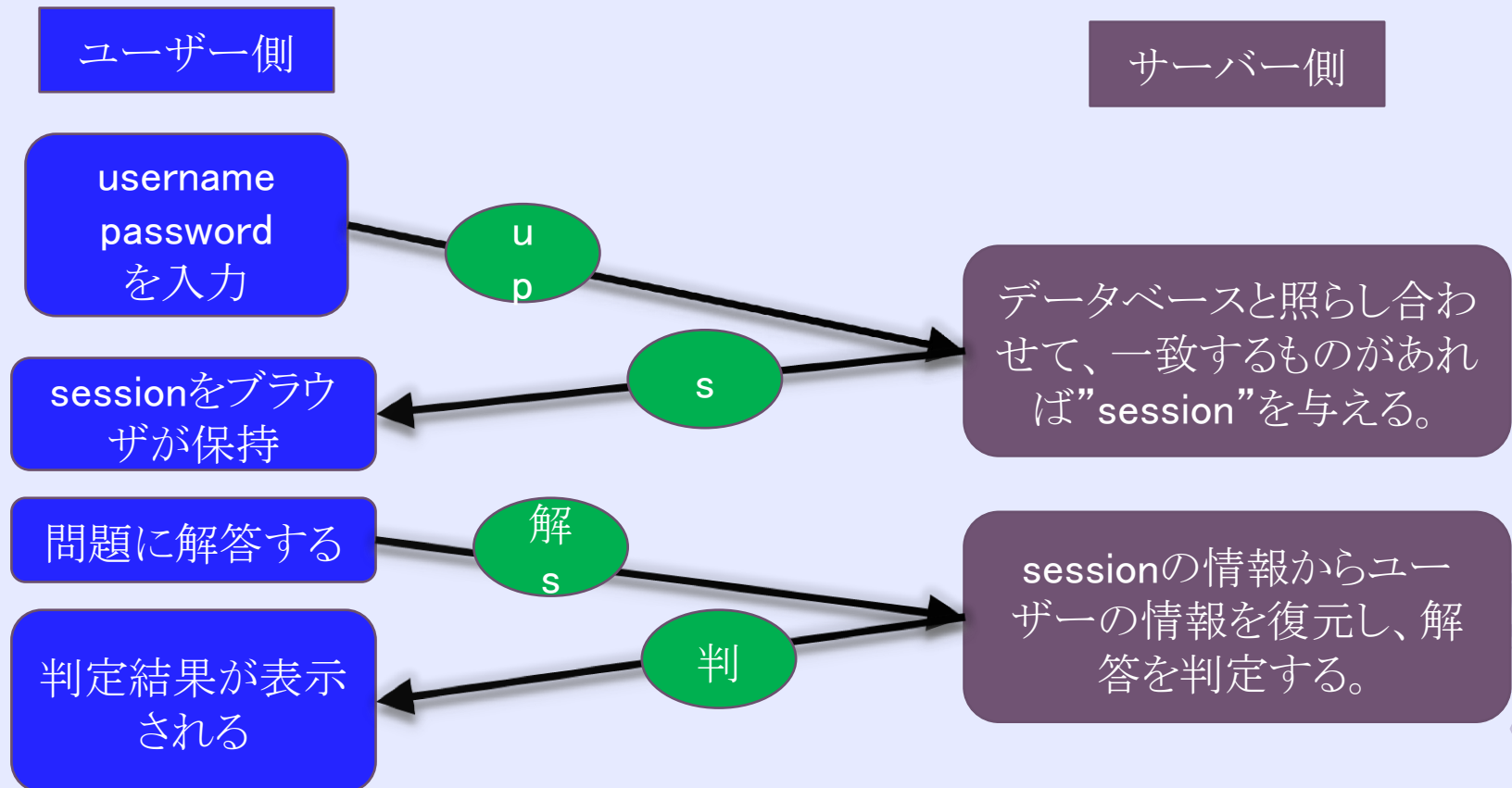
### Create new account

username

password

# 完成品の紹介

- ◆ このとき、サーバー側ではログイン処理を行います。



# 完成品の紹介

## ◆ ホーム画面

Home - Mozilla Firefox

Home

localhost/2012ks2/ks/home.cgi

Google ログイン 計算数学I/IIのページ [ThinkIT] 第2回: Apac... Login admin

id: 3, name: **wakatsuki**

prob list

prob_num	prob_title	prob_page	score					
1	Add two integers	go	<table border="1"><tr><td>date</td><td>verdict</td><td>execution_time</td><td>language</td><td>source_code</td></tr></table>	date	verdict	execution_time	language	source_code
date	verdict	execution_time	language	source_code				
2	prob2	go	<table border="1"><tr><td>date</td><td>verdict</td><td>execution_time</td><td>language</td><td>source_code</td></tr></table>	date	verdict	execution_time	language	source_code
date	verdict	execution_time	language	source_code				
3	prob3	go	<table border="1"><tr><td>date</td><td>verdict</td><td>execution_time</td><td>language</td><td>source_code</td></tr></table>	date	verdict	execution_time	language	source_code
date	verdict	execution_time	language	source_code				
4	prob4	go	<table border="1"><tr><td>date</td><td>verdict</td><td>execution_time</td><td>language</td><td>source_code</td></tr></table>	date	verdict	execution_time	language	source_code
