

研究室開閉時刻可視化システムの開発

大阪産業大学 デザイン工学部 情報システム学科
情報教育システム研究室

15H038 小藪竜也

研究室開閉時刻可視化システムの開発

15H038 小藪竜也

1 はじめに

研究室や教員の居室には、“帰宅”や“講義”など教員の行き先を示す行き先表示板が設置されている。これは教員が、居室を訪れる人に対して自分の居場所を示すとともに、部屋の鍵が開いているかどうかを示している。これにより、教員の居室においては開閉状況が明確になっている。

しかし、研究室においては学生が訪れた際に、研究室の鍵が開いているかどうか分かる方法がなければ、開いているかどうかは研究室の前に着くまでわからない状態になっている。この問題を解決するため、また研究室の開閉時刻を明確にするための専用システムが必要である。

2 目的

研究室において、研究室の開閉時刻が明確になっていない問題（以下、開閉時刻不明問題とする。）がある。そのため、学生が研究室を訪れるまで、研究室が開いているかどうかははっきりせず、訪れた際にはまだ開いていなかったり、すでに閉まっていたという問題が起こりうる。

本研究の目的は、研究室の解錠時刻と施錠時刻の可視化を実現するためのシステムの開発である。本システムによって、開閉時刻不明問題の解決を目指す。

3 開閉時刻可視化システム

本システムは、研究室の開閉時刻を可視化することにより、学生が研究室を訪れる前に研究室が開いているかどうかの確認を行うものである。そのため、研究室に配属されている学生が研究室への来訪時刻と帰宅時刻をメールで報告する必要がある。それらを基に開閉時刻を自動的に割り出し、メールでの送信や Web での表示を行うシステムを“人々の行動を手軽に共有する生活空間エージェント [1]”を参考に開発した。

4 検証

デザイン工学部情報システム学科の学生 15 名を対象に情報教育システム研究室への来訪/帰宅時刻のメールでの報告を求め、集まったデータを基に検証を行った。

4.1 概要

検証の対象者に月曜日から土曜日の 7:30 JST と 12:30 JST に案内のメールをシステムから送り、来訪時刻と帰宅時刻を書いて返信するよう求めた。

4.2 実際の開閉時刻の取得方法

実際の開閉時刻は、情報教育システム研究室において既に運用されている研究室内照明明滅検知システムを利用して取得を行った。

5 まとめ

本研究では、ユーザからメールで研究室に来訪時刻と帰宅時刻を送ってもらうことにより、研究室の開閉時刻の可視化を行い開閉時刻不明問題を解決するシステムを開発した。その結果、ユーザが研究室の解錠時刻と施錠予定時刻を研究室を訪れる前に確認することができる。また、ユーザ自身の来訪時刻も他のユーザに共有される為、ユーザの行動を可視化し、研究室を訪れていない学生の割り出しに活用することが期待できる。

参考文献

- [1] 川上 あゆみ・水上 彩・塚田 浩二・椎尾一郎. 人々の行動を手軽に共有する生活空間エージェント. ヒューマンインタフェースシンポジウム論文集, 2009.

目次

1	はじめに	1
2	目的	2
3	開閉時刻可視化システムの概要	3
3.1	本システムの概要	3
3.2	動作	4
3.3	本システムの詳細	4
4	関連研究 (人々の行動を手軽に共有する生活空間エージェント [1])	29
4.1	関連研究の概要	29
4.2	関連研究のメリット	29
4.3	関連研究の問題点	29
4.4	関連研究の問題点を本研究でどう解決するか	29
5	検証	30
5.1	検証対象者	30
5.2	検証方法	30
5.3	検証結果	31
6	結果と考察	42
6.1	検証結果の考察	42
6.2	本研究で開発したシステムの利点と問題点	42
6.3	利点	42
6.4	問題点	44
7	結論	45
7.1	今後の課題	45
付録 A	ソースコード	48
A.1	受信時処理	48
A.2	当日処理	50
A.3	翌日以降処理	62
A.4	随時処理	73
A.5	指定時刻処理	86
A.6	Web ページ装飾	102

1 はじめに

研究室や教員の居室には、“帰宅”や“講義”など教員の行き先を示す行き先表示板が設置されている。これは教員が、居室を訪れる人に対して自分の居場所を示すとともに、部屋の鍵が開いているかどうかを示している。これにより、教員の居室においては開閉状況が明確になっている。しかし、研究室においては学生が訪れた際に、研究室の鍵が開いているかどうか分かる方法がなければ、開いているかどうかは研究室の前に着くまでわからない状態になっている。この問題を解決するため、また研究室の開閉時刻を明確にするための専用システムが必要である。

第2章では本研究の目的について述べる。第3章では本研究で開発したシステムの概要と処理について述べる。第4章では本研究に関連する先行研究について述べる。第5章では本研究で開発したシステムを用いて実施した検証について述べる。第6章では検証の結果をその考察とともに述べる。第7章では研究の成果とともに今後の課題についてまとめる。

2 目的

研究室において、研究室の開閉時刻が明確になっていない問題（以下、開閉時刻不明問題とする。）がある。そのため、学生が研究室を訪れるまで、研究室が開いているかどうかははっきりせず、訪れた際にはまだ開いていなかったり、すでに閉まっていたという問題が起こりうる。

本研究の目的は、研究室の解錠時刻と施錠時刻の可視化を実現するためのシステムの開発である。本システムによって、開閉時刻不明問題の解決を目指す。

3 開閉時刻可視化システムの概要

本研究は、研究室の開閉時刻を可視化することにより、学生が研究室を訪れる前に研究室の開閉状況の確認を行うものである。そのため、研究室に配属されている学生が研究室への来訪時刻と帰宅時刻をメールで報告する必要がある。それらを基に開閉時刻を自動的に割り出し^{*1}、メールでの送信や Web での表示を行うシステムを開発した。

3.1 本システムの概要

本システム全体の概要図を図 1 に示す。主に 3 つの機能を実装した。次項より各機能の解説を行う。

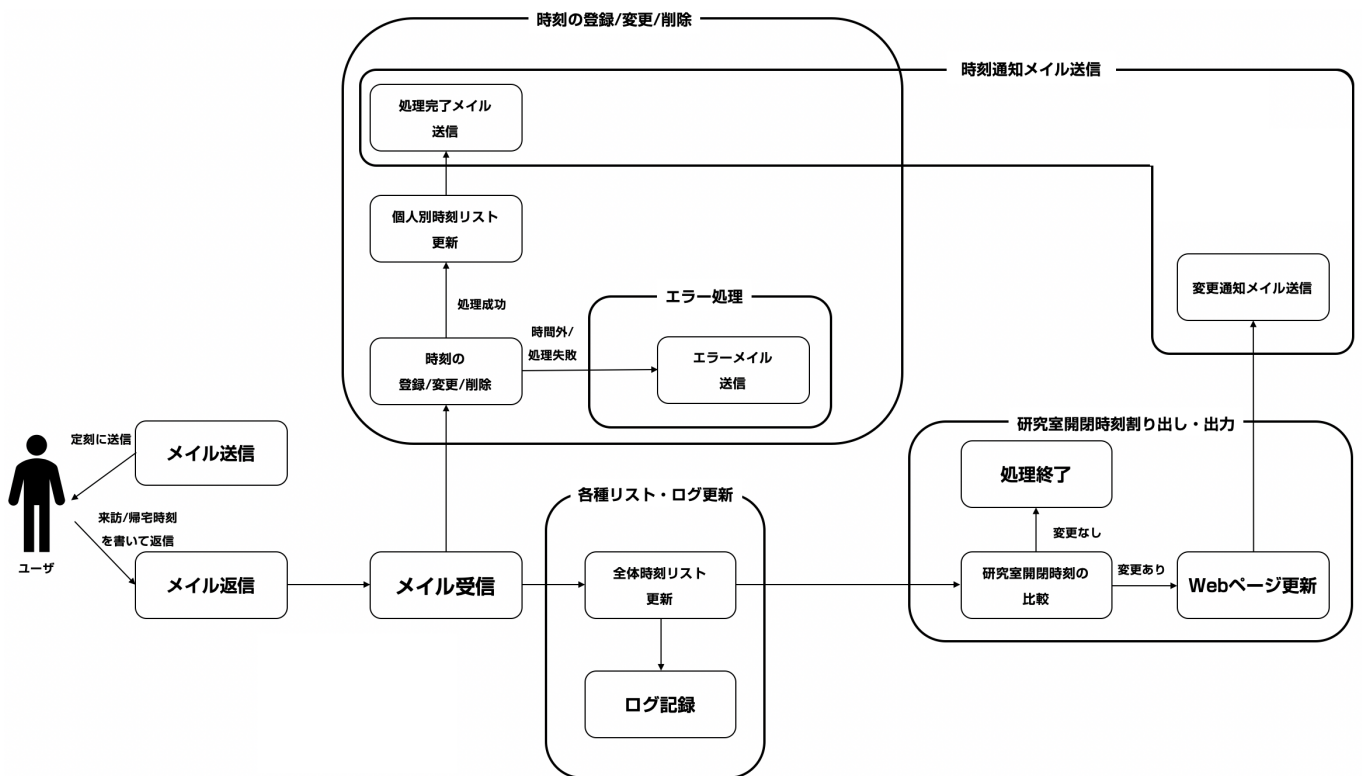


図 1 本システムのユーザから見た動作の概要。主に時刻の登録/変更/削除、研究室開閉時刻の割り出し・出力、時刻通知メールの送信の 3 つの機能を有する。本システムにおけるユーザは学生である。

1. 時刻の登録/変更/削除
2. 研究室開閉時刻割り出し・出力
3. 時刻通知メール送信

^{*1} 学生から送られた時刻をすべてリスト化し、一番早い来訪時刻を解錠予定時刻、一番遅い帰宅時刻を施錠予定時刻として割り出す。

3.2 動作

以下に本システムの動作を示す。

1. 定刻^{*2}にメールを本システムを運用するサーバ(以下、サーバとする。)から専用メイリングリスト^{*3}宛に自動送信。
2. 各自^{*4}そのメールに来訪時刻と帰宅時刻を返信する。
3. サーバにメールが届くと、時刻とメールアドレスを抜き出し、来訪時刻と帰宅時刻に分け、それぞれの個人リスト^{*5}に書き込む。
4. 正常に処理が完了した旨を本人宛にメールを自動送信。エラーの場合はエラー内容を書いたメールを本人宛に自動送信。
5. 開閉時刻割り出し用全体リストに個人リストの内容を書き込む。
6. 開閉時刻表示用 Web ページの自動更新。
7. 送られてきた時刻を表示する Web ページの自動更新。

3.3 本システムの詳細

3.3.1 ユーザからシステムへの報告

ユーザから送られてくる時刻を共有する必要があるため、本研究専用メイリングリストを作成した。そのメイリングリスト宛にメールを送ることによりメンバに共有され、サーバ側でも処理が行われるようにしている^{*6}。

3.3.2 返信用メールの送信

毎日 7:30 JST と 12:30 JST に当日の来訪時刻と帰宅時刻の登録を求める案内メールを専用のメイリングリスト宛に自動的に送信する。ユーザは届いたメールの本文に来訪/帰宅時刻を書いて返信するだけで時刻の登録を行うことができる。

^{*2} 7:30 JST と 12:30 JST。JST は Japan Standard Time、日本標準時のこと。

^{*3} 登録したユーザに同じメールが送られるシステムである。

^{*4} 3.1 におけるユーザ。

^{*5} メールアドレスに含まれるユーザ名を取り出してファイル名の最後につけ、個別にリストファイルを作成している。

^{*6} メールを処理するためのアカウントをメイリングリストに追加し、そのアカウントにメールが流れることにより処理が行われる。

3.3.3 処理の分岐

ユーザから本システムに対する処理の指示はすべてメールにて行う。その際、Subject:をエイリアス^{*7}により呼び出されたメール振り分けスクリプト^{*8}で読み取り、処理を分岐^{*9}させている。ユーザが本システムにメールを送る際に指定することができる Subject の一覧を表 1 に示す。

表 1 ユーザからシステムに送ることができる Subject の一覧。Subject に書く文字と役割が記されている。なお、“C” と “Closetime” についてはどちらを書いても同じ動作をする。

Subject:	役割
Closetime	来訪/帰宅時刻の登録 (自動送信メール返信用)
C	来訪/帰宅時刻の登録 (手動送信用)
E	登録済み来訪/帰宅時刻の変更
D	登録済み来訪/帰宅時刻の削除
F[日付 8 桁]	指定した日の来訪/帰宅時刻の登録
E[日付 8 桁]	指定した日の登録済み来訪/帰宅時刻の変更
D[日付 8 桁]	指定した日の登録済み来訪/帰宅時刻の削除
R	登録済み来訪/帰宅時刻の参照
R[日付 8 桁]	指定した日の登録済み来訪/帰宅時刻の参照

^{*7} 別名の意。ここではサーバにおいてメール受信をトリガにプログラムを起動させるシステムを指す。

^{*8} メールを標準入力で受け取り、Subject:を抜き出してコマンドを判別し、各種処理別に用意されたスクリプトを起動させるスクリプト。その際にメールを標準入力与えて起動させる。

^{*9} Subject:にコマンドを書き、指示内容を本文に記入する。

3.3.4 研究室来訪/帰宅時刻の登録

メールの Subject:を“Closetime^{*10}”もしくは“C^{*11}”とし、本文に来訪時刻と帰宅時刻を書いて送ることで時刻の登録を行うことができる。このとき、HH:MM^{*12}の形式で書く必要がある。サーバにメールが届き、エイリアスでメールを受け取った後スクリプトが起動され、From:^{*13}のメールアドレスと、メールのユーザ名^{*14}と時刻を抜き出し、時刻を個別時刻リストファイル^{*15}に書き込んで、処理完了次第ユーザに対してメールで通知し、時刻のリストを表示する Web ページを自動更新する。この際、締切として設定されている時刻^{*16}以降にメールが届いた場合にはエラーメールをユーザに対して送り、登録の処理は行われない。処理の概要を図 2、処理完了メールの例を図 3 にそれぞれ示す。

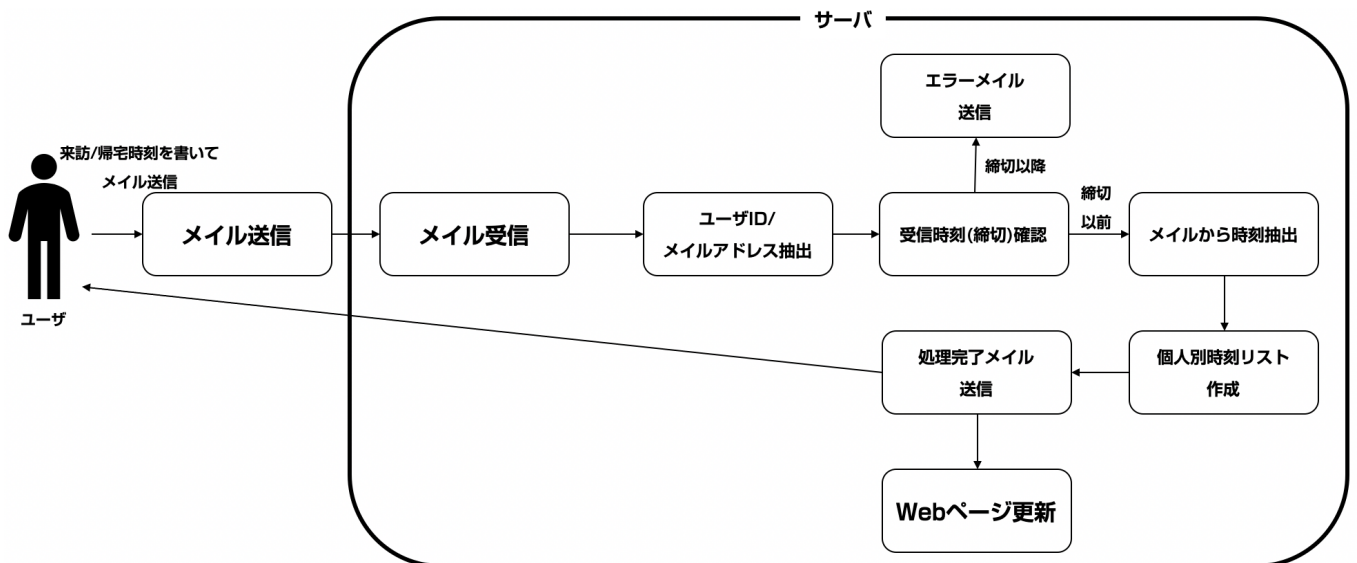


図 2 研究室来訪/帰宅時刻の登録処理の概要。ユーザの送信したメールからメールアドレスと時刻を抽出、個別時刻リストを作成し、正常に処理が完了すればユーザ宛に処理完了メールを、締切以降に送信されたメールである場合にはエラーメールを送信する。エラーの場合は登録処理を行わない。

*10 3.3.2 で送信するメールの Subject:も同じであるため、そのメールに対する返信を前提として Subject:を設定している。

*11 Closetime の頭文字

*12 2 バイト文字ではエラーとして処理される。

*13 送信元のメールアドレスが記されている項目。

*14 メールアドレスの左部分。

*15 ファイルは来訪時刻のリストと帰宅時刻のリストに分けて作成し、ファイル名の最後にユーザ名を入れている。

*16 毎日 17:45 JST を締切としている。これは大阪産業大学 (以下、本学とする。) の 5 時限目の終了時刻が 17:40 JST であり、その時刻以降に初めて研究室を訪れる人物はほとんどいないことを根拠としている。

████████████████████.a4w.ise.osaka-sandai.ac.jp

To 自分 ▼

s15h038 さんの来る予定の時刻と帰る予定の時刻が登録されました。

----->&----->&----->&----->&----->&-----

来る予定の時刻:15:25

帰る予定の時刻:18:40

----->&----->&----->&----->&----->&-----

登録された時刻の変更は、████████████████████.a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「E」、本文に新しい時刻を書いて送ってください。

キャンセルする場合は、████████████████████.a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「D」と書いて空メールを送って下さい。

(大文字と小文字は区別されません。████████████████████)

本メールは自動で作成されたメールです。返信しないでください。

図3 研究室来訪/帰宅時刻の登録処理完了メールの例。登録処理が行われたユーザ名、登録された時刻と、時刻の変更・削除の方法が書かれている。

3.3.5 研究室来訪/帰宅時刻の変更

本システムにメールで送っていた時刻を変更したい場合は、メールの Subject:を“E*¹⁷*¹⁸”とし、本文に変更後の時刻を書いて送ることで時刻の変更を行うことができる。本処理も登録処理と同様の方法で From:のメールアドレスとユーザ名、時刻を抜き出し、処理完了次第メーリングリスト宛にメールで通知し、時刻のリストを表示する Web ページを自動更新する。この際、締切として設定されている時刻以降にメールが届いた場合、登録処理がなされていないのに変更のメールが届いた場合にはエラーメールをユーザに対して送り、変更の処理は行われない。処理の概要を図 4、処理完了メールの例を図 5 にそれぞれ示す。

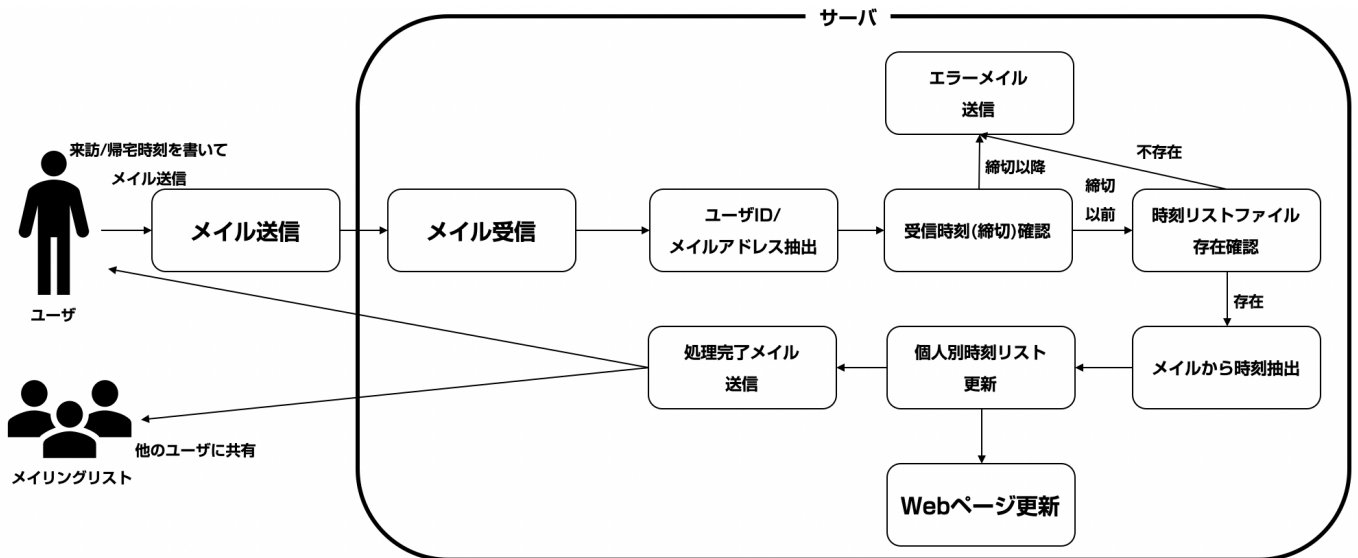


図 4 研究室来訪/帰宅時刻の変更処理の概要。ユーザの送信したメールからメールアドレスと時刻を抽出、個別時刻リストの存在を確認し、存在しない場合はエラーとする。正常に処理が完了すればメーリングリスト宛に処理完了メールを、締切以降に送信されたメールである場合にはユーザ宛にエラーメールを送信する。エラーの場合は変更処理を行わない。

*¹⁷ 大文字と小文字は区別せず、どちらで書いていても処理される。2 バイト文字の場合は処理されない。

*¹⁸ Edit の頭文字。

[Redacted] .a4w.ise.osaka-sandai.ac.jp

To [Redacted] ▼

s15h038 さんの送られた時刻が変更されました。

----->&----->&----->&----->&----->&-----

来る予定の時刻:13:54

帰る予定の時刻:15:30

----->&----->&----->&----->&----->&-----

帰る予定の時刻の変更は、[Redacted] .a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「E」、本文に新しい時刻を書いて送ってください。

キャンセルする場合は、[Redacted] .a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「D」と書いて空メールを送って下さい。

(大文字と小文字は区別されません。 [Redacted])

