

データベースを利用した汎用的な学生証出席管理システム

安本太一* 鈴木啓**

* 情報教育講座

** 卒業生

Student Attendance Management System for General Purpose Using Studnet ID Card and Database

Taichi YASUMOTO* and Hajime Suzuki**

*Department of Information Sciences, Aichi University of Education, Kariya 448-8542, Japan

** Graduate, Aichi University of Education, Kariya 448-8542, Japan

1 はじめに

現在、本学の教務課に配置されている学生証出席管理システムは、2010年度の卒業生によって開発されたものである。Windows パソコンに USB 接続されたソニー社製 FeliCa リーダに学生証をかざすと、学生証に搭載されている FeliCa チップに記録されている学籍番号を読み取り日時とともに記録し、読み取りの開始から終わりまでに記録されたものを、CSV ファイルとして出力するものである。

職員であれば、この学生証出席管理システムを、FeliCa リーダを含めて教務課ノートパソコンを借りて使うことができるが、教務課がガイダンスなどで出席をとる時に使うほかは、教員はほとんど使用していないようである。

本研究では、より使いやすく魅力のあるものとするため、学生証出席管理システムを作り直し、多くの OS 上で動き、個人情報流出に配慮しつつ出席情報データをサーバに集積し、授業担当教員以外も出席情報データを参照できるようにした。作り直しは、現在の学生証出席管理システムを OS 非依存にしたうえで、ネットワーク対応にするという 2 段階で行った。

2 現在の学生証出席管理システム

現在の学生証出席管理システムのプログラムは、ソニー社製 Windows 用 FeliCa ライブラリを利用する x86 Windows 実行形式のプログラムで、C++Builder を使って開発されたようである。教務課貸出用ノートパソコンにインストールされているプログラムをみて、そのように推測した。教務課や卒業生には相談せずに作り直しを行ったので、筆者はソースプログラムをみていない。学籍番号と出席日時の組 (以下、出席情報データ

という) からなる CSV ファイルは、ノートパソコンの内蔵ハードディスクにいったん保存され、出席管理システムの使用者はこの CSV ファイルを USB メモリにコピーして取り出すことが前提になっている。この現在の学生証出席管理システムは次のような問題がある。

1. 古い卒業生が C++ 言語で開発したので、サポートできる者がいない。プログラムに不具合があってもすぐには直せないし、機能拡張をしたくても容易にはできない。
2. USB メモリを介した Windows のウイルス感染のインシデントがあった。
3. macOS 派や Linux 派の教員は、Windows に嫌悪感をもって使用できない。教員が使用しているコンピュータの OS は、Windows だけではない。
4. 学生証出席管理システムを動かしているノートパソコンを紛失したり、出席情報データの CSV ファイルを格納した USB メモリを紛失した場合に、学生の名前そのものは記録されてはいないが、出席情報データを第三者にみられるのは、好ましくない。
5. 教務課貸出用ノートパソコンは、いわゆるお古のものが転用されていることから、旧式で大きく重く、出席をとるためだけに持ち出すのは、躊躇してしまう。

3 学生証出席管理システムのプログラムの再作成

問題点を解決するために、現在の学生証出席管理システムを、機能はそのままにして、全く別の開発環境で作り直した。

プログラミング言語は Python, FeliCa リーダにアクセスするためのライブラリは nfcpy [1], GUI は kivy [2] を使用した. nfcpy は, NFC カードリーダーのための Python モジュールである. nfcpy は, Python モジュール usb1 を使用し, usb1 は USB デバイスにアクセスするための C ライブラリ libusb を使用する. libusb は, macOS 版, Linux 版, Windows 版などが提供されている. Python モジュール kivy も, macOS, Linux, Windows などでも利用できる. Python の実行環境は, macOS, Linux, Windows などでも提供されているので, 現時点で一般消費者が使うほぼ全ての OS で, 学生証出席管理システムが使えるようになった. 従来より, 良くなった点は次のとおりである.

1. C++ より習得が容易な Python で実装し, Python の csv モジュールを用いたことにより, プログラムの保守性が向上した. 元の C++ のソースプログラムをみていないのでなんともしえないが, ソースプログラムの長さも短くなったと推測される.
2. macOS や Linux で動作させることが可能になった. 全ての教員のノートパソコンにインストールして, 使用することができるようになった. Windows 以外の OS で動作させた場合は, USB メモリを介したウィルス感染の心配はないといっても過言ではない.
3. 小型化が可能になった. Linux が動作する Raspberry Pi などの小型ボードコンピュータを使えば, 学生証出席管理システムはかなり小さくなる. Linux であれば, OS のライセンス料は不要である. 筆者のところでは, 7 インチタッチスクリーンディスプレイを備えた Raspberry Pi で学生証出席管理システムが動作しており, 第一筆者の 2017 年度前期の授業の出欠確認に試用した (図 1 参照). 古い卒業生が作成したものと同様, 学生証をかざして学籍番号が読み取られると, ディスプレイに表示されるようになっていた. 学生は, 学籍番号の表示をみて, 確実に学生証が読み取られたことを確認する. タッチスクリーンディスプレイは必須ではなく, LCD キャラクタディスプレイと押ボタンスイッチなどに置き換えることは可能である.