date	verdict	execution_time	language	source_code				
5	prob5	go	<table border="1"><tr><td>date</td><td>verdict</td><td>execution_time</td><td>language</td><td>source_code</td></tr></table>	date	verdict	execution_time	language	source_code
date	verdict	execution_time	language	source_code				

[Change password](#) [LogOut](#)

# 完成品の紹介

- ◆ 大きく見えている表は、現在解答可能な問題のリストです。”go”から問題のページに進めます。
- ◆ ちなみに、下の方のリンクから、パスワードの変更とログアウトができます。

Home - Mozilla Firefox

Home

localhost/2012ks2/ks/home.cgi

Google ログイン 計算数学I/IIのページ [Think

id: 3, name: **wakatsuki**

prob list

prob_num	prob_title	prob_page	score
1	Add two integers	go	date v
2	prob2	go	date v
3	prob3	go	date v
4	prob4	go	date v
5	prob5	go	date v

[Change password](#) [LogOut](#)

# 完成品の紹介

- ◆ 問題のページ。
  - ◆ 一番上に問題文が表示されています。
  - ◆ 言語を**C++**か**Ruby**から選択したのちに、下のテキストフォームにソースコードを入力して解答を提出します。

prob\_page - Mozilla Firefox

prob\_page

localhost/2012ks2/ks/prob\_page.cgi

Google ログイン 計算数学I/IIのページ [ThinkIT] 第

### Problem1

Input two integers, and return the sum of them.

test  
test  
test

### Answer

ProblemNumber  
1

Language  
Ruby

SourceCode

```
line = STDIN.gets
sum = 0
line.split.each do |a|
  sum += a.to_i
end
print sum
```

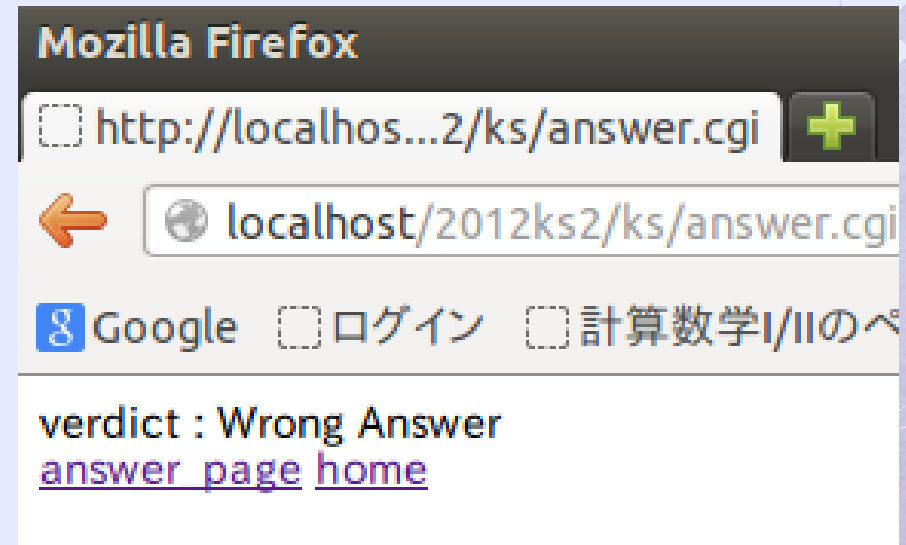
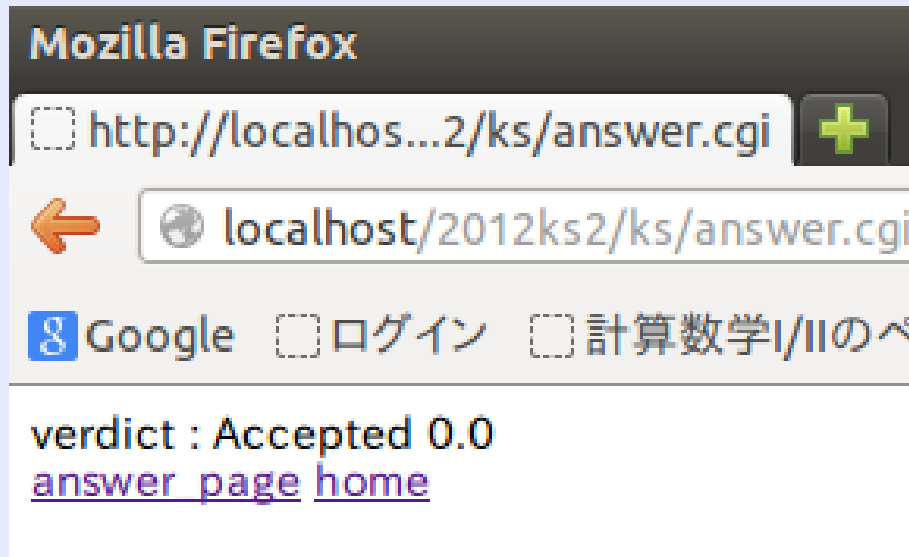
submit