本メールは自動で作成されたメールです。返信しないでください。

図 5 研究室来訪/帰宅時刻の変更処理完了メールの例。時刻の変更処理が行われたユーザ名と変更後の時刻、本人向けに時刻の変更・削除の方法が書かれている。

3.3.6 研究室来訪/帰宅時刻の取り消し

本システムにメールで時刻を送っていても、都合が悪くなり研究室に行くことができなくなったなどの場合は、メールの Subject:を“D^{*19}”とし、空メールを送ることで時刻の取り消しを行うことができる。本処理は送られてきたメールのヘッダを読み取り、From:のメールアドレスとユーザ名を抜き出し、該当ユーザのファイルを削除する。処理完了次第メイリングリスト宛にメールで通知し、時刻のリストを表示する Web ページを自動更新する。この場合も変更処理と同様に、締切として設定されている時刻以降にメールが届いた場合、登録処理がなされていないのに取り消しのメールが届いた場合にはエラーメールをユーザに対して送り、削除の処理は行われない。処理の概要を図 6、処理完了メールの例を図 7 にそれぞれ示す。

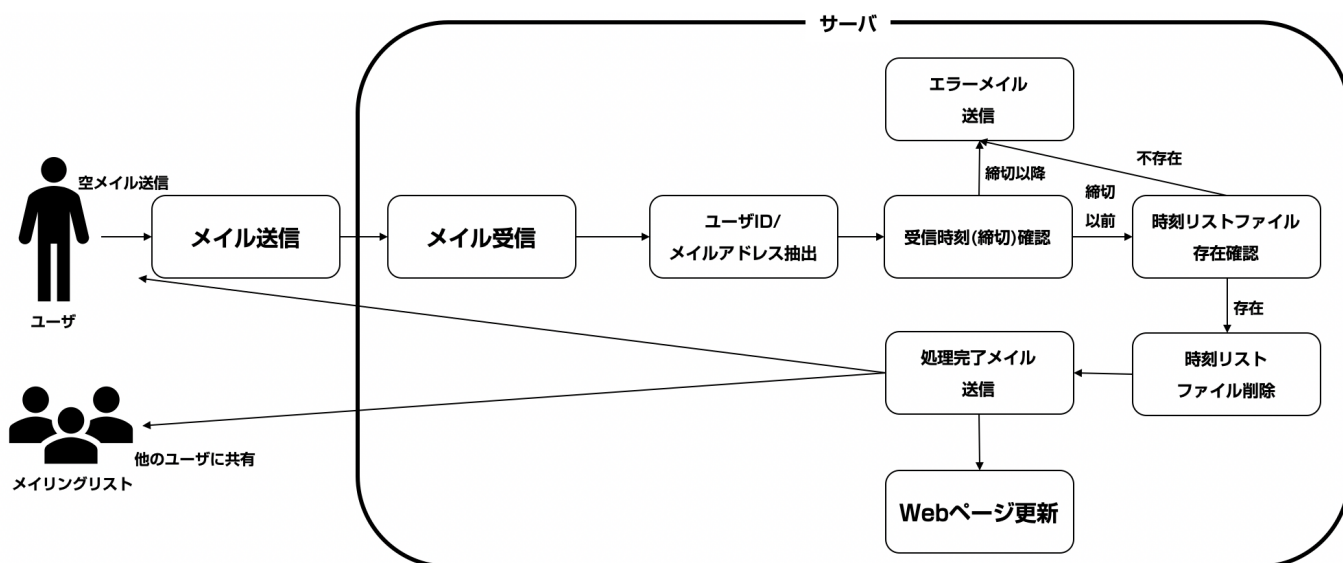


図 6 研究室来訪/帰宅時刻の取り消し処理の概要。ユーザの送信したメールからメールアドレスを抽出、個別時刻リストの存在を確認し、存在しない場合はエラーとする。正常に処理が完了すればメイリングリスト宛に処理完了メールを、締切以降に送信されたメールである場合にはユーザ宛にエラーメールを送信する。エラーの場合は取り消し処理を行わない。

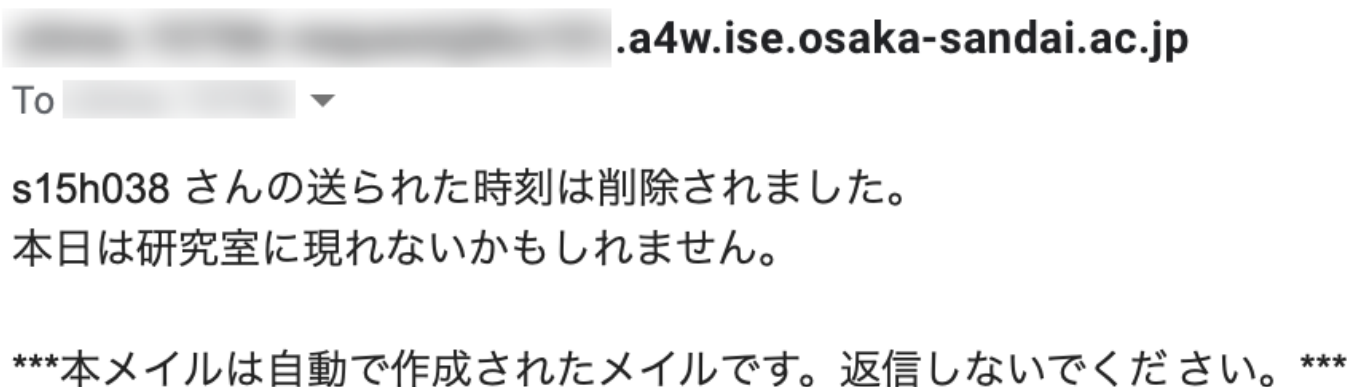


図 7 時刻の取り消し処理完了メールの例。取り消し処理が行われたユーザ名と当日は研究室を訪れない可能性がある旨が書かれている。

*19 Delete の頭文字。

3.3.7 翌日以降の研究室来訪/帰宅時刻の登録

翌日以降の研究室を訪れる時刻が判明している場合、メールの Subject:を“F*²⁰[数字 8 桁]*²¹”として日付を指定し、本文に指定した日の来訪時刻と帰宅時刻を書いて送ることで翌日以降の時刻の登録を行うことができる。本処理もメールから From:メールアドレスとユーザ名、時刻の抜き出しに加えて、Subject:から日付*²²の抜き出しも行う。また、指定された日付が翌日から 1 週間以上先、あるいは過去の日付*²³である場合にはエラーメールをユーザに対して送り、登録の処理は行われない。処理の概要を図 8、処理完了メールの例を図 9 にそれぞれ示す。

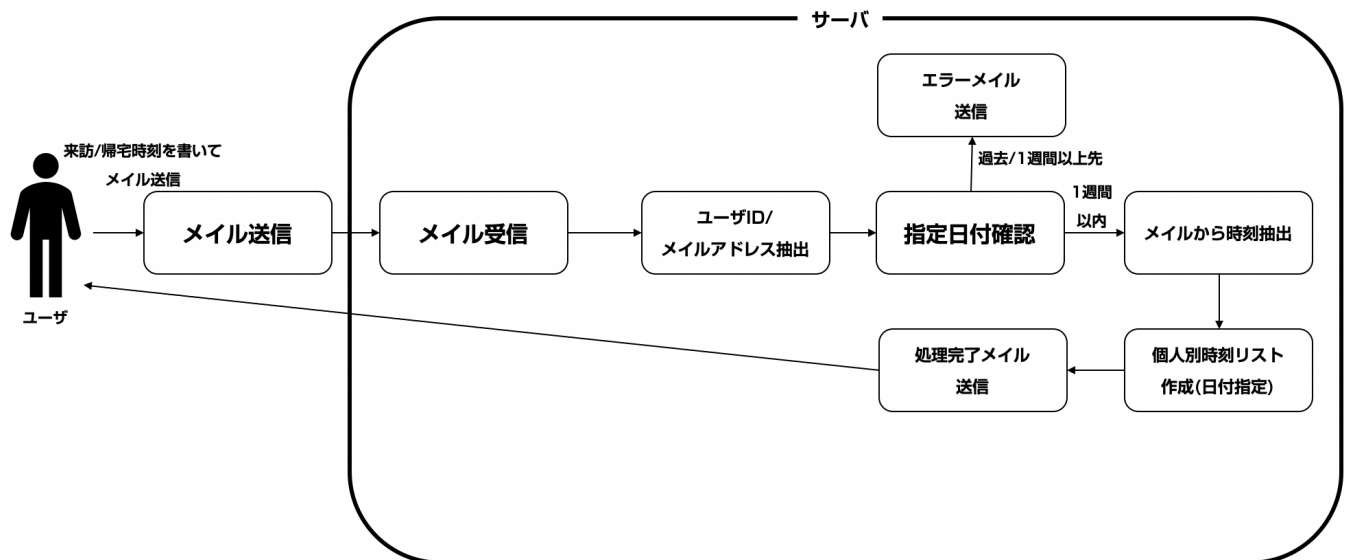


図 8 翌日以降の研究室来訪/帰宅時刻の登録処理の概要。ユーザの送信したメールからメールアドレスと時刻、日付を抽出し、指定された日付の確認を行い、日付が正しければ指定された日付の個別時刻リストを作成する。正常に処理が完了すればユーザ宛に処理完了メールを、指定された日付が 1 週間以上先、または過去の日付の場合にはエラーメールを送信する。エラーの場合は変更処理を行わない。

*²⁰ Future の頭文字。

*²¹ YYYYMMDD 形式で書く必要がある。

*²² 抜き出した日付はリストファイルの頭につけられ、当日用との区別を行う。

*²³ 当日の日付もエラーで返す。当日は、日付を指定する必要がないからである。

[redacted].a4w.ise.osaka-sandai.ac.jp

To 自分 ▼

<<この時刻は2018/12/18分です>>

s15h038 さんの2018/12/18の来る予定の時刻と帰る予定の時刻が登録されました。

----->&----->&----->&----->&----->&-----

来る予定の時刻:09:00

帰る予定の時刻:21:00

----->&----->&----->&----->&----->&-----

帰る予定の時刻の変更は、[redacted].a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「E20181218」、本文に新しい時刻を書いて送ってください。

キャンセルする場合は、[redacted].a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「D20181218」と書いて空メールを送って下さい。

(大文字と小文字は区別されません。[redacted])

本メールは自動で作成されたメールです。返信しないでください。

図9 翌日以降の研究室来訪/帰宅時刻登録処理完了メールの例。登録処理が行われたユーザ名、指定された日付と登録された時刻、時刻の変更・削除の方法が書かれている。

3.3.8 翌日以降の研究室来訪/帰宅時刻の変更

翌日以降の研究室を訪れる時刻を送っていても、何らかの理由により変更があることが前日までの段階で判明している場合、メールの Subject:を“E[数字 8 桁]”として日付を指定し、本文に指定した日の変更後の時刻を書いて送ることで翌日以降の時刻の変更を行うことができる。本処理も翌日以降の登録処理と同様の方法で From:のメールアドレスとユーザ名、時刻、日付の抜き出しを行い、処理完了次第メーリングリスト宛にメールで通知する。この際、指定された日付が翌日から1週間以上先、あるいは過去の日付である場合、登録処理がなされていないのに変更のメールが届いた場合にはエラーメールをユーザに対して送り、変更の処理は行われない。処理の概要を図10、処理完了メールの例を図11にそれぞれ示す。

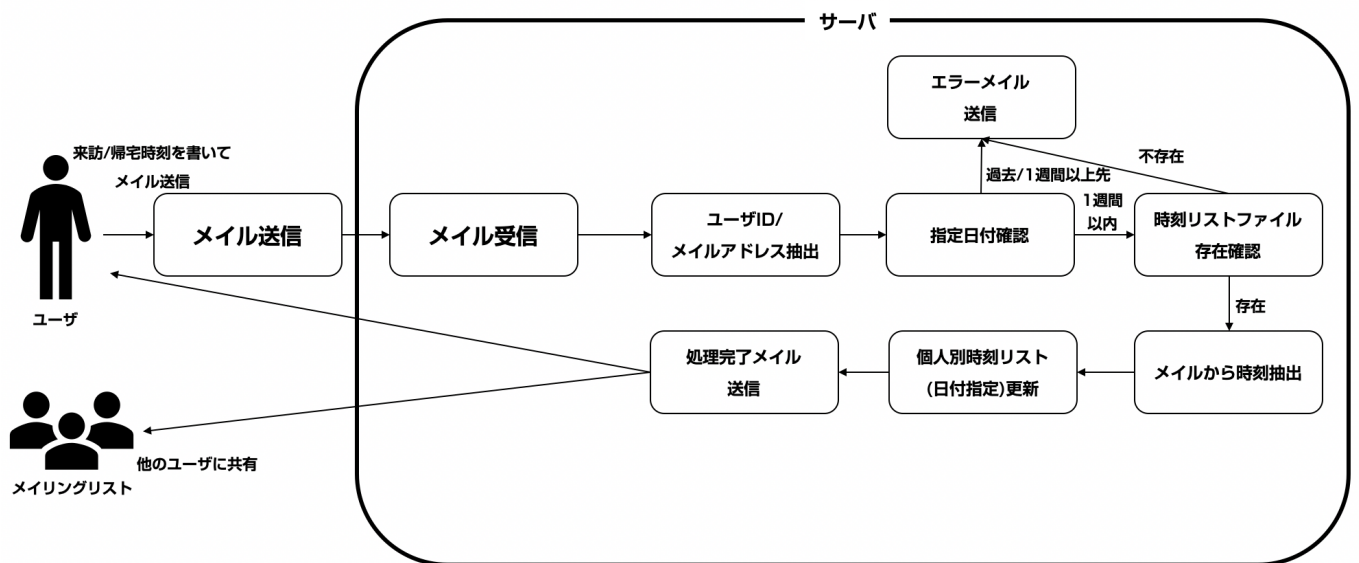


図10 翌日以降の研究室来訪/帰宅時刻の変更処理の概要。ユーザの送信したメールからメールアドレスと時刻、日付を抽出し、指定された日付の個別時刻リスト存在確認を行い、存在しない場合はエラーとする。正常に処理が完了すればメーリングリスト宛に処理完了メールを、指定された日付が1週間以上先、または過去の日付の場合、指定された日付の個別時刻リストファイルが存在しない場合にはユーザ宛にエラーメールを送信する。エラーの場合は変更処理を行わない。

[redacted].a4w.ise.osaka-sandai.ac.jp

To [redacted] ▼

<<この時刻は2018/12/18分です>>

s15h038 さんの2018/12/18の来る/帰る予定として送られた時刻が変更されました。

----->&----->&----->&----->&----->&-----

来る予定の時刻:09:15

帰る予定の時刻:21:00

----->&----->&----->&----->&----->&-----

帰る予定の時刻の変更は、[redacted].a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「E20181218」、本文に新しい時刻を書いて送ってください。

キャンセルする場合は、[redacted].a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「D20181218」と書いて空メールを送って下さい。
(大文字と小文字は区別されません。[redacted])

本メールは自動で作成されたメールです。返信しないでください。

図 11 翌日以降の研究室来訪/帰宅時刻変更処理完了メールの例。変更処理が行われたユーザ名、指定された日付と変更された時刻、本人向けに時刻の変更・削除の方法が書かれている。

3.3.9 翌日以降の研究室来訪/帰宅時刻の取り消し

翌日以降の研究室を訪れる時刻を送っていても、都合がつかず研究室に行くことができないことが前日までの段階で判明している場合はメールの Subject:を“D[数字 8 桁]”として日付を指定し、空メールを送ることで翌日以降の時刻の取り消しを行うことができる。本処理は送られてきたメールのヘッダを読み取り、From:のメールアドレスとユーザ名、Subject:に含まれる日付を抜き出し、指定された日付の該当ユーザのファイルを削除する。処理完了次第メイリングリスト宛にメールで通知する。この場合も翌日以降の時刻変更処理と同様に、指定された日付が翌日から1週間以上先、あるいは過去の日付である場合、登録処理がなされていないのに変更のメールが届いた場合にはエラーメールをユーザに対して送り、削除の処理は行われない。処理の概要を図12、処理完了メールの例を図13にそれぞれ示す。

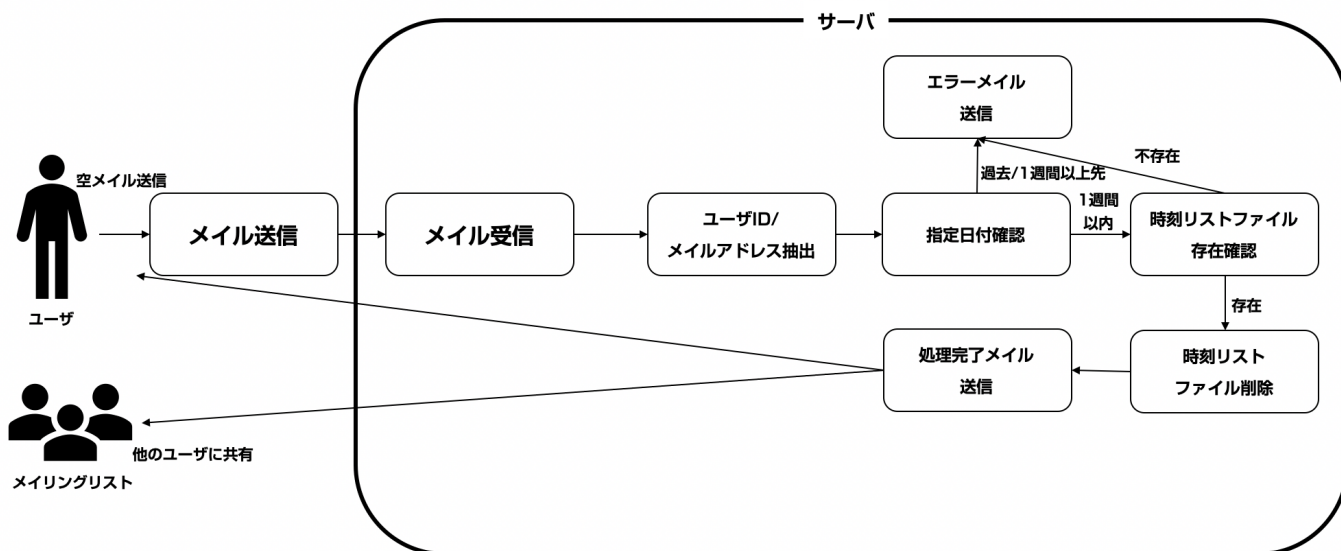


図12 翌日以降の研究室来訪/帰宅時刻の取り消し処理の概要。ユーザの送信したメールからメールアドレスと日付を抽出し、指定された日付の個別時刻リスト存在確認を行い、存在しない場合はエラーとする。正常に処理が完了すればメイリングリスト宛に処理完了メールを、指定された日付が1週間以上先、または過去の日付の場合、指定された日付の個別時刻リストファイルが存在しない場合にはユーザ宛にエラーメールを送信する。エラーの場合は取り消し処理を行わない。

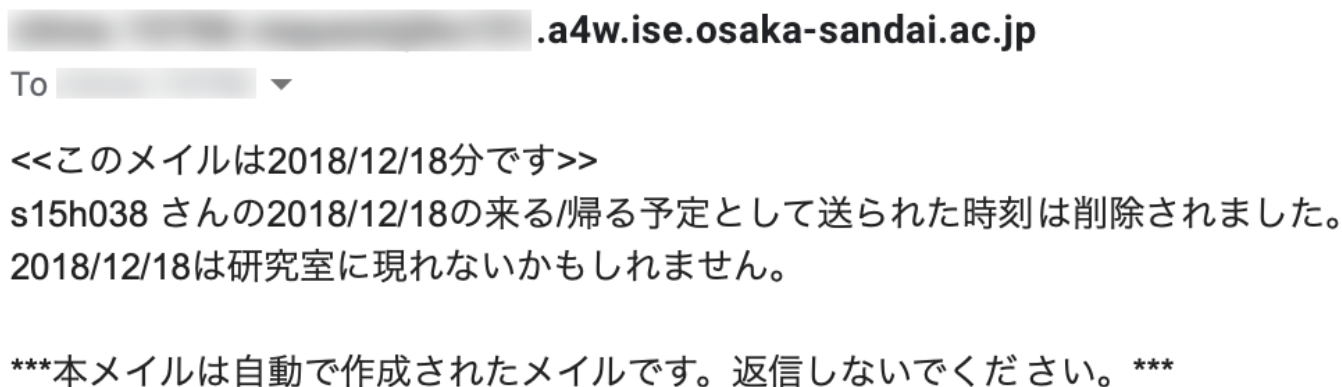


図13 翌日以降の研究室来訪/帰宅時刻取り消し処理完了メールの例。取り消し処理が行われたユーザ名と指定された日付、指定された日は研究室を訪れない可能性がある旨が書かれている。

3.3.10 登録済みの研究室来訪/帰宅時刻の参照

時刻を送っていても、ユーザ自身が忘れてしまったり、送信済みの時刻を確認したくなった場合にはメールの Subject:を “R*²⁴” として空メールを送ることで登録済みの時刻を確認することができる。翌日以降の時刻を確認する場合は Subject:を “R[日付 8 桁]” とし、同様に空メールを送ることで確認が可能となる。本処理は送られてきたメールのヘッダを読み取り、From:のメールアドレスとユーザ名を、日付指定がある場合は、Subject:に含まれる日付を抜き出し、ファイルに含まれる時刻を確認し、処理完了次第ユーザ宛にメールで通知する。この際、登録処理がなされていないのに参照のメールが届いた場合、日付指定があった場合は指定された日付が翌日から 1 週間以上先、あるいは過去の日付である場合にはエラーメールをユーザに対して送り、参照の処理は行われない。

処理の概要を図 14、処理完了メールの例を図 15、日付を指定した場合の処理の概要を図 16、日付を指定した場合の処理完了メールの例を図 17 にそれぞれ示す。

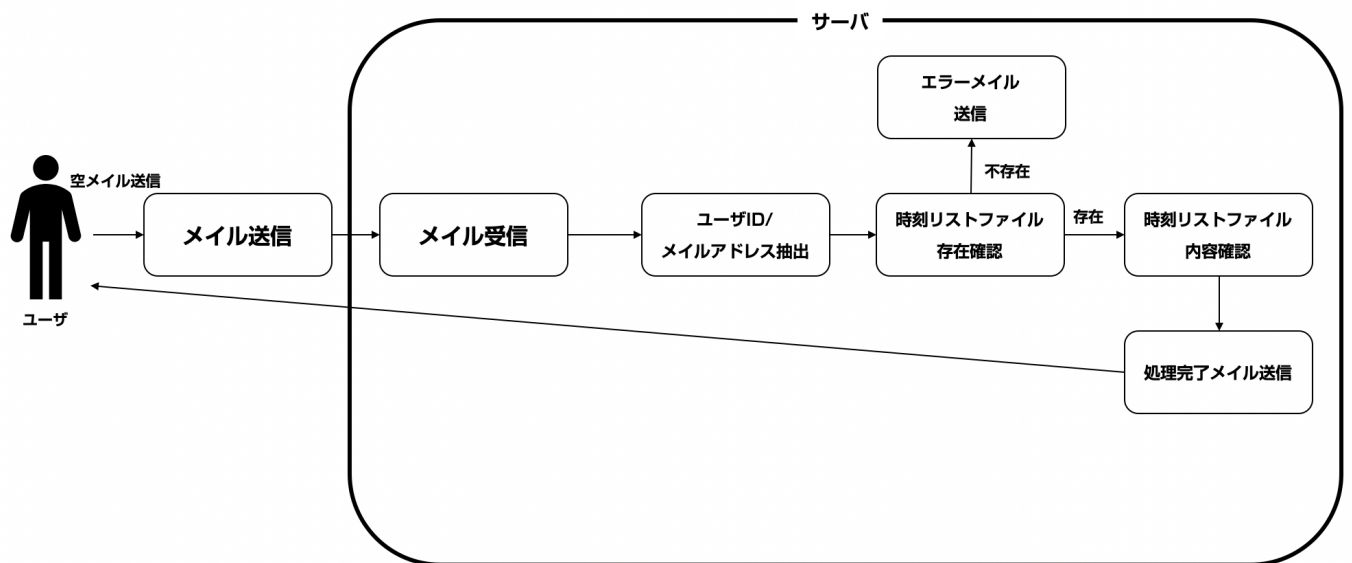


図 14 研究室来訪/帰宅時刻参照処理の概要。ユーザの送信したメールからメールアドレスを抽出、個別時刻リストを参照し、正常に処理が完了すればユーザ宛に処理完了メールを、ファイルが存在しない場合にはエラーメールを送信する。エラーの場合は参照処理を行わない。