4 学生証出席管理システムのネットワーク対応

4.1 サーバへの集積と学籍番号に替えるもの

学生証出席管理システムを作り直したが, 出席情報データの CSV ファイルを USB メモリにコピーするな



図 1: 小型ボードコンピュータによる学生証出席管理システム

どしなければいけないことには変わりはなく, わずらわしい. そこで, 学生証をかざしてもらって得られた出席情報データをネットワークを介してサーバに集積し, 教員などがネットワークを通じて後から参照できるようにすると便利である.

しかしながら, 学籍番号と日時からなる出席情報データをそのままサーバに格納するのは, 万一サーバに不正侵入された場合のことを考えると, 適切ではない. そこで, 学籍番号に替えて FeliCa の IDm (製造番号のようなもの) を用いることにした. IDm は各 FeliCa カードに固有の 16 桁の 16 進数である. 具体的には 012E3D24690D7849 といった数字で, 無味乾燥であり, 個人の特定につながらない. 学生証を持っている本人も, IDm を知らないことから, 出席情報データを集積したサーバに不正侵入され, サーバに格納されている出席情報データを盗み見られても, 差し支えないと考えられる. サーバにアクセスする者が, IDm と学籍番号の対応データをもっていれば, IDm から学籍番号への変換はできる. 授業担当教員や指導教員といった本当に必要な者だけが, 責任の対象の学生分に限って IDm と学籍番号の対応データを保持するのが, 個人情報保護の観点から望ましい.

4.2 サーバへの集積方法

複数の教員や事務系職員に使ってもらうことを想定して, 学生証の FeliCa の IDm, 日時に加えて, 教室などに対応する番号 (deviceid と名付けた) も記録することにした. したがって, 本稿では, これから, 出席情報データとは, 日時, 学生証の IDm, 教室 (deviceid) の組のことをいう.

出席情報データを集積するために, 汎用性を考え, 独自のサーバプログラムを作するのではなく, RDBMS (関

係データベース管理システム)を用いることとした。RDBMSの中でも、PostgreSQLを用いることにした。そして、出席情報データを格納するために、5つの列 id(integer, primary key), date(date), time(time without time zone), idm(text), deviceid(text)を持つテーブルを作成することにした。括弧内はデータ型である。

idは1つの行(1件の出席情報データ)を識別するための自動連番で、serialというデータ型を指定するとともに、プライマリーとした。dateとtimeは日時、idmは学生証のIDm、deviceidは教室(deviceid)に対応する。

通常同一の時に複数の授業が開講されており、複数の授業の出席情報データを単一のテーブルに集積しても、時間帯と教室(deviceid)を指定すれば、自分の担当している授業の出席情報データを取得できる。例えば、2017年11月16日14時55分から16時35分という4限目の時間帯と310教室を指定して検索(SELECT)するのである。4限目は15時00分から16時30分であるが、早く来た学生や授業が延びた場合を考慮して、幅を持たせた方が良い。

教室をdeviceidとしているのは、FeliCaリーダを接続したコンピュータ機器にidを付けるという考え方に基づいている。教室に1台ずつFeliCaリーダを接続したコンピュータ機器を配置する場合はidを教室の部屋番号とすれば良いし、教員が自分専用のFeliCaリーダを接続したコンピュータ機器を持ち歩く場合は、教員を特定する唯一のidを付ければよい。講堂におけるガイダンス、大教室で行われる授業では、効率よく出席をとるために、FeliCaリーダを備えたコンピュータ機器を複数台用意するかもしれないが、どの機器へ学生証をかざしたを区別したい場合は、教室部屋番号に枝番が続くような唯一のidを機器につければ良い。

4.3 サーバへの集積方針

RDBMSであるPostgreSQLを採用し、出席情報データを格納するテーブルを定義したので、多くの教員、できれば全ての教員が担当している授業の出席情報データをサーバに集積するのが望ましい。そうなれば、例えば、指導教員が、指導学生の学籍番号に対応するIDmを指定して検索(SELECT)すると、指導学生の授業の出席情報データ(の行)が取得でき、指導学生がきちんと授業を履修しているかどうか把握できる。後からどのように使うかは別として、出席情報を積極的にサーバへ集積して行くことが大切である。