[Back to home.](#)



# 完成品の紹介

- ◆ 解答を提出すると、サーバー側で解答が正しいかどうか判定し、それに応じて以下のような結果が返ってきます。



# 完成品の紹介

- ◆ ホーム画面に戻るとちゃんと解答の結果が反映されています。

Home - Mozilla Firefox

Home

localhost/2012ks2/ks/home.cgi

Google ログイン 計算数学I/IIのページ [ThinkIT] 第2回:Apac... Login admin

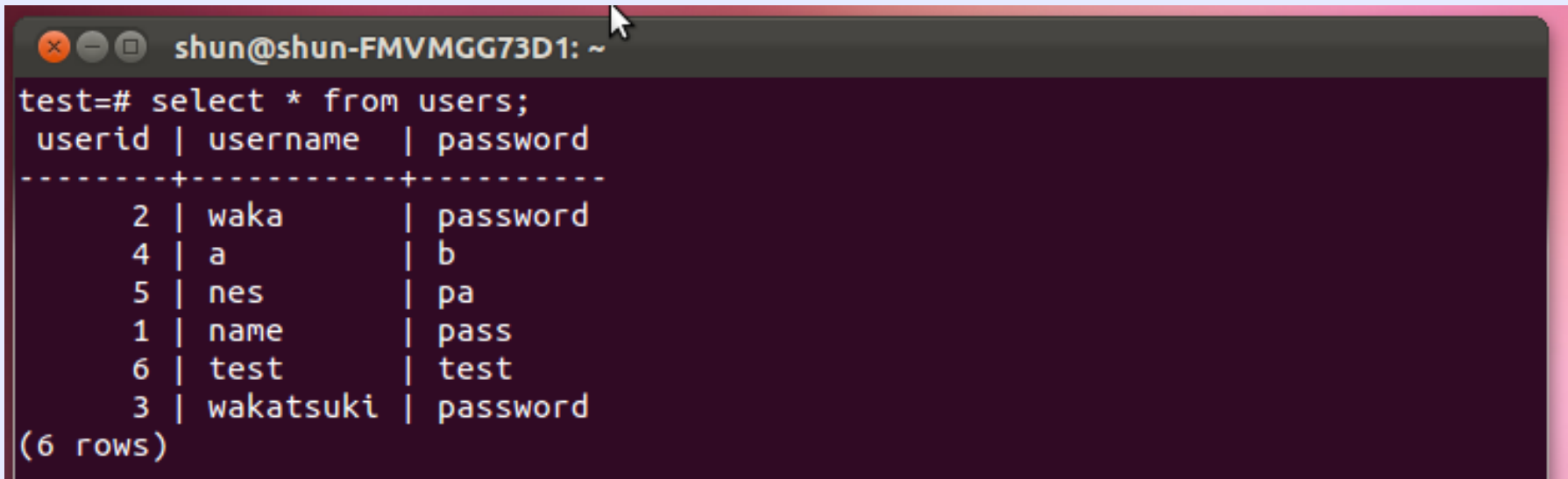
id: 3, name: **wakatsuki**

prob list

prob_num	prob_title	prob_page	score				
1	Add two integers	go	date	verdict	execution_time	language	source_code
			01/23/13 23:55:34	Accepted 0.0	0.0	Ruby	line = STDIN.gets sun
			01/23/13 23:56:25	Wrong Answer		Ruby	print 100
2	prob2	go	date	verdict	execution_time	language	source_code