*²⁴ Reference の頭文字。

.a4w.ise.osaka-sandai.ac.jp

To 自分 ▼

s15h038 さんの送られた時刻をお知らせします。

----->&----->&----->&----->&----->&-----

来る予定の時刻:15:25

帰る予定の時刻:18:40

----->&----->&----->&----->&----->&-----

帰る予定の時刻の変更は、.a4w.ise.osaka-sandai.ac.jp 宛に Subjectに「E」、本文に新しい時刻を書いて送ってください。

キャンセルする場合は、.a4w.ise.osaka-sandai.ac.jp 宛に Subjectに「D」と書いて空メールを送って下さい。
(大文字と小文字は区別されません。)

本メールは自動で作成されたメールです。返信しないでください。

図 15 研究室来訪/帰宅時刻参照処理完了メールの例。登録された時刻と時刻の変更・削除の方法が書かれている。

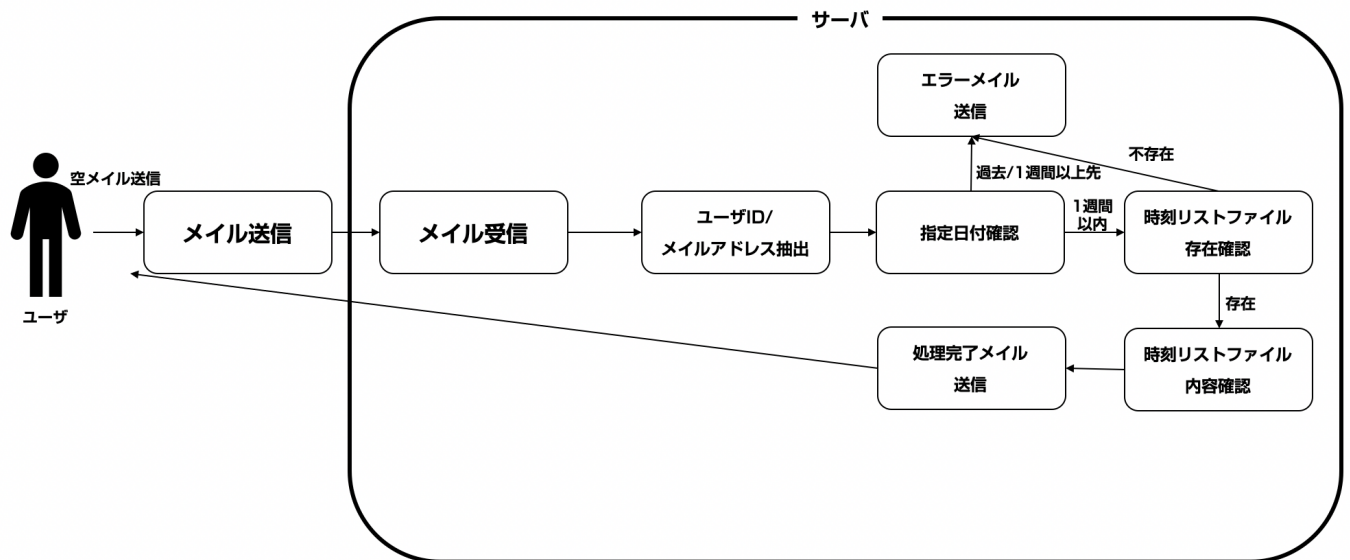


図 16 翌日以降の研究室来訪/帰宅時刻参照処理の概要。ユーザの送信したメールからメールアドレスと日付を抽出、指定した日付の個別時刻リストを参照し、正常に処理が完了すればユーザ宛に処理完了メールを、指定した日付のファイルが存在しない場合にはエラーメールを送信する。エラーの場合は参照処理を行わない。

████████████████████.a4w.ise.osaka-sandai.ac.jp

To 自分 ▼

<<この時刻は2018/12/18分です>>

s15h038 さんの2018/12/18の来る/帰る予定として送られた時刻をお知らせします。

----->&----->&----->&----->&----->&-----

来る予定の時刻:09:00

帰る予定の時刻:21:00

----->&----->&----->&----->&----->&-----

帰る予定の時刻の変更は、████████████████████.a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「E20181218」、本文に新しい時刻を書いて送ってください。

キャンセルする場合は、████████████████████.a4w.ise.osaka-sandai.ac.jp 宛に
Subjectに「D20181218」と書いて空メールを送って下さい。
(大文字と小文字は区別されません。████████████████████)

本メールは自動で作成されたメールです。返信しないでください。

図 17 翌日以降の研究室来訪/帰宅時刻参照処理完了メールの例。指定した日付の登録された時刻と時刻の変更・削除の方法が書かれている。

3.3.11 エラー処理

ユーザが研究室来訪/帰宅時刻を書いて送ったメールをシステムが受け取った際に、日付の指定に誤りがある、締切として設定された時刻を過ぎてしまっている、個別時刻リストファイルが存在しないのに変更や削除の処理を求めるコマンドが Subject:に書かれている場合などにユーザにエラーメールを自動送信する。登録や変更の処理を行うスクリプトにおいてエラー検知を行い、エラーと判断された場合はエラーメール送信処理用スクリプトが起動され、その際の引数としてエラーコード*²⁵とメールから抜き出した From:のメールアドレスを指定する。case 文によりエラーコードが読み取られ、その状況に応じたメッセージが表示用変数に代入される。続いて、現在時刻を取得し、エラーメッセージとともに Subject:に設定される。送信先にはユーザのメールアドレスを指定し、メールを送信する。処理の概要を図 18 に示す。

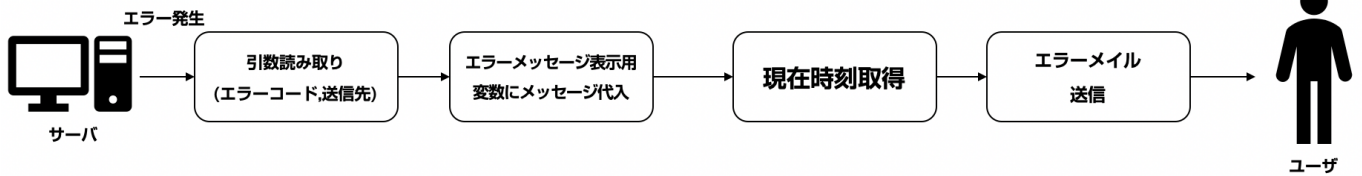


図 18 エラー処理の概要。エラーが発生したら各種時刻の処理を行うスクリプトからエラーコードを第一引数、From:のメールアドレスを第二引数として、エラー処理用スクリプトが起動する。エラーコードを基に該当のエラーを判別し、エラーメッセージ表示用変数にエラーメッセージを代入し、エラーが発生した現在の時刻を取得し、第二引数のメールアドレスを To:に代入してエラーメールを送信する。

*²⁵ エラーの種類を判別するための文字列として一意のものが設定されている。

3.3.12 全体リスト生成

時刻の処理を行うリストは2種類存在する。ユーザが送ってきた時刻を書き込んでいる個別リストと、それら^{*26}から一意の時刻を取り出した全体リストである。個別リストの更新があったら、個別リストの内容を取り出し、個別リストの内容と全体リストの内容を比較する。変更がなければ処理を終え、変更があれば個別リストに書き込まれた時刻から一意のものだけ全体リストに書込を行う。なお、本処理を実行した際に個別リストが存在しなければ全体リストには空白文字を書き込む。本処理は個別リストを操作する各種スクリプトから自動的に呼び出される。処理の概要を図19に示す。

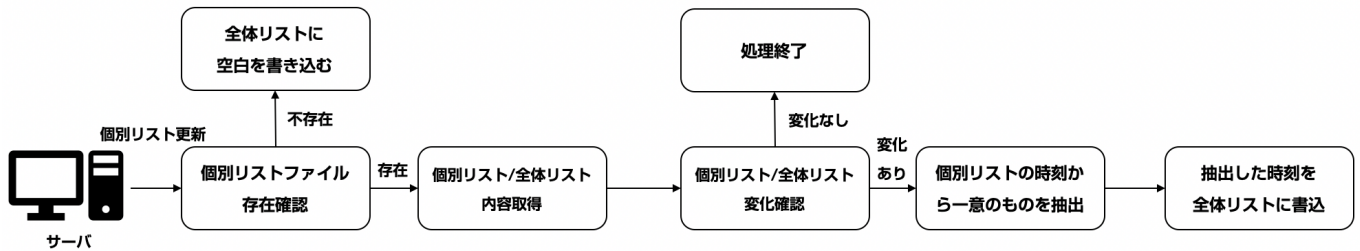


図19 全体リスト生成処理の概要。個別リストファイルの存在確認を行い、存在しない場合は全体リストに空白を書き込む。存在する場合は個別リストと全体リストの内容を確認し、個別リストの内容と全体リストの内容に変更がない場合は全体リストの更新を行わず、変更があった場合は個別リストに含まれる時刻から一意のものを抽出し全体リストに書き込みを行う。

^{*26} 複数人の個別リスト。

3.3.13 開閉時刻表示用 Web ページ自動更新

開閉時刻を表示する Web ページも自動的に更新を行う。まずは全体リストに含まれる時刻のうち、一番早い時刻^{*27}と一番遅い時刻^{*28}を取り出し、研究室の開閉時刻の変更確認を行う。この際に変更があれば変更を通知するメールをメイリングリスト宛に送信する。その後、表示中の時刻が書き込まれた一時ファイルと、全体リストから取得した時刻を比較し、変化がなければ処理を終える。変化があれば全体リストの内容を確認し、時刻の取得に失敗した場合はエラーメッセージを、取得成功の場合は時刻を表示用変数に代入し、施錠時刻が 21:45 JST 以降であれば退出催促メッセージ^{*29}を追加、施錠時刻が 21:50 JST 以降であれば施錠時刻を 21:50 JST に強制変更する。各種処理が終了した後、時刻が入った表示用変数を用いて表示用メッセージを生成し、開閉時刻表示用 Web ページを自動更新する。また、9:00 JST の更新の際は一時ファイルの変更の有無に関わらず強制的に更新を行う。その際にすでに時刻が全体リストにあった場合には時刻の通知メールをメイリングリスト宛に送信する。18:00 JST の更新の際も強制的に更新を行い、施錠時刻の通知メールをメイリングリスト宛に送信する。処理の概要を図 20 に、開閉時刻表示用 Web ページの正常な表示例を図 21 に、開閉時刻表示用 Web ページのエラー発生時表示例^{*30}を図 22 に、9:00 JST の時刻通知メールの例を図 23 に、時刻の変更通知メールの例を図 24、図 25 に、18:00 JST の通知メールの例 3 種類を図 26、図 27、図 28 にそれぞれ示す。

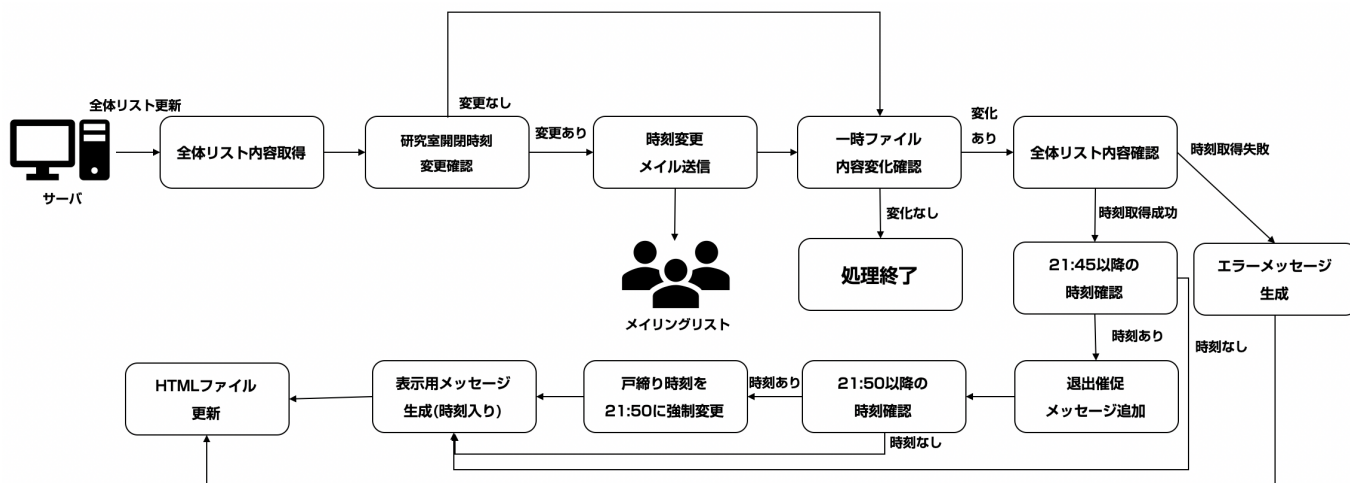


図 20 開閉時刻表示用 Web ページ自動更新処理の概要。全体リストの更新があれば、内容を取得し、研究室の開閉時刻に変更があればメイリングリスト宛に通知メールを送信する。変更があった場合にのみメールが送信され、変更がない場合にはメールは送信されない。次に現在の開閉時刻が格納されている一時ファイルと内容の差異を確認し、差異がなければ処理終了、差異があれば一時ファイルを更新する。続いて全体リストの内容を確認し、時刻の記録がなければエラーメッセージを生成し、時刻が記録されていれば 21:45 JST 以降の時刻の有無を確認する。あれば退出催促メッセージを追加する。さらに 21:50 JST 以降の時刻の有無を確認し、それ以降の時刻があれば施錠時刻を 21:50 JST に強制的に変更する。最後に表示用メッセージを生成し、開閉時刻表示用 Web ページを更新する。

*27 研究室の解錠予定時刻となる。

*28 研究室の施錠予定時刻となる。

*29 22:00 JST までに大学の敷地外へ出るといった規程に対応するためである。

*30 このエラーの際はエラーメールは送信されない。

情報教育システム研究室 2018/12/28 のオープン/戸締り時刻

研究室は 13:16 に開く予定です。

戸締り時刻は 16:00 の予定です。

Last updated 2018/12/28 15:05:29
Copyright© 2018 T-Koyabu.
[Go List.](#)

図 21 開閉時刻表示用 Web ページの正常な表示例。日付と開閉時刻、更新日時が表示されている。

情報教育システム研究室 2018/12/28 のオープン/戸締り時刻

オープン時刻取得失敗

戸締り時刻取得失敗

Last updated 2018/12/28 09:00:01
Copyright© 2018 T-Koyabu.
[Go List.](#)

図 22 開閉時刻表示用 Web ページのエラー発生時の表示例。日付と、開閉時刻取得失敗のメッセージ、更新日時が表示されている。

.....a4w.ise.osaka-sandai.ac.jp

To

研究室の開く予定/戸締り予定時刻についてお知らせします。

----->&----->&----->&-----

本日午前9時現在での研究室が開く予定/戸締り予定時刻をお知らせします。

本日の開く予定時刻は12:10、戸締り予定時刻は18:20です。(あくまで予定ですので、変わることがあります。)

----->&----->&----->&-----

本メールは自動で作成されたメールです。返信しないでください。

図 23 9:00 JST の段階で時刻が設定されている場合のメールの例。当日午前 9 時現在でシステムに登録のある開閉予定時刻が記されている。

.....a4w.ise.osaka-sandai.ac.jp

To

研究室の開く予定/戸締り予定時刻についてお知らせします。

----->&----->&----->&-----

本日の研究室が開く予定/戸締り予定時刻が設定されました。

本日の開く予定時刻は15:25、戸締り予定時刻は18:40です。(あくまで予定ですので、変わることがあります。)

----->&----->&----->&-----

本メールは自動で作成されたメールです。返信しないでください。

図 24 時刻が設定された際の通知メールの例。メールを送信する前の時点で時刻の登録がなく、その日初めての登録であった場合に送られる。システムから割り出された開閉時刻が記されている。

.....a4w.ise.osaka-sandai.ac.jp

To ▼

研究室の開く予定/戸締り予定時刻についてお知らせします。

----->&----->&----->&-----

本日の研究室が開く予定/戸締り予定時刻が取り消されました。
本日は研究室は開かないかもしれません。

----->&----->&----->&-----

本メールは自動で作成されたメールです。返信しないでください。

図 25 時刻が取り消された際の通知メールの例。全体リストから時刻がすべて取り消された場合に送られる。
当日は研究室が開かない可能性がある旨が記されている。

.....a4w.ise.osaka-sandai.ac.jp

To ▼

情報教育システム研究室の戸締り予定時刻をお知らせします。

----->&----->&----->&----->&----->&-----

本日の戸締り予定時刻は 12:30 ですので
すでに閉まっている可能性があります。

----->&----->&----->&----->&----->&-----

返信は17時45分で締め切りました。

本メールは自動で作成されたメールです。返信しないでください。

図 26 18:00 JST より前の時刻に施錠時刻が設定されている際の通知メールの例。施錠予定時刻と、時刻の登録/変更/削除処理は 17:45 JST にて締切、すでに閉まっている可能性がある旨が記されている。

.....a4w.ise.osaka-sandai.ac.jp

To ▼

情報教育システム研究室の戸締り予定時刻をお知らせします。

----->&----->&----->&----->&----->&-----

本日の戸締り予定時刻は 20:30 です。

----->&----->&----->&----->&----->&-----

返信は17時45分で締め切りました。

本メールは自動で作成されたメールです。返信しないでください。

図 27 18:00 JST より後の時刻に施錠時刻が設定されている際の通知メールの例。施錠予定時刻と、時刻の登録/変更/削除処理は 17:45 JST にて締め切った旨が記されている。

.....a4w.ise.osaka-sandai.ac.jp

To ▼

情報教育システム研究室の戸締り予定時刻をお知らせします。

----->&----->&----->&----->&----->&-----

本日の戸締り予定時刻はメールが一通も来なかったか、取得に失敗しているため、表示されません。

----->&----->&----->&----->&----->&-----

返信は17時45分で締め切りました。

本メールは自動で作成されたメールです。返信しないでください。

図 28 時刻の設定がなかった際の通知メールの例。一通もメールが届かなかったか、取得失敗の旨と、時刻の登録/変更/削除処理は 17:45 JST にて締め切った旨が記されている。

3.3.14 当日に登録のあった研究室来訪/帰宅時刻の集計

毎日 23:55 JST にその日送られてきた研究室来訪/帰宅時刻を集計し、メールで送信している。はじめに全体時刻リストの存在確認を行い、ファイルの内容が空もしくはファイルが存在しない場合、エラーとしてエラーメールをメイリングリスト宛に送り、処理を終える。ファイルの内容が空でない場合は来訪時刻の時間帯別^{*31}件数と個別リストの件数を集計、標準出力に出力し、同じ内容をテキストファイルに出力する。次に帰宅時刻の時間帯別件数と個別リストの件数を集計し、標準出力に出力し、同じ内容をテキストファイルに出力する。最後にメールの本文に出力結果を表示し、メイリングリスト宛にメールを送信する。処理の概要を図 29 に、集計メールの例を図 30 に、エラーメールを図 31 にそれぞれ示す。

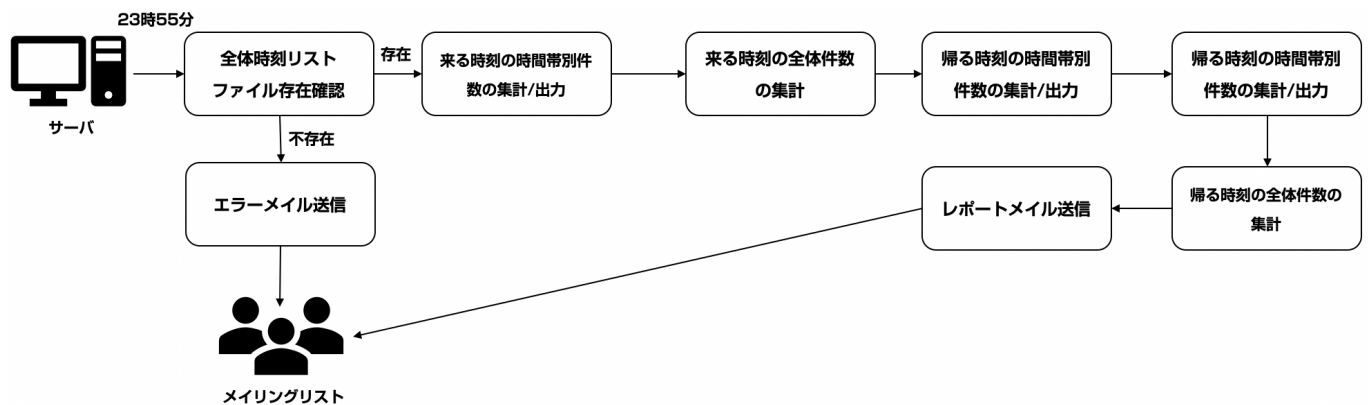


図 29 研究室来訪/帰宅時刻集計処理の概要。まず全体時刻リストの存在確認を行い、存在しない場合はエラーメールを送信して処理を終える。存在する場合は来訪時刻と帰宅時刻の時間帯別件数と個別リストの件数をそれぞれ集計し、結果を標準出力に出力、同じ内容をファイルにも出力する。最後にメールの本文に出力結果を表示し、メイリングリスト宛にメールを送信する。

^{*31} 7:00 JST から 21:59 JST の間のみ。

.a4w.ise.osaka-sandai.ac.jp

To

Lab Come/Go Home Time Reports.