そこで、先述の再作成した出席情報データをCSVファイルに出力するPythonプログラムに、ネットワークを介してPostgreSQLのデータベースのテーブルに出席情報データを追加(INSERT)する機能を追加した。図1の機器では、Stopボタンを押して学生証の読み取りを

停止し、Endボタンを押してプログラムの終了の指示をすると、プログラムの終了直前にネットワークを介してサーバに出席情報データを追加(INSERT)するようにした。図1の機器は、Wi-Fiインタフェースを備えているので、教室のWi-Fiアクセスポイントに接続して使っている。PythonからPostgreSQLサーバに接続するモジュールには、psycopg2を使用した[3]。本研究で用いているPostgreSQLサーバは、第一筆者の教員室のRaspberry Pi上で動作させているものである。

5 サーバに集積した出席情報の利用

サーバに集積した出席情報データの利用例として、授業の出欠表を作成するクライアントプログラムと特定の学生の出席状況を取得するクライアントプログラムをPythonで記述した。これらは、macOS, Linux, Windowsのいずれでも動作する。

5.1 出欠表

授業担当教員が、サーバに集積した出席情報データから、第一に欲しいのは担当授業の出欠表であろう。出席情報を登録するテーブルは、学期ごとに作成しておき、曜日、時限、教室を指定して検索(SELECT)すれば、担当授業の出席情報データ(の列)が取得できそうなものだが、時間割変更、教室変更、休講と補講があるので、一律の指定では無理がある。そこで、図2に示す授業の各回の日時と教室を記載したCSVファイルを入力とし、図3に示す出欠表をCSVファイルとして出力する出欠表作成プログラムをPythonで作成した。授業の日時と教室のCSVファイルの作成は面倒なように思われるかもしれないが、CSVファイルの作成に表計算ソフトを使うのであれば、オートフィルで、通常授業の1週間おきの日時を自動入力できる場合があるので大した手間ではない。CSVファイルなら、集中講義や週2回授業にも対応できる。

図3の出欠表は、1のところが出席、空欄が欠席、最後の列は出席回数である。出欠表には学籍番号が表示されているが、サーバから取得できる出席情報データには、学生を識別する情報としてはIDmが使われているので、学生を学籍番号として表示するためには出欠表作成プログラムが動作するコンピュータに、IDmと学籍番号の対応データが必要である。

5.2 特定の学生の授業出席状況

学生番号をキーにして、検索(SELECT)すれば、該当する学生が出席した全ての授業において学生証をFeliCaリーダにかざした日時の一覧を取得することができる。

```

授業開始日時 授業終了日時 教室
2017/4/6, 10:52, 2017/4/6, 12:30, 314
2017/4/13,10:52, 2017/4/13,12:30, 314
2017/4/20,10:44, 2017/4/20,12:30, 314
2017/4/27,10:52, 2017/4/27,12:30, 314
2017/5/11,10:52, 2017/5/11,12:30, 314
2017/5/18,10:52, 2017/5/18,12:30, 314
      ⋮
2017/7/20,10:52, 2017/7/20,12:40, 314
2017/7/27,10:52, 2017/7/27,12:40, 314
    
```

図 2: 授業時間と教室を記した CSV ファイルの内容

出席情報を登録するテーブルを、学期ごとに作成しているならば、学期の始めからの一覧を取得することができる。指導教員などが、授業を休みがちな学生を、注視するのに活用できる。学籍番号を与えると、授業出席状況をダンプするプログラムを Python で作成した。実行例を図 4 に示す。このプログラムを利用するためにも、IDm と学籍番号の対応データが必要で、全ての学生の授業出席状況が無制限に取得できるわけではない。

```

2017-04-06 09:25:22 314
2017-04-10 15:25:33 314
2017-04-13 09:25:06 314
      ⋮
2017-07-24 16:26:26 314
2017-07-27 10:34:52 314
2017-07-31 15:32:25 314
出席回数 34
    
```

図 4: 特定の学生の出席状況のダンプ

5.3 その他の利用用途

教室 (deviceid) をキーにして検索 (SELECT) すれば、当該教室で学生証をかざした出席情報の一覧が得られるが、これを授業の各時限の時間帯ごとに集約すれば、その教室の利用状況が得られる。例えば、火曜日の 3 限目に 314 教室で、学生証をかざした出席情報があれば、火曜日 3 限目に 314 教室が使われていると判定する。教務課の教室予約表では使っていることになっているのに、実際には使われていない教室が存在することがあるので、出席情報から間接的に教室の利用状況の実態がわかることは有意義である。この応用として、教員がきちんと授業を実施しているか否かの調査にも使える。出席状況データを 1 つのテーブルに集積していけば、さまざまな使い道がある。

5.4 メンテナンスプログラム

ネットワークが不調でサーバに出席情報データを送信できなかったりするトラブルなどに対処するために、次に示すいくつかのメンテナンスプログラムを Python で作成した。