# PostgreSQL

- ◆ 下図のように、「SQL文」と呼ばれるものを用いて、データベースにアクセスできます。
  - ◆ これはユーザーたちのID、名前、パスワードが保存されているテーブルです。



A terminal window with a dark background and light text. The title bar shows a window icon, a minus sign, a maximize button, and the text 'shun@shun-FMVMGG73D1: ~'. The terminal content shows a SQL query being executed and its results in a table format.

```
test=# select * from users;
```

userid	username	password
2	waka	password
4	a	b
5	nes	pa
1	name	pass
6	test	test
3	wakatsuki	password

(6 rows)

# PostgreSQL

- ◆ 下のようにユーザー名で検索したりすることもできます。
  - ◆ 実際、ログイン処理のときはこのような**SQL文**を用いています。

```
test=# select * from users where username = 'wakatsuki';
userid | username | password
-----+-----+-----
      3 | wakatsuki | password
(1 row)

test=#
```

# PostgreSQL

- ◆ 問題の解答結果は、下のようにすべてのユーザーで一括して一つのテーブルに保存しています。

shun@shun-FMVMGG73D1: ~							
scoreid	userid	prob_num	date	verdict	execution_time	language	source_code
1	0	0		dammy			
2	1	1	01/19/13 19:29:36	Runtime Error		Ruby	\
3	1	1	01/19/13 19:29:37	Runtime Error		Ruby	\
4	1	1	01/19/13 19:29:39	Runtime Error		Ruby	\
5	1	1	01/19/13 19:29:41	Runtime Error		Ruby	\
6	1	1	01/19/13 19:29:44	Runtime Error		Ruby	\
7	1	1	01/19/13 19:29:46	Runtime Error		Ruby	\
8	1	1	01/19/13 19:29:48	Runtime Error		Ruby	\
9	1	1	01/19/13 19:29:51	Runtime Error		Ruby	\
10	1	1	01/19/13 19:30:53	Accepted 0.0	0.0	Ruby	line = STDIN.gets\r : sum = 0\r : line.split.each do  a \r : sum += a.to_i\r : end\r : print sum : print 2
11	5	1	01/19/13 19:31:44	Wrong Answer		Ruby	print 2
12	5	2	01/19/13 19:31:51	Compile Error		C++	fewafe
13	5	0	01/19/13 19:31:58	Compile Error		C++	a
14	2	1	01/23/13 19:35:39	Compile Error		C++	line = SDTIN.gets\r : sum = 0\r : line.split.each do  a \r : sum += a\r : end\r : print sum\r :

# 苦勞した点

- ◆ 最も苦勞したのは、データベースの扱いです。
  - ◆ そもそも**SQL**というのがどんなものなのかすら知らなかったなので、ある程度慣れるまではかなり苦戦しました。
  - ◆ データベースの類を扱ったことが無かったので、保存するデータが増えてくるにつれて、どこにどのような形式で保存するかなどについても悩まされました。
- ◆ **HTML**も地味に大変でした。

# 勉強になった点

- ◆ Apache, PostgreSQL, HTML...などの基本的な扱い方
  - ◆ どれも今まで触ったことすらありませんでした。
- ◆ このような本格的(?)なプログラミングの際の根本的な組み立て方
  - ◆ 関数を細かく分けたり、クラスを用いて構造化したり...

# 今後の課題

- ◆ 一番の課題はもちろんセキュリティの問題。
  - ◆ 特に、ファイルのパーミッションは、何か問題が起きたら”**chmod 777**”としていたので、このあたりをしっかりと管理するように修正する必要がある。
- ◆ ネットワーク特有の問題に対処する必要がある。
  - ◆ ユーザー側が同じデータを**2度**送信してしまった場合など。
- ◆ ユーザーインターフェース
  - ◆ 見栄えを良くする。
  - ◆ 他のユーザーの情報を見れるようにする。