----->&----->&----->&-----

来る予定の時刻

07時台:0人

08時台:0人

09時台:0人

10時台:0人

11時台:0人

12時台:1人

13時台:0人

14時台:1人

15時台:0人

16時台:0人

17時台:0人

18時台:0人

19時台:0人

20時台:0人

21時台:0人

----->&----->&----->&-----

今日メールを送ってきたのは2人

----->&----->&----->&-----

帰る予定の時刻

07時台:0人

08時台:0人

09時台:0人

10時台:0人

11時台:0人

12時台:0人

13時台:0人

14時台:0人

15時台:0人

16時台:0人

17時台:1人

18時台:1人

19時台:0人

20時台:0人

21時台:0人

----->&----->&----->&-----

今日メールを送ってきたのは2人

このメールは、本日送られてきたメールに記載されていた時刻の統計をお知らせするものです。

本メールは自動で作成されたメールです。返信しないでください。

図 30 研究室来訪/帰宅時刻集計メールの例。来訪時刻と帰宅時刻に分かれて時間帯別件数と個別リストの件数が書かれている。

[redacted].a4w.ise.osaka-sandai.ac.jp

To [redacted] ▼

Lab Come/Go Home Time Reports.

----->&----->&----->&-----

本日のReportはメールが一通も来なかったか、取得に失敗しているため、表示されません。

----->&----->&----->&-----

このメールは、本日送られてきたメールに記載されていた時刻の統計をお知らせするものです。

本メールは自動で作成されたメールです。返信しないでください。

図 31 研究室来訪/帰宅時刻集計ができなかった場合のメールの例。一通も登録のメールが届かなかったか、全体リストから時刻が取得できなかったため表示されない旨が書かれている。

3.3.15 日付変更時の処理

日付が変わった際に、前日に作成された時刻リストやログファイル^{*32}をアーカイブ^{*33}用のディレクトリに移動し、当日に送られてくる時刻を受け付けられるように準備を行う。はじめに前日分の時刻リストとログファイルをアーカイブ用のディレクトリに移動し、前日分の一時ファイルを削除する。続いて、当日受付分の全体時刻リストを作成し、前日までに日付を指定して送信された時刻リストのうち、当日の日付のものを当日用個別時刻リストにファイル名を変更する。次に、当日分のログファイルを自動生成し、前日分の時刻リストとログファイルのオーナーを変更する^{*34}。続いて、当日分全体時刻リストのパーミッション^{*35}を変更し、当日分の全体時刻リストとログファイルを初期化し、開閉時刻表示用 Web ページと、来訪/帰宅時刻一覧表示用 Web ページを自動更新する。処理の概要を図 32 に、開閉時刻表示用 Web ページの表示例を図 33 にそれぞれ示す。

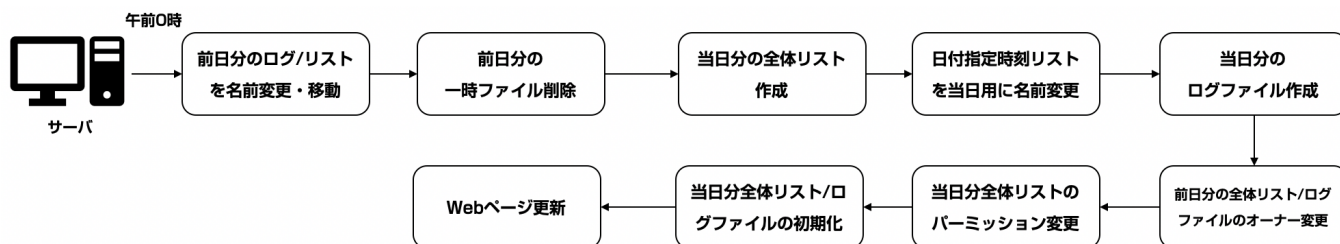


図 32 日付変更時の処理の概要。前日分のログ/リストファイルのファイル名を変更し、アーカイブ用ディレクトリに移動し、前日分の一時ファイルを削除する。次に当日分の全体リストを作成し、前日までに日付を指定して送られた時刻が含まれる個別時刻リストファイルを当日用個別時刻リストファイルのファイル名に変更する。続いて当日用のログファイルを作成し、当日分のログ/リストファイルの所有者を一般ユーザに変更する。最後に当日分の全体リストの権限を変更し、当日分全体リストとログファイルの初期化を行い、開閉時刻表示用 Web ページを更新する。

情報教育システム研究室 2018/12/28 のオープン/戸締り時刻

毎日 9:00 JST に最初の更新を予定

Last updated 2018/12/28 00:00:02
Copyright© 2018 T-Koyabu.

図 33 日付変更時処理後の開閉時刻表示用 Web ページの表示例。日付と、毎日 9:00 JST にその日最初の更新を行う旨と、更新日時が表示されている。なお、来訪/帰宅時刻一覧表示用 Web ページにも同じ内容が表示されている。

*32 各種処理を実行した際に逐次ログを記録している。

*33 この場合は過去の時刻リストやログを保存すること。

*34 root 権限で処理がされるため、一般ユーザでも確認が可能となるようにオーナーを変更する。

*35 ファイルの権限設定。

4 関連研究 (人々の行動を手軽に共有する生活空間エージェント [1])

4.1 関連研究の概要

研究室に所属する学生の行き先情報をコンピュータを組み込んだ専用掲示板を用いて研究室に所属する学生および教員の行き先の共有を行うためのシステムを開発している。これは専用掲示板に設置されている RFID^{*36}リーダに FeliCa^{*37}方式の ID カードをかざすことで行き先の選択が可能になり、選択を行うことでシステムに記録され、専用掲示板の表示内容が更新されるというものである。さらに、システムに記録されている情報を Twitter に投稿するという機能も備えている。

4.2 関連研究のメリット

ユーザである学生および教員が専用掲示板に設置されている RFID リーダに ID カードをかざすだけでユーザ認証が行われ、行き先の選択を行うことで専用掲示板ならびに選択内容が専用 Twitter アカウントにて共有が行われる。これにより専用掲示板の前まで見に行かなくとも状態が Twitter 経由で確認が可能となる。

4.3 関連研究の問題点

システムには ID と行き先、情報が登録された時刻に加え、実名が記録されており、実名と時刻は Twitter にも投稿されてしまっている。

4.4 関連研究の問題点を本研究でどう解決するか

本研究で開発するシステムの特徴はユーザ ID^{*38}と時刻、メールアドレスは記録するものの、実名は記録されない。ユーザから送られてきた時刻は Web でも確認可能ではあるものの、ユーザ ID から生成された学籍番号と時刻しか表示せず、ユーザが懸念するプライバシーの保護も実現が可能となる。

*36 Radio Frequency Identifier

*37 ソニー株式会社が開発した非接触式 IC カードの技術方式。

*38 大阪産業大学 (以下、本学とする。) の学生ユーザのユーザ ID。s[入学年度西暦下 2 桁][学科記号][個人番号] となっており、2015 年度入学の情報システム学科の学生の場合は s15h777 のような形式となる。学生に配付されるメールアドレスの左パートも同一の ID が使用されているため、個人の識別が可能となっている。

5 検証

2018年9月1日から同年12月28日^{*39}までの間、研究室への来訪時刻と帰宅時刻のメールによる報告を求め、集まったデータを基に検証を行った。

5.1 検証対象者

1. デザイン工学部情報システム学科4年生...11名
2. デザイン工学部情報システム学科3年生...3名
3. デザイン工学部情報システム学科2年生...1名

検証の対象者は15名^{*40}、そのうちシステムで時刻受信実績がある参加者はデザイン工学部情報システム学科4年生の7名である。

5.2 検証方法

5.2.1 送信時の書式

時刻をメールに書く際に、システム側で認識するために書式の指定を行った。メールの本文に00:00^{*41}といった形式で時刻を書いて送るように案内のメールに記載し、時刻の登録を実施してもらった。

5.2.2 概要

検証の対象者に月曜日から土曜日^{*42}の7:30 JST と 12:30 JST に案内のメールをシステムから送り、来訪時刻と帰宅時刻を書いて返信するよう求めた。

5.2.3 実際の開閉時刻の取得方法

実際の開閉時刻は、情報教育システム研究室において既に運用されている研究室内照明明滅検知システムを利用して取得を行った。

5.2.4 研究室内照明明滅検知システム

情報教育システム研究室には複数台カメラが設置されているが、そのうち1台は、研究室内の照明の明滅を検知し、検知時刻ならびにその時撮影された写真とともに、研究室内向けメイリングリスト(以下、team.andrew-MLとする。)にメールを送るシステムに繋がっている。そのメールに書かれている時刻のうち、1日で一番早い時刻を実際の解錠時刻、一番遅い時刻を施錠時刻とした。

^{*39} 2018年12月28日から同年12月31日までの間は、本学の年末年始一斉休業期間のため、本システムを停止していた。そのため、検証期間外としている。ただし、2018年12月28日は臨時で研究室が開いていたため、システムを動作させていた。そのため、検証期間としている。

^{*40} 対象者は全員、情報教育システム研究室の関係者である。

^{*41} 0:00 JST から 9:59 JST までの場合は頭に0をつけないと認識されない。

^{*42} 研究開始当初は日曜日に研究室が解錠されることを想定していなかったため、月曜日から土曜日までの間だけ、各種メールの送信、来訪/帰宅時刻の受付を行っていたが、2018年12月17日より毎日、各種メールの送信、来訪/帰宅時刻の報告を受け付けるように変更した。

5.3 検証結果

今回結果をまとめるにあたって、開閉時刻可視化システムに登録されている時刻のうち、2018年9月から同年12月に記録された時刻とその件数を用いる。

5.3.1 2018年9月の検証結果

2018年9月のうち半分は夏季休業期間^{*43}であったため、学生が研究室を訪れる日数は少なかった。それに比例して、時刻の登録があった日数は7日、平均件数は0.37件と非常に少なかった。研究室の解錠予定時刻と実際の解錠時刻との差異を図34に、施錠予定時刻と実際の施錠時刻の差異を図35に、集計したデータを表2にそれぞれ示す。

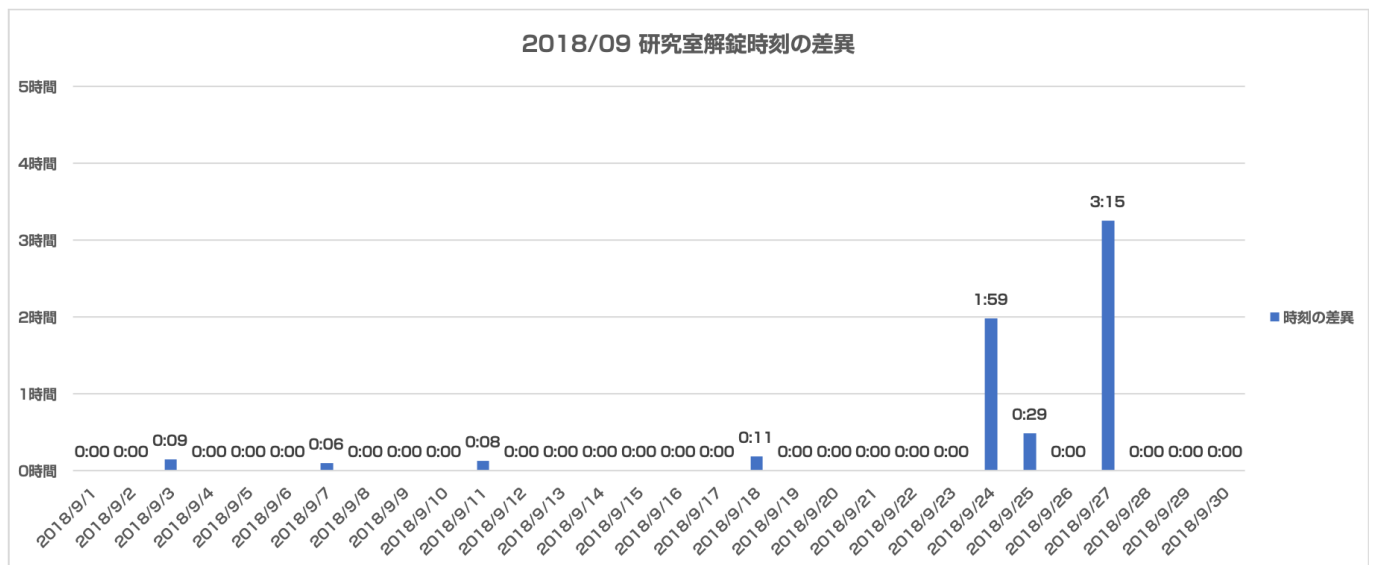


図34 2018年9月の解錠予定時刻と実際の解錠時刻との差異のグラフ。棒グラフが差異を表す。時間が0:00の部分は差異がないか、データがないことを表す。

*43 2018年9月20日まで夏季休業期間。

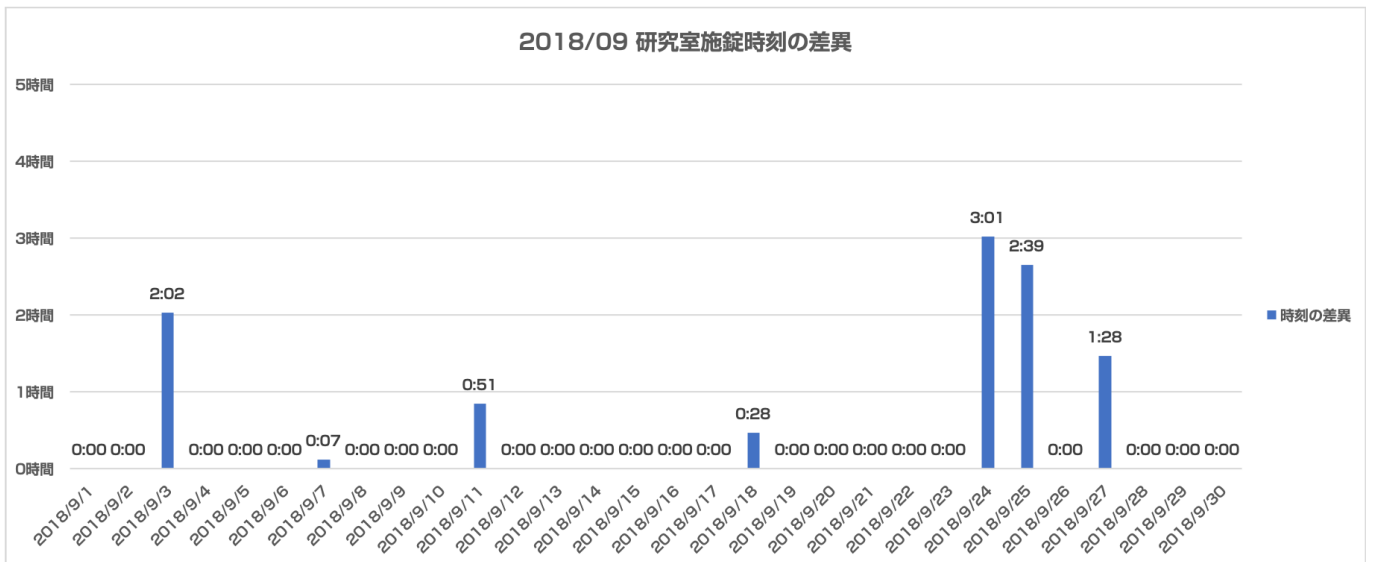


図 35 2018 年 9 月の施錠予定時刻と実際の施錠時刻との差異のグラフ。棒グラフが差異を表す。時間が 0:00 の部分は差異がないが、データがないことを表す。

表2 2018年9月の開閉時刻の表。システムで受信したメールの件数、システムに記録されている開閉時刻と、実際の開閉時刻、その差異がそれぞれ示されている。空白はデータがないことを表す。システムに記録された時刻もしくは実際の開閉時刻がない場合は比較不可としている。時刻はすべてJSTである。

日付	件数	解錠時刻	実際の解錠時刻	差異	施錠時刻	実際の施錠時刻	差異
2018/9/1	0		11:06	比較不可		11:12	比較不可
2018/9/2	0						
2018/9/3	1	11:00	11:09	0:09	18:00	20:02	2:02
2018/9/4	0						
2018/9/5	0		9:17	比較不可		19:55	比較不可
2018/9/6	0		11:03	比較不可		17:52	比較不可
2018/9/7	2	10:30	10:36	0:06	18:00	17:53	0:07
2018/9/8	0		8:50	比較不可		18:12	比較不可
2018/9/9	0						
2018/9/10	0		9:40	比較不可		18:23	比較不可
2018/9/11	2	9:00	8:52	0:08	18:00	18:51	0:51
2018/9/12	0		9:50	比較不可		18:22	比較不可
2018/9/13	0		9:01	比較不可		15:34	比較不可
2018/9/14	0		9:56	比較不可		16:49	比較不可
2018/9/15	0						
2018/9/16	0						
2018/9/17	0						
2018/9/18	2	8:45	8:34	0:11	20:30	20:58	0:28
2018/9/19	0		7:58	比較不可		19:37	比較不可
2018/9/20	0		9:43	比較不可		14:58	比較不可
2018/9/21	0		9:38	比較不可		18:55	比較不可
2018/9/22	0		9:22	比較不可		16:25	比較不可
2018/9/23	0						
2018/9/24	1	13:20	11:21	1:59	17:00	20:01	3:01
2018/9/25	2	9:00	8:31	0:29	18:00	20:39	2:39
2018/9/26	0		10:12	比較不可		18:38	比較不可
2018/9/27	1	15:00	11:45	3:15	17:40	19:08	1:28
2018/9/28	0		10:26	比較不可		18:59	比較不可
2018/9/29	0						
2018/9/30	0						
平均	0.37件	10時56分	9時50分		18時10分	18時09分	

5.3.2 2018年10月の検証結果

2018年10月は講義実施期間中であり、学生は研究や勉学に勤しむため、研究室を訪れる日数は多かった。それに比例して、時刻の登録があった日数は19日、平均件数は1.16件と2018年9月と比べて多くなった。研究室の解錠予定時刻と実際の解錠時刻との差異を図36に、施錠予定時刻と実際の施錠時刻の差異を図37に、集計したデータを表3にそれぞれ示す。

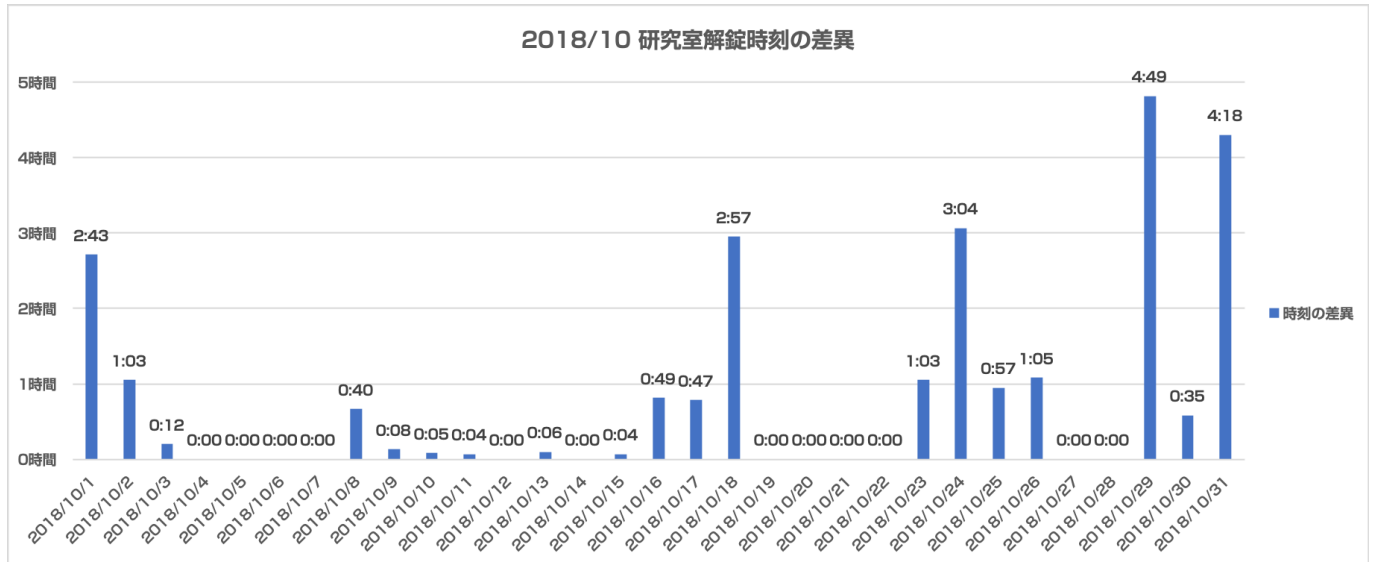


図36 2018年10月の解錠予定時刻と実際の解錠時刻との差異のグラフ。棒グラフが差異を表す。時間が0:00の部分は差異がないか、データがないことを表す。

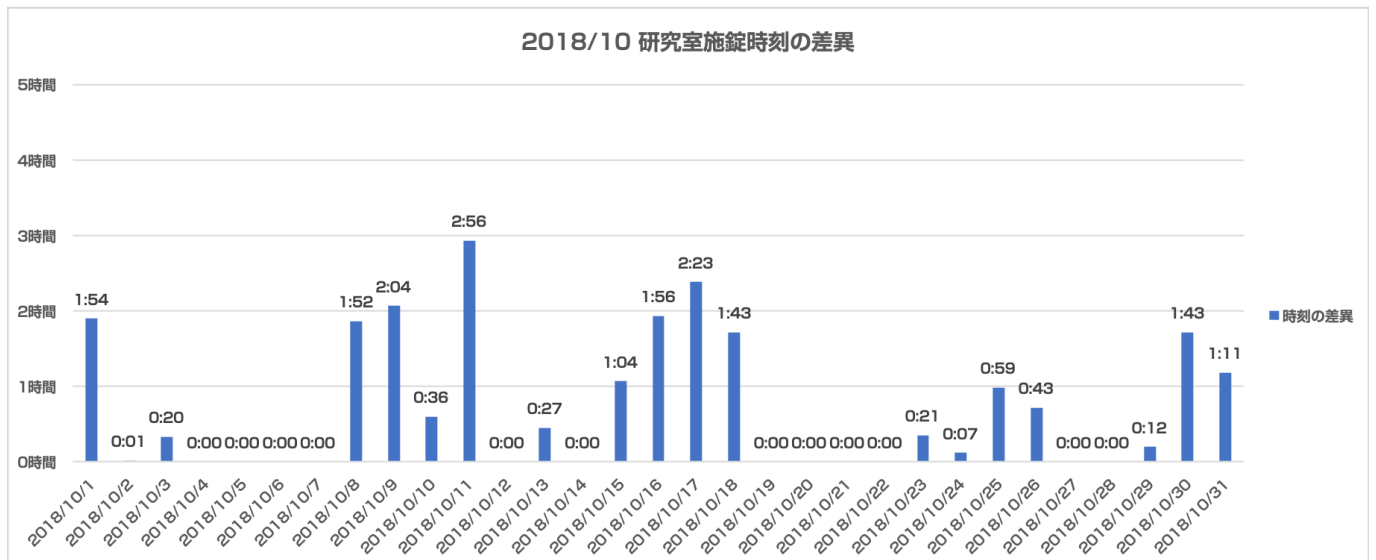


図37 2018年10月の施錠予定時刻と実際の施錠時刻との差異のグラフ。棒グラフが差異を表す。時間が0:00の部分は差異がないか、データがないことを表す。

表3 2018年10月の開閉時刻の表。システムで受信したメールの件数、システムに記録されている開閉時刻と、実際の開閉時刻、その差異がそれぞれ示されている。空白はデータがないことを表す。システムに記録された時刻もしくは実際の開閉時刻がない場合は比較不可としている。時刻はすべてJSTである。