- 出席情報遅延追加
日時, IDm, 教室 (deviceid) からなる CSV ファイ

StartTime	2017/4/6	2017/4/13	2017/4/20		2017/7/13	2017/7/20	2017/7/27	
	10:52	10:52	10:44		10:52	10:52	10:52	
FinishTime	2017/4/6	2017/4/13	2017/4/20	2017/7/13	2017/7/20	2017/7/27	
	12:30	12:30	12:30		12:40	12:40	12:40	
2190760	1	1	1		1	1	1	13
2190763	1		1		1	1		14
2190764	1	1	1		1	1		14
2190765	1	1	1					5
2190766	1	1	1		1			14
		⋮				⋮		
2190787	1	1	1		1	1	1	14
2190788	1	1	1		1	1	1	16
2190789	1	1	1	1	1	1	16
2190790	1	1	1		1	1	1	16
2190791	1	1	1		1	1	1	14

CSV ファイルを表計算ソフトで表示したもの。学籍番号などは架空。

図 3: 出欠表

ルを読み込み、サーバに出席情報データを送信して追加する。学生証をかざしてもらった出席情報データはあるが、ネットワークが不調だった場合に、後でデータ送信をするために使用する。

- データの手動入力 (対話入力版)
学籍番号, 日時, 教室 (deviceid) を対話的に入力して, サーバに 1 件の出席情報を送信して追加する。学生証を忘れた学生の出席情報データを追加するために使用する。IDm と学籍番号の対応データが必要である。対応データがない場合は, エラーとなる。
- データの手動入力 (一括版)
上記と役割は同じだが, 学籍番号, 日時, 教室 (deviceid) からなる CSV ファイルを入力とする。

5.5 Python 以外によるクライアントプログラム

Python 以外のプログラミング言語によっても, 例えば C 言語でもサーバにアクセスするクライアントプログラムを作成することが可能である。FileMaker, LibreOffice Base, Microsoft Access といったパソコン上のデータベースソフトウェアからも, PostgreSQL サーバにアクセスすることが可能である。出席情報データの集積に独自のプログラムではなく, PostgreSQL を使ったので, 出席情報データを利用したい者が好む方法でクライアントとなって出席情報データを利用可能である。

6 まとめ

学生証の FeliCa に格納されている情報を用いて PostgreSQL サーバへ出席情報データを集積するプログラムと, 集積した出席情報データを利用するプログラムを開発した。これらは, Python で開発されており, Windows 以外の OS でも使用可能である。

大学内の全ての教室に FeliCa リーダを備えた機器を設置できるのが望ましいが, 費用の問題がある。Raspberry Pi をベースに表示器を LCD キャラクタディスプレイにするなどの低廉化の工夫をすれば, 費用を抑えることができる。Raspberry Pi は小さくて軽くバッテリー駆動も可能なので, FeliCa リーダを接続した Raspberry Pi を授業中に学生に回してもらって出席をとることも可能であり, そのように運用すれば学生証をかざすための学生の列ができることもない。

リーダを備えた機器を設置しても, ネットワークが整備されていない教室では, サーバへ出席情報データを自動送信できない。そのような教室では, モバイル Wi-Fi ルータを携帯した守衛が夕方に教室を施錠する

ときに, 機器から一日分の出席情報をサーバに一括送信するというバッチ処理のような形態が考えられる。例えば, 図 1 の機器に, 未送信の出席情報データを一括送信する機能を付加すれば良い。モバイル Wi-Fi ルータの利用費用は, いわゆる格安 SIM というものを利用すれば, 低く抑えられる。

出欠表を作成する Python プログラムは, 出席情報データをサーバに要求する日時や場所の指定を CSV ファイルとすることにより, 柔軟な対応が可能である。サーバに格納される出席情報データにおいて, 学生を特定するのを学籍番号ではなく FeliCa の IDm にして, 個人情報流失対策を講じた。IDm と学籍番号の対応づけの管理が, 本論文で提案するシステムの安全性の要である。この管理を安全に行う方法については, 今後の課題である。サーバに逐次格納されていく出席情報データをもとに, 欠席が多いと判定される学生について, 手遅れになる前に, 学生本人, 指導教員, 保証人 (親など) に注意喚起のメールを自動送信するといった機能の開発も, 今後の課題としてあげられる。

参考文献

- [1] Tiedemann, S. : Python module for near field communication, <http://nfcpy.readthedocs.io/en/latest/>.
- [2] Kivy community project : Kivy - Open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps, <http://kivy.org/>.
- [3] Gregorio, F. D. and Varrazzo, D. : Psycopg - PostgreSQL database adapter for Python, <http://initd.org/psycopg/docs/>.

(2017年11月30日受理)