日付	件数	解錠時刻	実際の解錠時刻	差異	施錠時刻	実際の施錠時刻	差異
2018/10/1	1	16:23	13:40	2:43	18:00	19:54	1:54
2018/10/2	3	9:00	7:57	1:03	19:00	18:59	0:01
2018/10/3	2	9:40	9:52	0:12	18:30	18:10	0:20
2018/10/4	0		8:19	比較不可		20:06	比較不可
2018/10/5	0		11:13	比較不可		19:24	比較不可
2018/10/6	0						
2018/10/7	0						
2018/10/8	1	14:30	13:50	0:40	17:00	18:52	1:52
2018/10/9	2	8:45	8:37	0:08	18:00	20:04	2:04
2018/10/10	3	10:00	10:05	0:05	18:00	18:36	0:36
2018/10/11	3	10:00	9:56	0:04	18:00	20:56	2:56
2018/10/12	0		10:11	比較不可		18:37	比較不可
2018/10/13	1	7:30	7:36	0:06	19:00	19:27	0:27
2018/10/14	0						
2018/10/15	1	10:00	10:04	0:04	17:30	18:34	1:04
2018/10/16	3	9:00	8:11	0:49	18:30	20:26	1:56
2018/10/17	2	9:00	9:47	0:47	18:20	20:43	2:23
2018/10/18	2	12:20	9:23	2:57	19:00	20:43	1:43
2018/10/19	0		10:33	比較不可		20:29	比較不可
2018/10/20	0						
2018/10/21	0						
2018/10/22	0		14:19	比較不可		18:00	比較不可
2018/10/23	3	9:00	7:57	1:03	21:00	21:21	0:21
2018/10/24	2	13:00	9:56	3:04	18:00	18:07	0:07
2018/10/25	1	10:00	10:57	0:57	19:00	19:59	0:59
2018/10/26	2	10:30	11:35	1:05	19:30	20:13	0:43
2018/10/27	0		14:33	比較不可		18:19	比較不可
2018/10/28	0						
2018/10/29	1	14:30	9:41	4:49	19:00	19:12	0:12
2018/10/30	1	9:00	8:25	0:35	18:00	19:43	1:43
2018/10/31	2	14:30	10:12	4:18	19:00	20:11	1:11
平均	1.16 件	10 時 52 分	10 時 16 分		18 時 32 分	19 時 33 分	

5.3.3 2018年11月の検証結果

2018年11月は講義実施期間中であり、月末に情報教育システム研究室の卒業論文初稿提出締切^{*44}が迫っていたため、卒業論文執筆に勤しむべく、学生が研究室を訪れる日数は多かった。しかし、時刻の登録があった日数は13日、平均件数は0.53件と2018年10月と比較して少なくなった。研究室の解錠予定時刻と実際の解錠時刻との差異を図38に、施錠予定時刻と実際の施錠時刻の差異を図39^{*45}に、集計したデータを表4にそれぞれ示す。

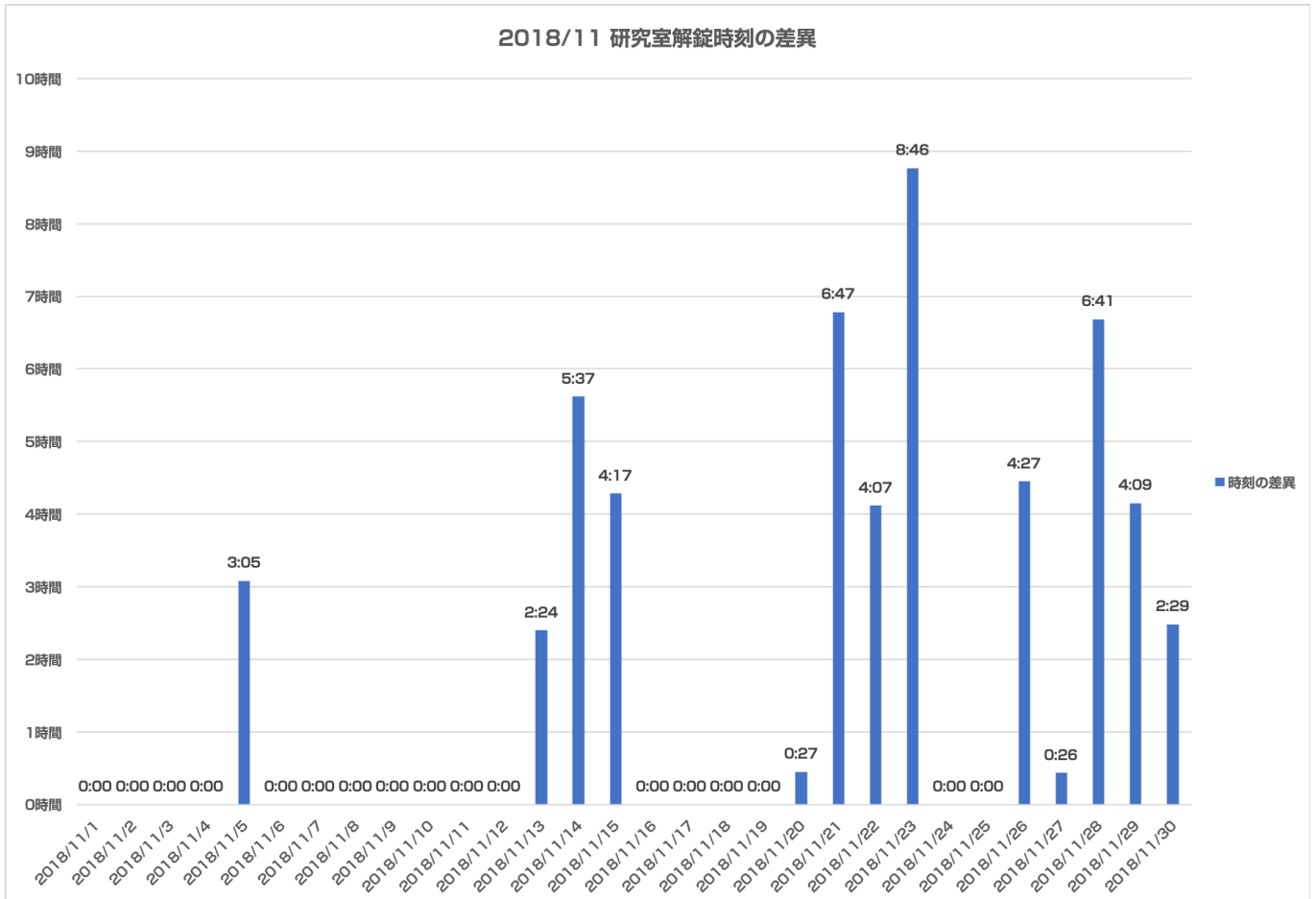


図 38 2018年11月の解錠予定時刻と実際の解錠時刻との差異のグラフ。棒グラフが差異を表す。時間が0:00の部分は差異がないか、データがないことを表す。

^{*44} 2018年11月30日の日没が締切である。

^{*45} 図38と図39のデータの最大値の違いにより、図のサイズを変えているが、軸のサイズは統一している。

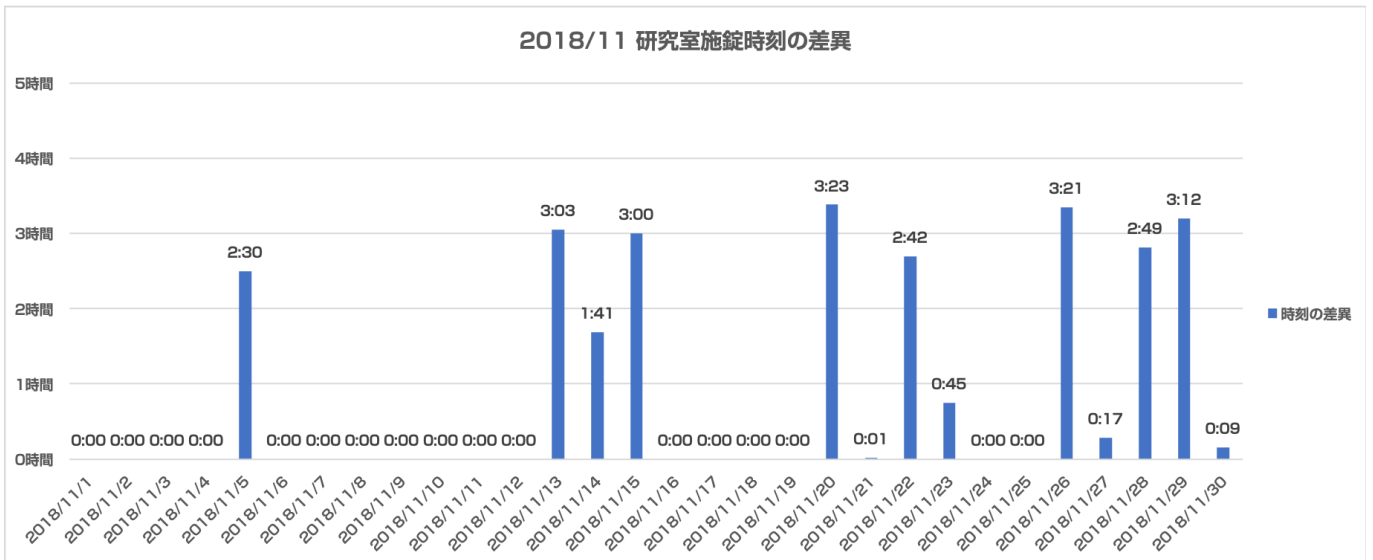


図 39 2018 年 11 月の施錠予定時刻と実際の施錠時刻との差異のグラフ。棒グラフが差異を表す。時間が 0:00 の部分は差異がないか、データがないことを表す。

表4 2018年11月の開閉時刻の表。システムで受信したメールの件数、システムに記録されている開閉時刻と、実際の開閉時刻、その差異がそれぞれ示されている。空白はデータがないことを表す。システムに記録された時刻もしくは実際の開閉時刻がない場合は比較不可としている。時刻はすべて JST である。

日付	件数	解錠時刻	実際の解錠時刻	差異	施錠時刻	実際の施錠時刻	差異
2018/11/1	0		9:50	比較不可		18:44	比較不可
2018/11/2	0						
2018/11/3	0		16:37	比較不可		20:53	比較不可
2018/11/4	0						
2018/11/5	1	17:10	14:05	3:05	18:30	21:00	2:30
2018/11/6	0		10:16	比較不可		17:49	比較不可
2018/11/7	0		18:44	比較不可		18:47	比較不可
2018/11/8	0		13:42	比較不可		20:05	比較不可
2018/11/9	0		9:34	比較不可		18:59	比較不可
2018/11/10	0		7:06	比較不可		18:13	比較不可
2018/11/11	0						
2018/11/12	0		8:56	比較不可		20:10	比較不可
2018/11/13	2	9:30	7:06	2:24	18:00	21:03	3:03
2018/11/14	1	14:30	8:53	5:37	18:20	20:01	1:41
2018/11/15	2	13:20	9:03	4:17	17:40	20:40	3:00
2018/11/16	0		12:15	比較不可		15:00	比較不可
2018/11/17	0		10:02	比較不可		16:03	比較不可
2018/11/18	0						
2018/11/19	0		8:57	比較不可		21:26	比較不可
2018/11/20	1	9:00	8:33	0:27	18:00	21:23	3:23
2018/11/21	1	14:30	7:43	6:47	19:30	19:31	0:01
2018/11/22	1	12:20	8:13	4:07	18:00	20:42	2:42
2018/11/23	1	17:00	8:14	8:46	18:10	18:55	0:45
2018/11/24	0						
2018/11/25	0						
2018/11/26	1	12:30	8:03	4:27	16:30	19:51	3:21
2018/11/27	2	9:00	8:34	0:26	21:00	21:17	0:17
2018/11/28	1	14:30	7:49	6:41	18:20	21:09	2:49
2018/11/29	1	12:10	8:01	4:09	18:00	21:12	3:12
2018/11/30	1	11:00	8:31	2:29	21:50	21:41	0:09
平均	0.53 件	12 時 48 分	9 時 56 分		18 時 36 分	19 時 46 分	

5.3.4 2018年12月の検証結果

2018年12月は講義実施期間中であり、下旬に情報教育システム研究室の卒業論文の一次完成締切^{*46}が迫っていたため、卒業論文執筆に勤しむべく、学生が研究室を訪れる日数は多かった。それに比例して、時刻の登録があった日数は18日、平均件数は0.81件と2018年11月と比較して多くなった。研究室の解錠予定時刻と実際の解錠時刻との差異を図40に、施錠予定時刻と実際の施錠時刻の差異を図41^{*47}に、集計したデータを表5にそれぞれ示す。

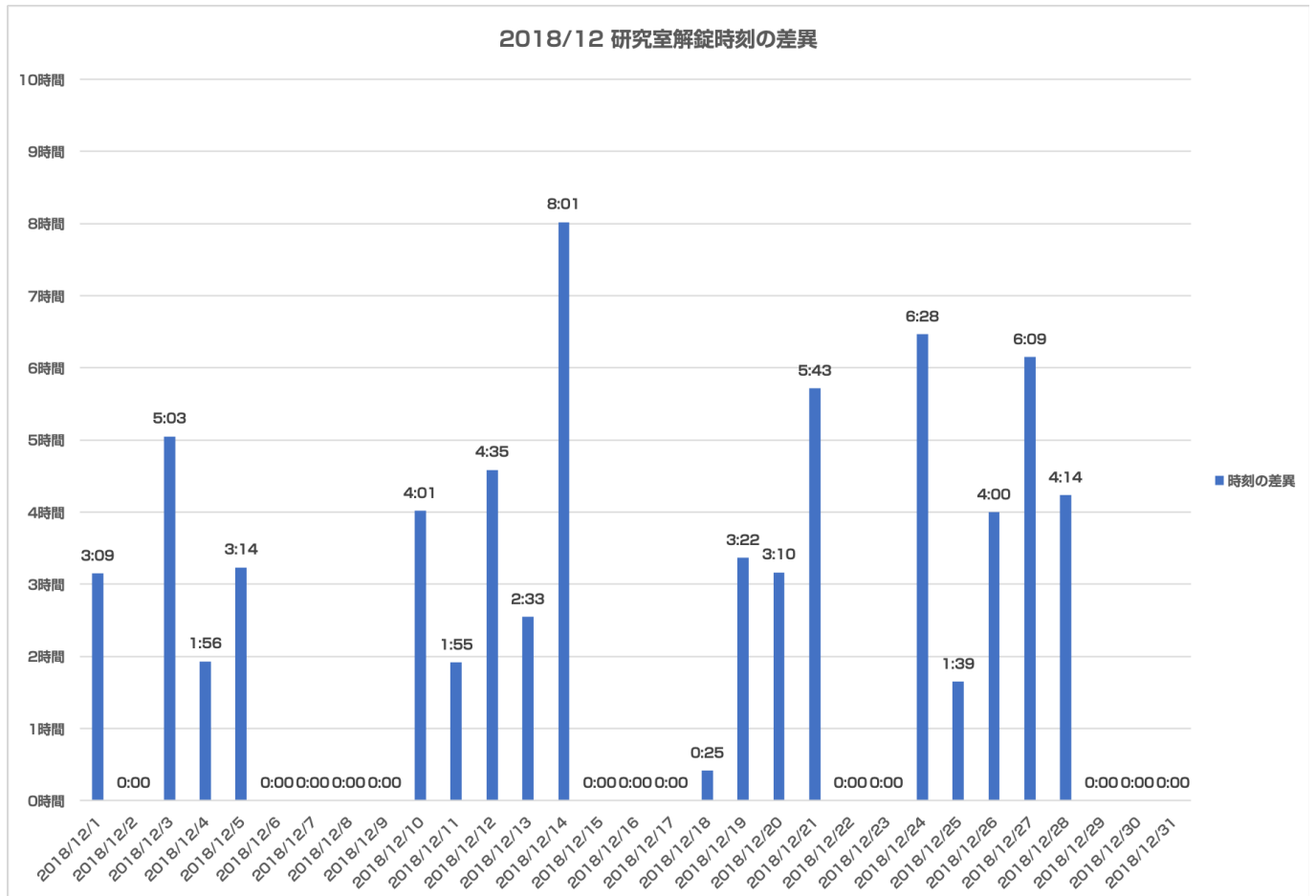


図40 2018年12月の解錠予定時刻と実際の解錠時刻との差異のグラフ。棒グラフが差異を表す。時間が0:00の部分は差異がないか、データがないことを表す。

*46 2018年12月28日が締切である。

*47 図40と図41のデータの最大値の違いにより、図のサイズを変えているが、軸のサイズは統一している。

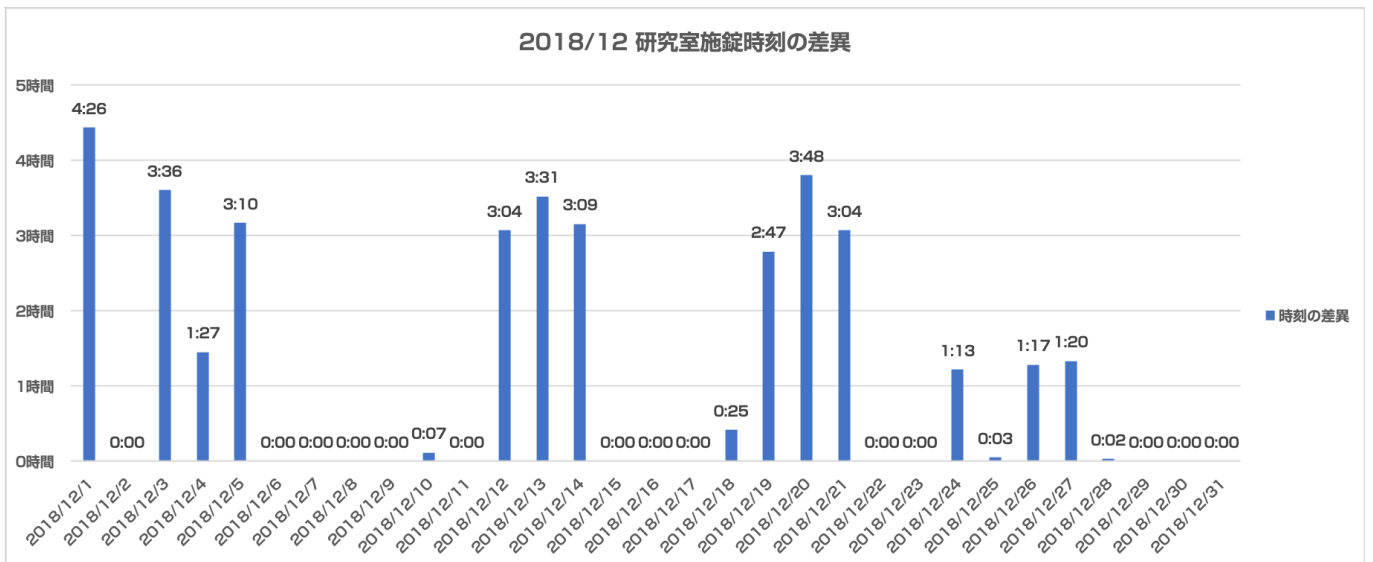


図 41 2018 年 12 月の施錠予定時刻と実際の施錠時刻との差異のグラフ。棒グラフが差異を表す。時間が 0:00 の部分は差異がないか、データがないことを表す。

表 5 2018 年 12 月の開閉時刻の表。システムで受信したメールの件数、システムに記録されている開閉時刻と、実際の開閉時刻、その差異がそれぞれ示されている。空白はデータがないことを表す。システムに記録された時刻もしくは実際の開閉時刻がない場合は比較不可としている。時刻はすべて JST である。

日付	件数	解錠時刻	実際の解錠時刻	差異	施錠時刻	実際の施錠時刻	差異
2018/12/1	1	13:54	10:45	3:09	15:40	20:06	4:26
2018/12/2	0						
2018/12/3	1	14:30	9:27	5:03	18:00	21:36	3:36
2018/12/4	2	8:58	7:02	1:56	20:30	21:57	1:27
2018/12/5	2	12:10	8:56	3:14	18:20	21:30	3:10
2018/12/6	0		9:05	比較不可		21:38	比較不可
2018/12/7	0		9:24	比較不可		20:25	比較不可
2018/12/8	0		8:46	比較不可		20:00	比較不可
2018/12/9	0						
2018/12/10	1	14:30	10:29	4:01	21:30	21:23	0:07
2018/12/11	1	9:00	7:05	1:55	21:20	21:20	0:00
2018/12/12	1	12:46	8:11	4:35	18:30	21:34	3:04
2018/12/13	1	12:10	9:37	2:33	18:20	21:51	3:31
2018/12/14	1	15:25	7:24	8:01	18:40	21:49	3:09
2018/12/15	0		9:19	比較不可		21:57	比較不可
2018/12/16	0						
2018/12/17	0		8:35	比較不可		20:15	比較不可
2018/12/18	1	9:00	8:35	0:25	21:00	21:25	0:25
2018/12/19	1	12:42	9:20	3:22	18:20	21:07	2:47
2018/12/20	1	12:10	9:00	3:10	18:00	21:48	3:48
2018/12/21	1	15:04	9:21	5:43	18:20	21:24	3:04
2018/12/22	0						
2018/12/23	0						
2018/12/24	2	14:30	8:02	6:28	19:00	20:13	1:13
2018/12/25	3	9:00	7:21	1:39	21:50	21:47	0:03
2018/12/26	2	13:07	9:07	4:00	20:00	21:17	1:17
2018/12/27	2	15:00	8:51	6:09	20:00	21:20	1:20
2018/12/28	1	13:16	9:02	4:14	16:40	16:38	0:02
2018/12/29	0						
2018/12/30	0						
2018/12/31	0						
平均	0.81 件	12 時 37 分	8 時 48 分		19 時 06 分	21 時 03 分	

6 結果と考察

2018年9月から12月にかけて、ユーザから報告のあった時刻を集計し、検証結果としてまとめた。次項に検証結果の考察を示す。

6.1 検証結果の考察

システムに記録された開閉時刻から算出された平均開閉時刻と、実際の平均開閉時刻を表6に示す。

表6 2018年9月から12月にシステムに記録された開閉時刻から割り出した平均開閉時刻と、実際の平均開閉時刻の表。検証を行った年月と、システムに記録された平均解錠時刻、平均施錠時刻、実際の平均解錠時刻、平均施錠時刻が月別で示されている。時刻はすべてJSTである。

検証を行った年月	システムに記録された時刻から算出した平均解錠時刻	実際の平均解錠時刻	システムに記録された平均施錠時刻	実際の平均施錠時刻
2018年9月	10時56分	9時50分	18時10分	18時09分
2018年10月	10時52分	10時16分	18時32分	19時33分
2018年11月	12時48分	9時56分	18時36分	19時46分
2018年12月	12時37分	8時48分	19時06分	21時03分

この結果から、情報教育システム研究室は夏季休業期間と講義実施期間では解錠時刻は10時台で安定しているが、施錠時刻は夏季休業期間より講義実施期間の方が遅くなることが判明した。また、システムから割り出した時刻は実際の開閉時刻と大きなズレが生じていることも判明した。

6.2 本研究で開発したシステムの利点と問題点

本研究では学生から研究室の来訪時刻、帰宅時刻の報告を求め、研究室の開閉時刻を割り出し可視化するシステムを開発し、開閉時刻不明問題を解決するための手法を提案した。次項で本研究で開発したシステムの考察を示す。

6.3 利点

6.3.1 時刻の確認が容易にできる

開閉時刻は専用のWebページに表示されるため、案内メール等で通知しているURL^{*48}にアクセスするだけで確認が可能となる。Webページのスクリーンショットを図42に示す。

この点についてはユーザがメールを送るだけで処理が行われ、自動的に開閉時刻の割り出しを行いWebページに表示する仕組みを構築した。多くのユーザが来訪/帰宅時刻の登録を行うことにより、より正確な開閉時刻の表示が期待できる。

^{*48} <http://www0.ise.osaka-sandai.ac.jp/ks151/>

情報教育システム研究室 2018/11/30 のオープン/戸締り時刻

研究室は 11:00 に開く予定です。

戸締り時刻は 20:00 の予定です。

Last updated 2018/11/30 10:25:01
Copyright© 2018 T-Koyabu.
[Go List.](#)

図 42 開閉時刻表示用 Web ページの表示例。日付と開閉時刻、ページの更新日時が表示されている。

6.3.2 来訪/帰宅時刻が確認しやすい

ユーザがシステムに送った時刻は、メイリングリストを通じて他のユーザに共有されるだけでなく、専用 Web ページにも学籍番号と送った時刻が表示されるため、多くのメールにメイリングリストから届いたメールが埋もれてしまっても、専用 Web ページをチェックするだけでその日誰がいつ来訪予定でいつ帰宅予定なのかが一目瞭然となる。専用 Web ページの表示例を図 43 に示す。

情報教育システム研究室 2018/11/30 時刻リスト	
来る予定時刻	帰る予定時刻
15H038 11:00	15H038 20:00

Last updated 2018/11/30 10:25:01
Copyright© 2018 T-Koyabu.
Go Toppage.

図 43 ユーザから送られた来訪/帰宅時刻表示用 Web ページの表示例。日付と来訪時刻、帰宅時刻に分かれてユーザの学籍番号と時刻、更新日時が表示されている。

6.4 問題点

6.4.1 ユーザ ID の制約

ユーザから送られてきた時刻を表示する Web ページを作成する際にサーバにあるデータのユーザ名を加工しているが、大阪産業大学の学生ユーザの ID にしか対応していないため、それ以外のユーザから時刻が送られてきた場合には正常に表示されない。

6.4.2 メイリングリストのメンテナンス

ユーザ間での時刻の共有にメイリングリストを使用しているため、学生の増減に伴いメンバリストの変更を手動で行う必要がある。これを行わないと時刻の共有が行われただけでなく、一部の通知メールが届かなくなる。

6.4.3 時刻形式の制約

ユーザが時刻をメールで送る際に、24 時間制で時刻が記入されていない場合、システム側で読み取ることができない^{*49}ため、正常に処理されない。

^{*49} システム側で使用している正規表現が 24 時間制の時刻表記にしか対応していない。

7 結論

本研究では、ユーザからメールで研究室に来訪時刻と帰宅時刻を送ってもらうことにより、研究室の解錠時刻と施錠時刻の可視化を行い開閉時刻不明問題を解決するシステムを開発した。

その結果、ユーザが研究室の解錠時刻と施錠予定時刻を研究室を訪れる前に確認することができるようになった。ユーザ自身の来訪時刻も他のユーザに共有されるため、ユーザの行動を可視化し、研究室を訪れていない学生の割り出しに活用することが期待できる。

多くのユーザがメールで時刻を送ることにより、研究室の開閉時刻の信頼性が上がる。それにより、システムの時刻と実際の開閉時刻との差異がなくなり、学生が訪れた際にシステム上は開いていることになってはいるものの実際は閉まっていたなどの問題が発生する可能性が低くなる。本システムに多くのユーザが時刻を登録し、データの精度が高くなると、より正確な研究室の開閉時刻を求めることができるため、研究室運営に貢献することが期待できる。

7.1 今後の課題

今後の課題は、データの精度をより高くすること及び、ユーザから送られてくるメールのあらゆる文面を想定して処理を行うことである。上記の内容から以下の課題が考えられる。

7.1.1 データ精度向上

システムに記録されている開閉時刻と実際の開閉時刻に大きな差異がある。また、登録される時刻の件数が少なすぎるため、データの精度が低くなっている。多くのユーザに時刻の登録の依頼と、変更があった際のメール送信を依頼することにより、データの精度を上げ、より正確な時刻の割り出しが行えると考ええる。

7.1.2 2バイト文字への対応

本研究で開発したシステムでは、時刻を2バイト文字で書いてメールを送った際に文字化けを起こし、正常に動作しない問題がある。メールから時刻を取り出す方法を変えることにより、2バイト文字に対応できると考える。

謝辞

本研究を進めていく上で、担当教員である大垣斉准教授に御指導およびご協力を頂きました。また、情報教育システム研究室所属の学生の皆様、team.andrew-MLのメンバの皆様にもご協力を賜り、深く感謝の意を表します。

参考文献

- [1] 川上 あゆみ・水上 彩・塚田 浩二・椎尾一郎. 人々の行動を手軽に共有する生活空間エージェント. ヒューマンインタフェースシンポジウム論文集, 2009.

付録 A ソースコード

A.1 受信時処理

ソースコード 1 受信したメールの Subject:を読み取り、各種スクリプトに横流しを行うスクリプト (sortmail)

```
1 #!/bin/bash
2 # 届いたメールを振り分けるスクリプト
3 # 変数定義
4 input='cat -'
5 sj='echo "$input" | grep ^Subject | sed -e "s/Subject:_/" '
6 nowtime='date +"%Y/%m/%d.%H:%M%S" '
7 mailadd='echo "$input" | grep ^From: | sed -e 's/ //' | sed -e "s/From:/" -e
   "s/^.*<(.*)>.*$/\1/" -e "s/(.*)/" '
8
9 # メール振り分け処理
10 # ML管理者自動送信
11 if [[ $mailadd == "hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp" ]]; then
12 echo "$input" > /dev/null
13 echo "$nowtime_Autosend_MLadmin_Passing" >> /home/user/sortmail.log
14 exit 0
15 fi
16 # 登録(手動送信)
17 if [[ "$sj" =~ .*[cC]$ ]]; then
18 echo "$input" | sh /home/user/inbx
19 echo "$nowtime_Regist_Passing" >> /home/user/sortmail.log
20 exit 0
21 fi
22 # 登録(自動送信メール返信)
23 if [[ "$sj" =~ .*Closetime.* ]]; then
24 echo "$input" | sh /home/user/inbx
25 echo "$nowtime_Regist_Passing" >> /home/user/sortmail.log
26
27 exit 0
28 fi
29 # 変更
30 if [[ "$sj" =~ .*[eE]$ ]]; then
31 echo "$input" | sh /home/user/inbx-ed
32 echo "$nowtime_Edit_Passing" >> /home/user/sortmail.log
33 exit 0
34 fi
```

```

35 # 削除
36 if [[ "$sj" =~ .*[dD]$ ]]; then
37 echo "$input" | sh /home/user/inbx-cancel
38 echo "$nowtime Delete Passing" >> /home/user/sortmail.log
39 exit 0
40 fi
41 # 参照
42 if [[ "$sj" =~ .*[rR]$ ]]; then
43 echo "$input" | sh /home/user/inbx-re
44 echo "$nowtime Reference Passing" >> /home/user/sortmail.log
45 exit 0
46 fi
47 # 翌日
48 if [[ "$sj" =~ .*[tT]$ ]]; then
49 echo "$input" | sh /home/user/inbx-t
50 echo "$nowtime Regist-Tomorrow Passing" >> /home/user/sortmail.log
51 exit 0
52 fi
53 # 日付指定登録
54 if [[ "$sj" =~ .*[fF][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$ ]]; then
55 echo "$input" | sh /home/user/inbx-f
56 echo "$nowtime Regist-Future Passing" >> /home/user/sortmail.log
57 exit 0
58 fi
59 # 日付指定変更
60 if [[ "$sj" =~ .*[eE][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$ ]]; then
61 echo "$input" | sh /home/user/inbx-edf
62 echo "$nowtime Edit-Future Passing" >> /home/user/sortmail.log
63 exit 0
64 fi
65 # 日付指定削除
66 if [[ "$sj" =~ .*[dD][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$ ]]; then
67 echo "$input" | sh /home/user/inbx-cancel
68 echo "$nowtime Cancel-Future Passing" >> /home/user/sortmail.log
69 exit 0
70 fi
71 # 日付指定参照
72 if [[ "$sj" =~ .*[rR][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$ ]]; then
73 echo "$input" | sh /home/user/inbx-ref
74 echo "$nowtime Reference-Future Passing" >> /home/user/sortmail.log

```

```

75 exit 0
76 fi
77 # ヘルプ
78 if [[ "$sj" =~ .*[hH]$ ]]; then
79 echo "$input" | sh /home/user/inbx-h
80 echo "$nowtime_LHelp_LPassing" >> /home/user/sortmail.log
81 exit 0
82 fi
83
84 # 該当なし
85 echo "$input" > /dev/null
86 echo "$nowtime_LNot-applicable_LPassing" >> /home/user/sortmail.log

```

A.2 当日処理

ソースコード 2 返信用メールの自動送信を行うスクリプト (remindmail)

```

1  #/bin/sh
2  # 変数定義
3  today='date +%Y/%m/%d'
4  # メール配信
5  sendmail -t << EOF
6  From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
7  To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
8  Subject: Closetime $today
9
10 情報教育システム研究室に来る時間と帰る時間を
11 このメールに対する返信で報告してください。
12 送られて来た時刻のうち一番早い時刻を研究室が開く予定の時刻として、
13 一番遅い時刻を戸締りの予定時刻としてWebページに表示します。
14
15 なお、送られる際はHH:MM形式で時刻を書いてください。
16 (9:00のような書き方ではシステム側で拾われません)
17 また、21時50分(JST)以降の時刻を送られた場合は自動で21時50分(JST)に変換されます
18 。
19 <返信の例>
20 xx:yyに来ます
21 yy:zzに帰ります
22
23 ----->&----->&----->&----->&----->&-----
24 9時00分(JST)~18時00分(JST)までの間、時刻の変更があった場合のみ更新されます。

```

```

25 http://www0.ise.osaka-sandai.ac.jp/*****/
26
27 送られた時刻はMLに流れる他、Webでも確認できます。
28 http://www0.ise.osaka-sandai.ac.jp/*****/list/
29
30 このメールへの返信は17時45分(JST)まで受け付けています。
31 (返信の際、引用返信はせずに時間だけを書いてください)
32 ----->&----->&----->&----->&----->&-----
33 [お願い]
34 メールはG Suite for Educationのメールアドレス(@ge.osaka-sandai.ac.jp)
35 から送ってください。
36
37 時間を間違えて送ってしまった場合、Subjectを「E」として再度時刻を書き
38 送り直してください。
39
40 取り消す場合はSubjectを「D」と書いて空メールを送ってください。
41
42 また、翌日以降の来る/帰る予定の時刻を送られる場合は、
43 Subjectを「F'日付8桁'」として時刻を書いてメールを送ってください。
44 今日は、'date --date 'tomorrow' +%Y/%m/%d'から'date --date 'tomorrow 7days' +%
    Y/%m/%d'までの予定時刻を送ることができます。
45 ただし、過去の日付は指定できません。
46 <Subjectの例>
47 F'date --date 'tomorrow' +%Y/%m/%d'
48
49 ---
50 Tatsuya Koyabu
51 *****@4w.ise.osaka-sandai.ac.jp
52 .
53 EOF

```

ソースコード 3 時刻の登録処理を行うスクリプト (inbox)

```

1 #!/bin/sh
2 # 変数定義
3 # 標準入力(メール)を標準出力へ
4 input='cat -'
5 # @の左を取り出す
6 name='echo "$input" | grep ^From: | sed -e "s/From:/" -e "s/^.*<(\.*)>.*$
    /\1/" -e "s/(.*)/" | awk '{sub("@.*", ""); print $0;}''
7 # 返信用アドレスを取り出す

```

```

8 mailadd='echo "$input" | grep ^From: | sed -e 's/ //' | sed -e "s/From:/" -e
   "s/^.*<(\.*\)>.*$/\1/" -e "s/(.*)/" '
9 # 日付と時刻を取得
10 day='date +%a'
11 today='date +%Y/%m/%d'
12 nowtime='date +%H%M'
13 if test "1745" -le "$nowtime";then
14 # 締め切り以降のメールは/dev/nullへ
15 echo "$input" | sed '1,/^$/d' | nkf -mB | nkf -Z1 -w | grep -a -o "
   [0-2][0-9]:[0-5][0-9]" > /dev/null
16 sh /home/user/errors E10 ${mailadd}
17 exit 0
18 else
19 # 通常時処理
20 # 自動送信メールは/dev/nullへ入れ、その後の処理をしない。
21 if test "$mailadd" = "hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp";then
22 echo "$input" > /dev/null
23 exit 0
24 else
25 if test " `echo` $input `|` `grep` Content-Transfer-Encoding: `|` `sed` 's/Content-
   Transfer-Encoding: //g' `|` `uniq` " = "base64";then
26 # base64エンコードの場合
27 # 来る予定の時刻をリストへ
28 echo "$input" | sed '1,/^$/d' | nkf -mB | nkf -Z1 -w | grep -a -o "
   [0-2][0-9]:[0-5][0-9]" | sort | uniq | head -1 > /home/user/
   comelablist.$name
29 # 帰る予定の時刻をリストへ
30 echo "$input" | sed '1,/^$/d' | nkf -mB | nkf -Z1 -w | grep -a -o "
   [0-2][0-9]:[0-5][0-9]" | sort -r | uniq | head -1 > /home/user/
   gohomelist.$name
31 else
32 # Signatureが短いなどbase64でない場合
33 # 来る予定の時刻をリストへ
34 echo "$input" | sed '1,/^$/d' | nkf -Z1 -w | grep -a -o "
   [0-2][0-9]:[0-5][0-9]" | sort | uniq | head -1 > /home/user/
   comelablist.$name
35 # 帰る予定の時刻をリストへ
36 echo "$input" | sed '1,/^$/d' | nkf -Z1 -w | grep -a -o "
   [0-2][0-9]:[0-5][0-9]" | sort -r | uniq | head -1 > /home/user/
   gohomelist.$name

```

```

37 fi
38 # 来る時間をリストから抽出
39 cometime='cat /home/user/comelablist_$name | grep -o "[0-2][0-9]:[0-5][0-9]" |
    sort | head -1'
40 # 帰る時間をリストから抽出
41 gohometime='cat /home/user/gohomelist_$name | grep -o "[0-2][0-9]:[0-5][0-9]"
    | sort -r | head -1'
42 # 来る時間と帰る時間が同じの場合はエラーとする
43 if test "$cometime" = "$gohometime";then
44 sh /home/user/errors E43 ${mailadd}
45 rm -rf /home/user/comelablist_$name
46 rm -rf /home/user/gohomelist_$name
47 exit 0
48 fi
49 cometime2='echo $cometime | sed s://g'
50 gohometime2='echo $gohometime | sed s://g'
51 # 21:51以降の時刻が書かれていた場合はエラーとする
52 if test "$cometime2" -ge "2151" -o "$gohometime2" -ge "2151";then
53 sh /home/user/errors E16 ${mailadd}
54 rm -rf /home/user/comelablist_$name
55 rm -rf /home/user/gohomelist_$name
56 exit 0
57 fi
58 # ログ出力
59 echo " `date +%Y/%m/%d %H:%M%S` '$_$name_comelabtime_Registration_success...'
    $cometime" >> /home/user/clrecieve.log
60 echo " `date +%Y/%m/%d %H:%M%S` '$_$name_gohometime_Registration_success...'
    $gohometime" >> /home/user/ghrecieve.log
61 # メール配信
62 sendmail -t << EOF
63 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
64 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
65 To: $mailadd
66 Subject:$name Lab Come/GoHome time Registration success. $today
67
68 $name さんの来る予定の時刻と帰る予定の時刻が登録されました。
69 _____>&_____>&_____>&_____>&_____>&_____
70 来る予定の時刻:$cometime
71 帰る予定の時刻:$gohometime
72 _____>&_____>&_____>&_____>&_____>&_____

```

```

73
74 登録された時刻の変更は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
75 Subjectに「E」、本文に新しい時刻を書いて送ってください。
76
77 キャンセルする場合は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
78 Subjectに「D」と書いて空メールを送って下さい。
79 (大文字と小文字は区別されません。また、半角で入力して下さい。)
80
81 ***本メールは自動で作成されたメールです。返信しないでください。***
82
83 —
84 Tatsuya Koyabu
85 *****@a4w.ise.osaka-sandai.ac.jp
86 .
87 EOF
88 fi
89 fi
90
91 # 各種更新処理呼出
92 if test "$nowtime" -le "0900";then
93 # 09:00より前はhtmlの更新をしない
94 sh /home/user/listin
95 else
96 sh /home/user/listin
97 sh /home/user/comehtml
98 sh /home/user/grepmember
99 fi

```

ソースコード 4 時刻の変更処理を行うスクリプト (inbx-ed)

```

1 #!/bin/sh
2 # 変数定義
3 # 標準入力(メール)を標準出力へ
4 input='cat -'
5 # @の左を取り出す
6 name='echo "$input" | grep ^From: | sed -e "s/From:/" -e "s/^.*<\(.*\)>.*$
   /\1/" -e "s/(.*)/" | awk '{sub("@.*", "");print $0;}''
7 # 返信用アドレスを取り出す(このスクリプトではこのアドレスに送る機能がない)
8 mailadd='echo "$input" | grep ^From: | sed -e 's/ //' | sed -e "s/From:/" -e
   "s/^.*<\(.*\)>.*$\1/" -e "s/(.*)/"'
9 # 日付と時刻を取得
10 day='date +%a'

```

```

11 today='date +%Y/%m/%d'
12 nowtime='date +%H%M'
13 if test "1745" -le "$nowtime";then
14 # 締め切り以降のメールは/dev/nullへ
15 echo "$input" | sed '1,/^$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" > /dev/null
16 sh /home/user/errors E20 ${mailadd}
17 exit 0
18 elif test "`find /home/user/comelablist_${name}_2>/dev/null " = "" -a "`find /
    home/user/gohomelist_${name}_2>/dev/null " = "" ;then
19 # 送られた人のファイルが存在しない場合は/dev/nullへ
20 sh /home/user/errors E21 ${mailadd}
21 exit 0
22 else
23 # 通常時処理
24 # 自動送信メールは/dev/nullへ入れ、その後の処理をしない。
25 if test "$mailadd" = "hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp";then
26 echo "$input" > /dev/null
27 exit 0
28 else
29 if test "`echo "$input" | grep Content-Transfer-Encoding: | sed 's/Content-
    Transfer-Encoding: //g' | uniq " = "base64";then
30 # base64エンコードの場合
31 # 来る予定の時刻をリストへ
32 echo "$input" | sed '1,/^$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" | sort | uniq | head -1 > /home/user/
    comelablist.$name
33 # 帰る予定の時刻をリストへ
34 echo "$input" | sed '1,/^$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" | sort -r | uniq | head -1 > /home/user/
    gohomelist.$name
35 else
36 # Signatureが短いなどbase64でない場合
37 # 来る予定の時刻をリストへ
38 echo "$input" | sed '1,/^$/d' | nkf -Z | grep -a -o "[0-2][0-9]:[0-5][0-9]" |
    sort | uniq | head -1 > /home/user/comelablist.$name
39 # 帰る予定の時刻をリストへ
40 echo "$input" | sed '1,/^$/d' | nkf -Z | grep -a -o "[0-2][0-9]:[0-5][0-9]" |
    sort -r | uniq | head -1 > /home/user/gohomelist.$name
41 fi

```

```

42 # 来る時間をリストから抽出
43 cometime='cat /home/user/comelablist_$(name) | grep -o "[0-2][0-9]:[0-5][0-9]" |
    sort | head -1'
44 # 帰る時間をリストから抽出
45 gohometime='cat /home/user/gohomelist_$(name) | grep -o "[0-2][0-9]:[0-5][0-9]"
    | sort -r | head -1'
46 # 来る時間と帰る時間が同じの場合はエラーとする
47 if test "$cometime" = "$gohometime";then
48 sh /home/user/errors E43 ${mailadd}
49 : > /home/user/comelablist_$(name)
50 : > /home/user/gohomelist_$(name)
51 exit 0
52 fi
53 cometime2='echo $cometime | sed s://g'
54 gohometime2='echo $gohometime | sed s://g'
55 # 21:51以降の時刻が書かれていた場合はエラーとする
56 if test "$cometime2" -ge "2151" -o "$gohometime2" -ge "2151";then
57 sh /home/user/errors E23 ${mailadd}
58 : >/home/user/comelablist_$(name)
59 : > /home/user/gohomelist_$(name)
60 exit 0
61 fi
62 # ログ出力
63 echo " $(date +%Y/%m/%d %H:%M:%S) $(name)_comelabtime_Change_success_.$cometime"
    >> /home/user/clrecieve.log
64 echo " $(date +%Y/%m/%d %H:%M:%S) $(name)_gohometime_Change_success_.$gohometime
    " >> /home/user/ghrecieve.log
65 # メール配信
66 sendmail -t << EOF
67 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
68 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
69 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
70 Subject:Lab Come/Go Home time Change success. on $(name) at $today
71
72 $(name) さんの送られた時刻が変更されました。
73 ----->&----->&----->&----->&----->&-----
74 来る予定の時刻: $cometime
75 帰る予定の時刻: $gohometime
76 ----->&----->&----->&----->&----->&-----
77

```

```

78 帰る予定の時刻の変更は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
79 Subjectに「E」、本文に新しい時刻を書いて送ってください。
80
81 キャンセルする場合は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
82 Subjectに「D」と書いて空メールを送って下さい。
83 (大文字と小文字は区別されません。また、半角で入力して下さい。)
84
85 ***本メールは自動で作成されたメールです。返信しないでください。***
86
87 —
88 Tatsuya Koyabu
89 *****@a4w.ise.osaka-sandai.ac.jp
90 .
91 EOF
92 fi
93 fi
94
95 # 各種更新処理呼出
96 if test "$nowtime" -le "0900";then
97 # 09:00より前はhtmlの更新をしない
98 sh /home/user/listin
99 else
100 sh /home/user/listin
101 sh /home/user/comehtml
102 sh /home/user/grepmember
103 fi

```

ソースコード 5 時刻の取り消し処理を行うスクリプト (inbx-cancel)

```

1 #!/bin/sh
2 # 変数定義
3 # 標準入力(メール)を標準出力へ
4 input='cat -'
5 # @の左を取り出す
6 name='echo "$input" | grep ^From: | sed -e "s/From:/" -e "s/^.*< \(.*\)>.*$
    /\1/" -e "s/(.*)/" | awk '{sub("@.*", ""); print $0;}''
7 # 返信用アドレスを取り出す
8 mailadd='echo "$input" | grep ^From: | sed -e 's/ //' | sed -e "s/From:/" -e
    "s/^.*< \(.*\)>.*$\1/" -e "s/(.*)/"
9 # 日付と時刻を取得
10 day='date +%a'
11 today='date +%Y/%m/%d'

```

```

12 nowtime='date +%H%M'
13 if test "1745" -le "$nowtime";then
14 # 締め切り以降のメールは/dev/nullへ
15 echo "$input" | sed '1,/^\$/d' | nkf -mB | grep -a -o "[0-2][0-9]:[0-5][0-9]" >
    /dev/null
16 sh /home/user/errors E30 ${mailadd}
17 exit 0
18 elif test "`find ~/home/user/comelablist_${name}_2>/dev/null " = "" -a "`find ~/
    home/user/gohomelist_${name}_2>/dev/null " = "" ;then
19 # 送られた人のファイルが存在しない場合は/dev/nullへ
20 sh /home/user/errors E34 ${mailadd}
21 exit 0
22 else
23 # 通常時処理
24 # 自動送信メールは/dev/nullへ入れ、その後の処理をしない。
25 if test "$mailadd" = "hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp";then
26 echo "$input" > /dev/null
27 exit 0
28 else
29 # ファイルを強制的に削除
30 rm -rf /home/user/comelablist_${name}
31 rm -rf /home/user/gohomelist_${name}
32 # ログ出力
33 echo "`date +%Y/%m/%d %H:%M:%S" `_${name}_comelabtime_Delete_success." >> /home/
    user/clrecieve.log
34 echo "`date +%Y/%m/%d %H:%M:%S" `_${name}_gohometime_Delete_success." >> /home/
    user/ghrecieve.log
35 # メール配信
36 sendmail -t << EOF
37 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
38 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
39 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
40 Subject:Lab Come/Go Home time Delete success. on $name at $today
41 $name さんの送られた時刻は削除されました。
42 本日は研究室に現れないかもしれません。
43
44 ***本メールは自動で作成されたメールです。返信しないでください。***
45
46 —
47 Tatsuya Koyabu

```

```

48 *****@a4w.ise.osaka-sandai.ac.jp
49 .
50 EOF
51 fi
52 fi
53
54 # 各種更新処理呼出
55 if test "$nowtime" -le "0900";then
56 # 09:00より前はhtmlの更新をしない
57 sh /home/user/listin
58 else
59 sh /home/user/listin
60 sh /home/user/comehtml
61 sh /home/user/grepmember
62 fi

```

ソースコード 6 時刻の参照処理を行うスクリプト (inbx-re)

```

1 #!/bin/sh
2 # 変数定義
3 # 標準入力(メール)を標準出力へ
4 input='cat -'
5 # @の左を取り出す
6 name='echo "$input" | grep ^From: | sed -e "s/From:/" -e "s/^.*< \(.*\)>.*$
   /\1/" -e "s/(.*)/" | awk '{sub("@.*", "");print $0;}''
7 # 返信用アドレスを取り出す
8 mailadd='echo "$input" | grep ^From: | sed -e 's/ //' | sed -e "s/From:/" -e
   "s/^.*< \(.*\)>.*$/\1/" -e "s/(.*)/" '
9 # 日付と時刻を取得
10 day='date +%a'
11 today='date +%Y/%m/%d'
12 nowtime='date +%H%M'
13
14 if test "`find _/home/user/comelablist_${name}_2>/dev/null " = "" -a "`find _/
   home/user/gohomelist_${name}_2>/dev/null " = "" ;then
15 # 送られた人のファイルが存在しない場合は/dev/nullへ
16 sh /home/user/errors E41 ${mailadd}
17 exit 0
18 else
19 # 通常時処理
20 # 自動送信メールは/dev/nullへ入れ、その後の処理をしない。
21 if test "$mailadd" = "hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp" ;then

```

```

22 echo "$input" > /dev/null
23 exit 0
24 else
25 # 来る時間をリストから抽出
26 cometime='cat /home/user/comelablist_$name | grep -o "[0-2][0-9]:[0-5][0-9]" |
    sort | head -1'
27 # 帰る時間をリストから抽出
28 gohometime='cat /home/user/gohomelist_$name | grep -o "[0-2][0-9]:[0-5][0-9]"
    | sort -r | head -1'
29 # ログ出力
30 echo "`date +%Y/%m/%d %H:%M:%S` `$_name_comelabtime_Reference_success.`" >> /
    home/user/clrecieve.log
31 echo "`date +%Y/%m/%d %H:%M:%S` `$_name_gohometime_Reference_success.`" >> /
    home/user/ghrecieve.log
32 # メール配信
33 sendmail -t << EOF
34 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
35 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
36 To: ${mailadd}
37 Subject:Lab Come/Go Home inform $name sent time at $today
38
39 $name さんの送られた時刻をお知らせします。
40 ----->&----->&----->&----->&----->&-----
41 来る予定の時刻:$cometime
42 帰る予定の時刻:$gohometime
43 ----->&----->&----->&----->&----->&-----
44
45 帰る予定の時刻の変更は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
46 Subjectに「E」、本文に新しい時刻を書いて送ってください。
47
48 キャンセルする場合は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
49 Subjectに「D」と書いて空メールを送って下さい。
50 (大文字と小文字は区別されません。また、半角で入力して下さい。)
51
52 ***本メールは自動で作成されたメールです。返信しないでください。***
53
54 ---
55 Tatsuya Koyabu
56 *****@a4w.ise.osaka-sandai.ac.jp
57 .

```

58 EOF

59 fi

60 fi

A.3 翌日以降処理

ソースコード 7 翌日以降の時刻の登録処理を行うスクリプト (inbx-f)

```
1 #!/bin/sh
2 # 変数定義
3 # 標準入力(メール)を標準出力へ
4 input='cat -'
5 # @の左を取り出す
6 name='echo "$input" | grep ^From: | sed -e "s/From:/" -e "s/^.*< \(.*\)>.*$
   /\1/" -e "s/(.*)/" | awk '{sub("@.*", ""); print $0;}''
7 # 返信用アドレスを取り出す
8 mailadd='echo "$input" | grep ^From: | sed -e 's/ //' | sed -e "s/From:/" -e
   "s/^.*< \(.*\)>.*$/\1/" -e "s/(.*)/"'
9 # 日付と時刻を取得
10 day='date +%a'
11 today='date +%Y/%m/%d'
12 todaynum='date +%Y%m%d'
13 nowtime='date +%H%M'
14 dateno='echo "$input" | grep ^Subject: | sed -e 's/ //' | sed -e 's/Subject
   :/' | grep -o '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]''
15 datenum='date --date "${dateno}" +%Y%m%d'
16 dates='date --date "${dateno}" +%Y/%m/%d'
17 fda='date --date "${dateno}" +%a'
18 nw='date --date "tomorrow_7days" +%Y/%m/%d'
19 if test "${datenum}" -le "${todaynum}";then
20 # 本日含む過去の日付のメールは/dev/nullへ
21 echo "$input" | sed '1,/^$/d' | nkf -mB | nkf -Z | grep -a -o "
   [0-2][0-9]:[0-5][0-9]" > /dev/null
22 sh /home/user/errors E15 ${mailadd}
23 exit 0
24 elif test "${datenum}" -gt "${nw}";then
25 # 一週間後以降の日付を書いてきた場合は/dev/nullへ
26 echo "$input" | sed '1,/^$/d' | nkf -mB | nkf -Z | grep -a -o "
   [0-2][0-9]:[0-5][0-9]" > /dev/null
27 sh /home/user/errors E14 ${mailadd}
28 exit 0
29 elif test "${dateno}" -ge "20380119";then
30 # 2038年問題への対応
```

```

31 echo "$input" | sed '1,/^\$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" > /dev/null
32 sh /home/user/errors E40 ${mailadd}
33 exit 0
34 else
35 # 通常時処理
36 # 自動送信メールは/dev/nullへ入れ、その後の処理をしない。
37 if test "$mailadd" = "hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp";then
38 echo "$input" > /dev/null
39 exit 0
40 else
41 if test " `echo` "$input" `|` `grep` `Content-Transfer-Encoding:` `|` `sed` `s/Content-
    Transfer-Encoding: //g` `|` `uniq` " = "base64";then
42 # base64エンコードの場合
43 # 来る予定の時刻をリストへ
44 echo "$input" | sed '1,/^\$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" | sort | uniq | head -1 > /home/user/${datenum}
    comelablist_$name
45 # 帰る予定の時刻をリストへ
46 echo "$input" | sed '1,/^\$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" | sort -r | uniq | head -1 > /home/user/${datenum}
    gohomelist_$name
47 else
48 # Signatureが短いなどbase64でない場合
49 # 来る予定の時刻をリストへ
50 echo "$input" | sed '1,/^\$/d' | nkf -Z | grep -a -o "[0-2][0-9]:[0-5][0-9]" |
    sort | uniq | head -1 > /home/user/${datenum}comelablist_$name
51 # 帰る予定の時刻をリストへ
52 echo "$input" | sed '1,/^\$/d' | nkf -Z | grep -a -o "[0-2][0-9]:[0-5][0-9]" |
    sort -r | uniq | head -1 > /home/user/${datenum}gohomelist_$name
53 fi
54 # 来る時間をリストから抽出
55 cometime=`cat /home/user/${datenum}comelablist_$name | grep -o "
    [0-2][0-9]:[0-5][0-9]" | sort | head -1`
56 # 帰る時間をリストから抽出
57 gohometime=`cat /home/user/${datenum}gohomelist_$name | grep -o "
    [0-2][0-9]:[0-5][0-9]" | sort -r | head -1`
58 # 来る時間と帰る時間が同じの場合はエラーとする
59 if test "$cometime" = "$gohometime";then
60 sh /home/user/errors E43 ${mailadd}

```

```

61 rm -rf /home/user/${datenum}comelablist_${name}
62 rm -rf /home/user/${datenum}gohomelist_${name}
63 exit 0
64 fi
65 cometime2='echo $cometime | sed s://g'
66 gohometime2='echo $gohometime | sed s://g'
67 # 21:51以降の時刻が書かれていた場合はエラーとする
68 if test "$cometime2" -ge "2151" -o "$gohometime2" -ge "2151";then
69 sh /home/user/errors E16 ${mailadd}
70 rm -rf /home/user/${datenum}comelablist_${name}
71 rm -rf /home/user/${datenum}gohomelist_${name}
72 exit 0
73 fi
74 # ログ出力
75 echo "date_+%Y/%m/%d %H:%M:%S" "_${name}_${dates}_comelabtime_Registration_
    success._${cometime}" >> /home/user/clrecieve.log
76 echo "date_+%Y/%m/%d %H:%M:%S" "_${name}_${dates}_gohometime_Registration_success
    ._${gohometime}" >> /home/user/ghrecieve.log
77 # メール配信
78 sendmail -t << EOF
79 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
80 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
81 To: $mailadd
82 Subject:$name $datenum Lab Come/GoHome time Registration success. $dates
83
84 <<この時刻は${dates}分です>>
85 $name さんの${dates}の来る予定の時刻と帰る予定の時刻が登録されました。
86 ----->&----->&----->&----->&----->&-----
87 来る予定の時刻:$cometime
88 帰る予定の時刻:$gohometime
89 ----->&----->&----->&----->&----->&-----
90
91 帰る予定の時刻の変更は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
92 Subjectに「E${datenum}」、本文に新しい時刻を書いて送ってください。
93
94 キャンセルする場合は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
95 Subjectに「D${datenum}」と書いて空メールを送って下さい。
96 (大文字と小文字は区別されません。また、半角で入力して下さい。)
97
98 ***本メールは自動で作成されたメールです。返信しないでください。***

```

```

99
100 —
101 Tatsuya Koyabu
102 *****@a4w.ise.osaka-sandai.ac.jp
103 .
104 EOF
105 fi
106 fi

```

ソースコード 8 翌日以降の時刻の変更処理を行うスクリプト (inbx-edf)

```

1 #!/bin/sh
2 # 変数定義
3 # 標準入力(メール)を標準出力へ
4 input='cat -'
5 # @の左を取り出す
6 name='echo "$input" | grep ^From: | sed -e "s/From:/" -e "s/^.*< \(.*\)>.*$
   /\1/" -e "s/(.*)/" | awk '{sub("@.*", ""); print $0;}''
7 # 返信用アドレスを取り出す(このスクリプトではこのアドレスに送る機能がない)
8 mailadd='echo "$input" | grep ^From: | sed -e 's/ //' | sed -e "s/From:/" -e
   "s/^.*< \(.*\)>.*$\1/" -e "s/(.*)/" '
9 # 日付と時刻を取得
10 day='date +%a'
11 today='date +%Y/%m/%d'
12 todaynum='date +%Y%m%d'
13 nowtime='date +%H%M'
14 dateno='echo "$input" | grep '^Subject:' | sed -e 's/ //' | sed -e 's/Subject
   :/' | grep -o '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]''
15 datenum='date --date "${dateno}" +%Y%m%d'
16 dates='date --date "${dateno}" +%Y/%m/%d'
17 fda='date --date "${dateno}" +%a'
18 nw='date --date "tomorrow_7days" +%Y%m%d'
19 clfind='find /home/user/${datenum}comelablist_${name} 2>/dev/null '
20 ghfind='find /home/user/${datenum}gohomelist_${name} 2>/dev/null '
21 if test "${datenum}" -le "${todaynum}";then
22 # 本日含む過去の日付のメールは/dev/nullへ
23 echo "$input" | sed '1,/^\$/d' | nkf -mB | nkf -Z | grep -a -o "
   [0-2][0-9]:[0-5][0-9]" > /dev/null
24 sh /home/user/errors E15 ${mailadd}
25 exit 0
26 elif test "${datenum}" -gt "${nw}";then
27 # 一週間後以降の日付を書いてきた場合は/dev/nullへ

```

```

28 echo "$input" | sed '1,/^\$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" > /dev/null
29 sh /home/user/errors E14 ${mailadd}
30 exit 0
31 elif test "${dateno}" -ge "20380119";then
32 # 2038年問題への対応
33 echo "$input" | sed '1,/^\$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" > /dev/null
34 sh /home/user/errors E40 ${mailadd}
35 exit 0
36 elif test "${clfind}" = "" -a "${ghfind}" = "";then
37 # 送られた人のファイルが存在しない場合は/dev/nullへ
38 echo "$input" | sed '1,/^\$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" > /dev/null
39 sh /home/user/errors E22 ${mailadd}
40 exit 0
41 else
42 # 通常時処理
43 # 自動送信メールは/dev/nullへ入れ、その後の処理をしない。
44 if test "$mailadd" = "hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp";then
45 echo "$input" > /dev/null
46 exit 0
47 else
48 if test " `echo "$input" | _grep _Content-Transfer-Encoding: | _sed _'s/Content-
    Transfer-Encoding: //g' | _uniq ` " = "base64";then
49 # base64エンコードの場合
50 # 来る予定の時刻をリストへ
51 echo "$input" | sed '1,/^\$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" | sort | uniq | head -1 > /home/user/${datenum}
    comelablist_$name
52 # 帰る予定の時刻をリストへ
53 echo "$input" | sed '1,/^\$/d' | nkf -mB | nkf -Z | grep -a -o "
    [0-2][0-9]:[0-5][0-9]" | sort -r | uniq | head -1 > /home/user/${datenum}
    gohomelist_$name
54 else
55 # Signatureが短いなどbase64でない場合
56 # 来る予定の時刻をリストへ
57 echo "$input" | sed '1,/^\$/d' | nkf -Z | grep -a -o "[0-2][0-9]:[0-5][0-9]" |
    sort | uniq | head -1 > /home/user/${datenum}comelablist_$name
58 # 帰る予定の時刻をリストへ

```

```

59 echo "$input" | sed '1,/^\$/d' | nkf -Z | grep -a -o "[0-2][0-9]:[0-5][0-9]" |
    sort -r | uniq | head -1 > /home/user/${datenum}gohomelist_$name
60 fi
61 # 来る時間をリストから抽出
62 cometime='cat /home/user/${datenum}comelablist_$name | grep -o "
    [0-2][0-9]:[0-5][0-9]" | sort | head -1'
63 # 帰る時間をリストから抽出
64 gohometime='cat /home/user/${datenum}gohomelist_$name | grep -o "
    [0-2][0-9]:[0-5][0-9]" | sort -r | head -1'
65 # 来る時間と帰る時間が同じの場合はエラーとする
66 if test "$cometime" = "$gohometime";then
67 sh /home/user/errors E43 ${mailadd}
68 : > /home/user/${datenum}comelablist_$name
69 : > /home/user/${datenum}gohomelist_$name
70 exit 0
71 fi
72 cometime2='echo $cometime | sed s://g'
73 gohometime2='echo $gohometime | sed s://g'
74 # 21:51以降の時刻が書かれていた場合はエラーとする
75 if test "$cometime2" -ge "2151" -o "$gohometime2" -ge "2151";then
76 sh /home/user/errors E23 ${mailadd}
77 : > /home/user/${datenum}comelablist_$name
78 : > /home/user/${datenum}gohomelist_$name
79 exit 0
80 # ログ出力
81 echo "`date +%Y/%m/%d %H:%M:%S" `_${name}_${dates}_comelabtime_Change_success...
    $cometime" >> /home/user/clrecieve.log
82 echo "`date +%Y/%m/%d %H:%M:%S" `_${name}_${dates}_gohometime_Change_success...
    $gohometime" >> /home/user/ghrecieve.log
83 # メール配信
84 sendmail -t << EOF
85 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
86 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
87 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
88 Subject:Lab Come/Go Home time Change success. on $name at $dates
89
90 <<この時刻は${dates}分です>>
91 $name さんの${dates}の来る/帰る予定として送られた時刻が変更されました。
92 ----->&----->&----->&----->&----->&-----
93 来る予定の時刻: $cometime

```

```

94 帰る予定の時刻:$gohometime
95  ----->&----->&----->&----->&----->&-----
96
97 帰る予定の時刻の変更は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
98 Subjectに「E${datenum}」、本文に新しい時刻を書いて送ってください。
99
100 キャンセルする場合は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
101 Subjectに「D${datenum}」と書いて空メールを送って下さい。
102 (大文字と小文字は区別されません。また、半角で入力して下さい。)
103
104 ***本メールは自動で作成されたメールです。返信しないでください。***
105
106  —
107 Tatsuya Koyabu
108 *****@a4w.ise.osaka-sandai.ac.jp
109 .
110 EOF
111 fi
112 fi

```

ソースコード 9 翌日以降の時刻の取り消し処理を行うスクリプト (inbx-cancel)

```

1  #!/bin/sh
2  # 変数定義
3  # 標準入力(メール)を標準出力へ
4  input='cat -'
5  # @の左を取り出す
6  name='echo "$input" | grep ^From: | sed -e "s/From:/" -e "s/^.*<\(.*\)>.*$
   \1/" -e "s/(.*)/" | awk '{sub("@.*", ""); print $0;}''
7  # 返信用アドレスを取り出す
8  mailadd='echo "$input" | grep ^From: | sed -e 's/ //' | sed -e "s/From:/" -e
   "s/^.*<\(.*\)>.*$\1/" -e "s/(.*)/" '
9  # 日付と時刻を取得
10 day='date +%a'
11 today='date +%Y/%m/%d'
12 todaynum='date +%Y%m%d'
13 nowtime='date +%H%M'
14 dateno='echo "$input" | grep '^Subject:' | sed -e 's/ //' | sed -e 's/Subject
   :/' | grep -o '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]''
15 datenum='date --date "${dateno}" +%Y%m%d'
16 dates='date --date "${dateno}" +%Y/%m/%d'
17 fda='date --date "${dateno}" +%a'

```

```

18 nw='date --date "tomorrow_7days" +%Y/%m/%d'
19 clfind='find /home/user/${datenum}comelablist_${name} 2>/dev/null '
20 ghfind='find /home/user/${datenum}gohomelist_${name} 2>/dev/null '
21 if test "${datenum}" -le "${todaynum}";then
22 # 本日含む過去の日付のメールは/dev/nullへ
23 echo "$input" | sed '1,/^\$/d' | nkf -mB | grep -a -o "[0-2][0-9]:[0-5][0-9]" >
    /dev/null
24 sh /home/user/errors E15 ${mailadd}
25 exit 0
26 elif test "${datenum}" -gt "${nw}";then
27 # 一週間後以降の日付を書いてきた場合は/dev/nullへ
28 echo "$input" | sed '1,/^\$/d' | nkf -mB | grep -a -o "[0-2][0-9]:[0-5][0-9]" >
    /dev/null
29 sh /home/user/errors E14 ${mailadd}
30 exit 0
31 elif test "${dateno}" -ge "20380119";then
32 # 2038年問題への対応
33 echo "$input" | sed '1,/^\$/d' | nkf -mB | grep -a -o "[0-2][0-9]:[0-5][0-9]" >
    /dev/null
34 sh /home/user/errors E40 ${mailadd}
35 exit 0
36 elif test "${clfind}" = "" -a "${ghfind}" = "";then
37 # 送られた人のファイルが存在しない場合は/dev/nullへ
38 echo "$input" | sed '1,/^\$/d' | nkf -mB | grep -a -o "[0-2][0-9]:[0-5][0-9]" >
    /dev/null
39 sh /home/user/errors E35 ${mailadd}
40 exit 0
41 else
42 # 通常時処理
43 # 自動送信メールは/dev/nullへ入れ、その後の処理をしない。
44 if test "$mailadd" = "hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp";then
45 echo "$input" > /dev/null
46 exit 0
47 else
48 # ファイルを強制的に削除
49 rm -rf /home/user/${datenum}comelablist_${name}
50 rm -rf /home/user/${datenum}gohomelist_${name}
51 # ログ出力
52 echo "'date_+'%Y/%m/%d %H:%M:%S" '_$name_$dates_comelabtime_Delete_success.'" >>
    /home/user/clrecieve.log

```

```

53 echo " `date +%Y/%m/%d %H:%M%S` " `$_name`$_dates`_gohometime_Delete_success.` >>
    /home/user/ghrecieve.log
54 # メール配信
55 sendmail -t << EOF
56 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
57 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
58 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
59 Subject:Lab Come/Go Home time Delete success. on $name at $dates
60
61 <<このメールは${dates}分です>>
62 $name さんの${dates}の来る/帰る予定として送られた時刻は削除されました。
63 ${dates}は研究室に現れないかもしれません。
64
65 ***本メールは自動で作成されたメールです。返信しないでください。***
66
67 —
68 Tatsuya Koyabu
69 *****@a4w.ise.osaka-sandai.ac.jp
70 .
71 EOF
72 fi
73 fi

```

ソースコード 10 翌日以降の時刻の参照処理を行うスクリプト (inbx-ref)

```

1 #!/bin/sh
2 # 変数定義
3 # 標準入力(メール)を標準出力へ
4 input=`cat -`
5 # @の左を取り出す
6 name=`echo "$input" | grep ^From: | sed -e "s/From:/" -e "s/^.*<(\.*\)>.*$
    /\1/" -e "s/(.*)/" | awk '{sub("@.*", ""); print $0;}'`
7 # 返信用アドレスを取り出す
8 mailadd=`echo "$input" | grep ^From: | sed -e 's/ //' | sed -e "s/From:/" -e
    "s/^.*<(\.*\)>.*$\1/" -e "s/(.*)/"`
9 # 日付と時刻を取得
10 day=`date +%a`
11 today=`date +%Y/%m/%d`
12 todaynum=`date +%Y%m%d`
13 nowtime=`date +%H%M`
14 dateno=`echo "$input" | grep ^Subject: | sed -e 's/ //' | sed -e 's/Subject
    :/' | grep -o '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'`

```

```

15 datenum='date --date "${dateno}" +%Y%a%f%d'
16 dates='date --date "${dateno}" +%Y/%a/%d'
17 fda='date --date "${dateno}" +%a'
18 nw='date --date "tomorrow_7days" +%Y%a%f%d'
19 clfind='find /home/user/${datenum}comelablist_${name} 2>/dev/null '
20 ghfind='find /home/user/${datenum}gohomelist_${name} 2>/dev/null '
21 if test "${datenum}" -le "${todaynum}";then
22 # 今日含む過去の日付のメールは/dev/nullへ
23 echo "$input" > /dev/null
24 sh /home/user/errors E15 ${mailadd}
25 exit 0
26 elif test "${datenum}" -gt "${nw}";then
27 # 一週間後以降の日付を書いてきた場合は/dev/nullへ
28 echo "$input" > /dev/null
29 sh /home/user/errors E14 ${mailadd}
30 exit 0
31 elif test "${dateno}" -ge "20380119";then
32 # 2038年問題への対応
33 echo "$input" > /dev/null
34 sh /home/user/errors E40 ${mailadd}
35 exit 0
36 elif test "${clfind}" = "" -a "${ghfind}" = "";then
37 # 送られた人のファイルが存在しない場合は/dev/nullへ
38 echo "$input" > /dev/null
39 sh /home/user/errors E42 ${mailadd}
40 exit 0
41 else
42 # 通常時処理
43 # 自動送信メールは/dev/nullへ入れ、その後の処理をしない。
44 if test "$mailadd" = "hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp";then
45 echo "$input" > /dev/null
46 exit 0
47 else
48 # 来る時間をリストから抽出
49 cometime='cat /home/user/${datenum}comelablist_${name} | grep -o "
          [0-2][0-9]:[0-5][0-9]" | sort | head -1'
50 # 帰る時間をリストから抽出
51 gohometime='cat /home/user/${datenum}gohomelist_${name} | grep -o "
          [0-2][0-9]:[0-5][0-9]" | sort -r | head -1'
52 # ログ出力

```

```

53 echo " `date +%Y/%m/%d %H:%M%S` " `_$name_$dates_comelabtime_Reference_success.`"
    >>/home/user/clrecieve.log
54 echo " `date +%Y/%m/%d %H:%M%S` " `_$name_$dates_gohometime_Reference_success.`"
    >> /home/user/ghrecieve.log
55 # メール配信
56 sendmail -t << EOF
57 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
58 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
59 To: ${mailadd}
60 Subject:Lab Come/Go Home inform $name sent time at $dates
61
62 <<この時刻は${dates}分です>>
63 $name さんの${dates}の来る/帰る予定として送られた時刻をお知らせします。
64 ----->&----->&----->&----->&----->&-----
65 来る予定の時刻: $cometime
66 帰る予定の時刻: $gohometime
67 ----->&----->&----->&----->&----->&-----
68
69 帰る予定の時刻の変更は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
70 Subjectに「E${datenum}」、本文に新しい時刻を書いて送ってください。
71
72 キャンセルする場合は、hogefuga@*****.a4w.ise.osaka-sandai.ac.jp 宛に
73 Subjectに「D${datenum}」と書いて空メールを送って下さい。
74 (大文字と小文字は区別されません。また、半角で入力して下さい。)
75
76 ***本メールは自動で作成されたメールです。返信しないでください。***
77
78 ---
79 Tatsuya Koyabu
80 *****@a4w.ise.osaka-sandai.ac.jp
81 .
82 EOF
83 fi
84 fi

```

A.4 随時処理

ソースコード 11 全体リストを生成するスクリプト (listin)

```
1 #!/bin/sh
2
3 # ファイルが存在しない場合は空白を書き込む
4 if [ "$(ls _/home/user/comelablist_*)" = '' ] && [ "$(ls _/home/user/gohomelist_
    *)" = '' ]; then
5 : > /home/user/comelablist
6 : > /home/user/gohomelist
7 exit 0
8 fi
9
10 # 変数定義
11 otcats='cat /home/user/comelablist_* 2>/dev/null '
12 ctcats='cat /home/user/gohomelist_* 2>/dev/null '
13 otlist='cat /home/user/comelablist 2>/dev/null '
14 ctlist='cat /home/user/gohomelist 2>/dev/null '
15
16 # エラー処理
17 if test "$otcats" = "$otlist" -a "$ctcats" = "$ctlist"; then
18 exit 0
19 elif test "$otcats" = "" -a "$ctcats" = ""; then
20 : > /home/user/comelablist
21 : > /home/user/gohomelist
22 elif test "$otcats" = "" -a "$ctcats" != ""; then
23 : > /home/user/comelablist
24 cat /home/user/gohomelist_* | uniq > /home/user/gohomelist
25 elif test "$ctcats" = "" -a "$otcats" = ""; then
26 cat /home/user/comelablist_* | uniq > /home/user/comelablist
27 : > /home/user/gohomelist
28 else
29 # 通常処理
30 cat /home/user/comelablist_* | uniq > /home/user/comelablist
31 cat /home/user/gohomelist_* | uniq > /home/user/gohomelist
32 fi
```

ソースコード 12 開閉時刻変更通知メールの自動送信を行うスクリプト (cmail)

```
1 #!/bin/sh
2 # 戸締りおよびオープン時刻が変更された際にメールを送るスクリプト
```

```

3 # 文字列によって処理を分ける
4 # C*:戸締め O*:オープン
5
6 case $1 in
7 "CTL" )
8 # 戸締め時刻が遅くなった場合 (Closing up Time Late)
9 msg0="Lab_Today's_Closing_up_time_is_delayed."
10 msg1="本日の戸締め予定時刻が遅くなりました。"
11 msg2="戸締め予定時刻が${2}から${3}へと変更になっています。" ;;
12 "CTE" )
13 # 戸締め時刻が早くなった場合 (Closing up Time Early)
14 msg0="Lab_Today's_Closing_up_time_has_become_faster."
15 msg1="本日の戸締め予定時刻が早くなりました。"
16 msg2="戸締め予定時刻が${2}から${3}へと前倒しになっています。ご注意ください。" ;;
17 "OTL" )
18 # 開く時刻が遅くなった場合 (Open Time Late)
19 msg0="Lab_Today's_Open_time_was_delayed."
20 msg1="本日の研究室が開く予定の時刻が遅くなりました。"
21 msg2="開く予定の時刻が${2}から${3}へと変更になっています。" ;;
22 "OTE" )
23 # 開く予定の時刻が早くなった場合 (Open Time Early)
24 msg0="Lab_Today's_Open_time_has_become_faster."
25 msg1="本日の研究室が開く予定の時刻が早くなりました。"
26 msg2="開く予定の時刻が${2}から${3}へと前倒しになっています。" ;;
27 "OCTR" )
28 # 開く予定の時刻と戸締め予定時刻が既に全体リストから割り出せるやうになっている
    場合 (Open/Closing up Time Reference)
29 msg0="Lab_Today's_Open/Closing_up_time_schedule."
30 msg1="本日午前9時現在での研究室が開く予定/戸締め予定時刻をお知らせします。"
31 msg2="本日の開く予定時刻は${2}、戸締め予定時刻は${3}です。(あくまで予定ですので、
    変わることがあります。)" ;;
32 "OCTS" )
33 # 開く予定の時刻と戸締め予定時刻が全体リストに新たに書き込まれた場合 (Open/
    Closing up Time Set)
34 msg0="Lab_Today's_Open/Closing_up_time_set."
35 msg1="本日の研究室が開く予定/戸締め予定時刻が設定されました。"
36 msg2="本日の開く予定時刻は${2}、戸締め予定時刻は${3}です。(あくまで予定ですので、
    変わることがあります。)" ;;
37 "OCTC" )

```

```

38 # 開く予定の時刻と戸締り予定時刻が全体リストから消えた場合 (Open/Closing up
    Time Canceled)
39 msg0="Lab Today's Open/Closing up time Canceled."
40 msg1="本日の研究室が開く予定/戸締り予定時刻が取り消されました。"
41 msg2="本日は研究室は開かないかもしれません。" ;;
42 esac
43
44 sendmail -t << EOF
45 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
46 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
47 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
48 Subject: ${msg0} ('date +%Y/%m/%d_%H:%M:%S')
49
50 研究室の開く予定/戸締り予定時刻についてお知らせします。
51
52 ----->&----->&----->&-----
53 $msg1
54 $msg2
55 ----->&----->&----->&-----
56
57 ***本メールは自動で作成されたメールです。返信しないでください。***
58
59 ---
60 Tatsuya Koyabu
61 *****@a4w.ise.osaka-sandai.ac.jp
62 .
63 EOF

```

ソースコード 13 開閉時刻表示用 Web ページの自動更新を行うスクリプト (comehtml)

```

1 #!/bin/sh
2 # 変数定義
3 closetime='cat /home/user/gohomelist | grep -o "[0-2][0-9]:[0-5][0-9]" | sort
    -r | uniq | head -1 2>/dev/null '
4 cdate='date +%Y/%m/%d'
5
6 opentime='cat /home/user/comelablist | grep -o "[0-2][0-9]:[0-5][0-9]" | sort
    | uniq | head -1 2>/dev/null '
7
8 otmp='cat /home/user/otextra 2>/dev/null '
9 ctmp='cat /home/user/ctextra 2>/dev/null '
10 ctime='echo ${closetime} | sed 's/://''

```

```

11 ctmpt='echo ${ctmp} | sed 's/://''
12 otime='echo ${opentime} | sed 's/://''
13 otmpt='echo ${otmp} | sed 's/://''
14
15 if test "$otime" != "" -a "$ctime" != "" -a "$otmp" = "" -a "$ctmp" = "";then
16 # 一時ファイルに時刻がなく、かつその日初めてのメールが来た場合にお知らせメール
    を投げる(Open/Closing up Time Set)
17 sh /home/user/cmail OCTS ${opentime} ${closetime}
18 fi
19
20 if test "$otime" = "" -a "$ctime" = "" -a "$otmp" != "" -a "$ctmp" != "";then
21 # 一時ファイルに時刻があり、全体リストから時刻が消えた場合にお知らせメールを投
    げる(Open/Closing up Time Canceled)
22 sh /home/user/cmail OCTC
23 fi
24
25 # 戸締め時刻が早くなったらメール送信(Closing up Time Early)
26 if test "$ctmp" != "" -a "$ctmp" != "$closetime" -a "$ctime" -lt "$ctmpt";then
27 sh /home/user/cmail CTE ${ctmp} ${closetime}
28 elif test "$ctmp" != "" -a "$ctmp" != "$_closetime" -a "$ctime" -gt "$ctmpt";
    then
29 # 戸締め時刻が遅くなったらメール送信(Closing up Time Late)
30 sh /home/user/cmail CTL ${ctmp} ${closetime}
31 fi
32 # オープン時刻が早くなったらメール送信(Open Time Early)
33 if test "$otmp" != "" -a "$otmp" != "$opentime" -a "$otime" -lt "$otmpt";then
34 sh /home/user/cmail OTE ${otmp} ${opentime}
35 elif test "$otmp" != "" -a "$otmp" != "$opentime" -a "$otime" -gt "$otmpt";
    then
36 # オープン時刻が遅くなったらメール送信(Open Time Late)
37 sh /home/user/cmail OTL ${otmp} ${opentime}
38 fi
39
40 if test "$ctmp" = "" -o "$closetime" != "$ctmp";then
41 # 戸締め一時ファイルがなければつくる
42 echo "$closetime" > /home/user/ctextra
43 fi
44
45 if test "$otmp" = "" -o "$opentime" != "$otmp";then
46 # オープン一時ファイルがなければ作る

```

```

47 echo "$opentime" > /home/user/otextra
48 fi
49
50 if test "$otmp" = "$opentime" -a "$ctmp" = "$closetime";then
51 # 一時ファイルと戸締り時刻が同じであれば更新しない
52 echo "date_+"%Y/%m/%d %H:%M%S" "_opentime_no_change." >> /home/user/ot.log
53 echo "date_+"%Y/%m/%d %H:%M%S" "_closetime_no_change." >> /home/user/ct.log
54 exit 0
55 elif test "$otmp" = "$opentime" -a "$ctmp" != "$closetime";then
56 echo "date_+"%Y/%m/%d %H:%M%S" "_opentime_no_change." >> /home/user/ot.log
57 elif test "$ctmp" = "$closetime" -a "$otmp" != "$opentime";then
58 echo "date_+"%Y/%m/%d %H:%M%S" "_closetime_no_change." >> /home/user/ct.log
59 fi
60
61 # 最終更新時刻表示用変数
62 ldate="<i>Last_updated_`date_+"%Y/%m/%d %H:%M%S" `</i><br>"
63
64 # エラー処理
65 if test "$closetime" = "" -o "$opentime" = "";then
66 # ファイルがないなどの理由により変数が nullな場合の処理
67 otmsg="<h2><b>オープン時刻取得失敗</b></h2>"
68 ctmsg="<h2><b>戸締り時刻取得失敗</b></h2>"
69 opentime="Acquisition_failure."
70 closetime="Acquisition_failure."
71 echo "date_+"%Y/%m/%d %H:%M%S" "_opentime_error." >> /home/user/ot.log
72 echo "date_+"%Y/%m/%d %H:%M%S" "_closetime_error." >> /home/user/ct.log
73 else
74 # 通常時処理
75 # 計算用変換式
76 ctime2='echo $closetime | sed s/://g'
77 # 21:45より遅ければメッセージを追加表示
78 if [ "2145" -le "$ctime2" ]; then
79     cmsg="<h2><b>22:00までに大学の敷地外へ出ませう。</b></h2>"
80 #else
81 # cmsg=""
82 fi
83
84 # 21:50より遅い時刻があった場合は21:50を強制表示
85 if [ "2150" -le "$ctime2" ]; then
86     closetime="21:50"

```

```

87  fi
88  # エラーが発生しなければ各種処理後のオープン時刻を表示
89  otmsg="<h2><b>研究室は$_$opentime_に開く予定です。</b></h2>"
90  echo "`date_+"`%Y/%m/%d %H:%M%S" `$_$opentime_`success.`$_$opentime` >> /home/user
    /ot.log
91  # エラーが発生しなければ各種処理後の戸締り時刻を表示
92  ctmsg="<h2><b>戸締り時刻は$_$closetime_の予定です。</b></h2>"
93  echo "`date_+"`%Y/%m/%d %H:%M%S" `$_$closetime_`success.`$_$closetime` >> /home/
    user/ct.log
94  fi
95
96  cat << EOF > /home/user/keylocklab.html
97  <!DOCTYPE html>
98  <html>
99  <head>
100     <meta charset="UTF-8">
101     <title> 15706 $cdate Opentime $opentime / Closing up time $closetime</
        title>
102     <link rel="stylesheet" type="text/css" href="./comelab.css">
103  </head>
104  <body>
105     <h1>情報教育システム研究室 $cdate のオープン/戸締り時刻</h1>
106     $otmsg
107     $ctmsg
108     $cmsg
109     <hr>
110     $ldate
111     <i>Copyright&copy; 2018 T-Koyabu.</i>
112     <br>
113     <a href="list.html">Go List.</a>
114  </body>
115 </html>
116 EOF

```

ソースコード 14 時刻リスト表示用 Web ページの自動更新を行うスクリプト (grepmember)

```

1  #!/bin/sh
2  # 変数定義
3  comefiles='echo "`$(grep_`-H_`" `$_$home/user/comelablist_*_2>/dev/null)`' `
4  gohomefiles='echo "`$(grep_`-H_`" `$_$home/user/gohomelist_*_2>/dev/null)`' `
5
6  cltmp='cat /home/user/clextra 2>/dev/null `

```

```

7 ghtmp='cat /home/user/ghextra 2>/dev/null '
8 if test "$ghtmp" = "" -o "$gohomefiles" != "$ghtmp";then
9 # 帰る時刻一時ファイルがなければつくる
10 echo "$gohomefiles" > /home/user/ghextra
11 fi
12
13 if test "$cltmp" = "" -o "$comefiles" != "$cltmp";then
14 # 来る時刻一時ファイルがなければ作る
15 echo "$comefiles" > /home/user/clextra
16 fi
17
18 if test "$cltmp" = "$comefiles" -a "$ghtmp" = "$gohomefiles";then
19 # 一時ファイルとリストが同じであれば更新しない
20 echo "date_+"%Y/%m/%d %H:%M%S" "_cometimetelist_no_change." >> /home/user/
    cllist.log
21 echo "date_+"%Y/%m/%d %H:%M%S" "_gohometimetelist_no_change." >> /home/user/
    ghlist.log
22 exit 0
23 elif test "$cltmp" = "$comefiles" -a "$ghtmp" != "$gohomefiles";then
24 echo "date_+"%Y/%m/%d %H:%M%S" "_cometimetelist_no_change." >> /home/user/
    cllist.log
25 elif test "$ghtmp" = "$gohomefiles" -a "$cltmp" != "$comefiles";then
26 echo "date_+"%Y/%m/%d %H:%M%S" "_gohometimetelist_no_change." >> /home/user/
    ghlist.log
27 fi
28
29 # エラ - 処理
30 if test "$comefiles" = "" -a "$gohomefiles" = "";then
31 comefiles='echo "取得失敗" '
32 echo "date_+"%Y/%m/%d %H:%M%S" "_cometimetelist_error." >> /home/user/cllist.
    log
33 gohomefiles='echo "取得失敗" '
34 echo "date_+"%Y/%m/%d %H:%M%S" "_gohometimetelist_error." >> /home/user/
    ghlist.log
35
36 elif test "$comefiles" = "";then
37 # 来る時刻取得エラ - 処理
38 comefiles='echo "取得失敗" '
39 echo "date_+"%Y/%m/%d %H:%M%S" "_cometimetelist_error." >> /home/user/cllist.
    log

```

```

40
41 elif test "$gohomefiles" = "" ;then
42 # 帰る時刻取得エラー処理
43 gohomefiles='echo "取得失敗"'
44 echo "date_+"%Y/%m/%d %H:%M%S" "_gohometimelist_error." >> /home/user/
    ghlist.log
45 else
46 echo "date_+"%Y/%m/%d %H:%M%S" "_cometimelist_success." >> /home/user/
    cllist.log
47 echo "date_+"%Y/%m/%d %H:%M%S" "_gohometimelist_success." >> /home/user/
    ghlist.log
48 fi
49 sedname()
50 {
51 cat - |
52 sed -e "s/$/<br>/g" |
53 # フルパス削除
54 sed -e "s/\/home\/user\/\/g" |
55 sed -e "s/comelablist_/g" |
56 sed -e "s/gohomelist_/g" |
57 # 置換処理(a4w)
58 # sed -e "s/Ito.Hirofumi:/49H039 | /" |
59 # sed -e "s/INO.Tadataka:/49H041 | /" |
60 # sed -e "s/UTAGAWA.Hiroshige:/49H045 | /" |
61 # sed -e "s/KATSUSHIKA.Hokusai:/49H055 | /" |
62 # sed -e "s/KINOSHITA.Toukichiro:/49H061 | /" |
63 # sed -e "s/KOBAYASHI.Ichizo:/49H106 | /" |
64 # sed -e "s/SHOTOKU.Taishi:/49H086 | /" |
65 # sed -e "s/NAKATOMINO.Kamatari:/49H106 | /" |
66 # sed -e "s/HATTORI.Hanzo:/49H114 | /" |
67 # sed -e "s/MATSUO.Basho:/49H133 | /" |
68 # 置換処理(ge)
69 #sed -e "s/s\([0-9]\)\([0-9]\)h\([0-9]\)\([0-9]\)\([0-9]\):/1\2H\3\4\5 | /g
    "
70 sed -e "s/s\([0-9]\)\([0-9]\)\([a-z]\)\([0-9]\)\([0-9]\)\([0-9]\):/1\2\U
    \3\4\5\6_|_/g"
71 }
72
73 if test "$comefiles" = "取得失敗" ;then
74 comelist='echo "$comefiles"'

```

```

75 else
76 comelist='echo "$comefiles" | sedname | sort | uniq '
77 fi
78
79 if test "$gohomefiles" = "取得失敗";then
80 gohomelist='echo "$gohomefiles" '
81 else
82 gohomelist='echo "$gohomefiles" | sedname | sort | uniq '
83 fi
84 nowday='date +%Y/%m/%d '
85 ldate=" <i>Last updated </i>"+%Y/%m/%d %H:%M%S" '</i><br>'
86
87 cat << EOF > /home/user/list.html
88 <!DOCTYPE html>
89 <html>
90   <head>
91     <meta charset="UTF-8">
92     <title> 15706 $nowday Lab Come/GoHome time List.</title>
93     <link rel="stylesheet" type="text/css" href="./comelab.css">
94   </head>
95   <body>
96     <h1>情報教育システム研究室 $nowday 時刻リスト</h1>
97     <table>
98       <tr>
99         <th><h2>来る 予定時刻</h2></th>
100        <th><h2>帰る 予定時刻</h2></th>
101      </tr>
102      <tr>
103        <th><h2> $comelist </h2></th>
104        <th><h2> $gohomelist </h2></th>
105      </tr>
106    </table>
107    <hr>
108    $ldate
109    <i>Copyright&copy; 2018 T-Koyabu.</i>
110    <br>
111    <a href="index.html">Go Toppage.</a>
112  </body>
113 </html>
114 EOF

```

ソースコード 15 エラーメールを自動送信するスクリプト (errors)

```

1  #!/bin/sh
2  # エラーの場合メールを送るスクリプト
3  # エラーコードを引数にして処理を分ける
4  # 10番台:登録 20番台:変更 30番台:削除 40番台:特殊
5
6  case $1 in
7  "E10" )
8  # 締切以降の登録メール受信
9  msg0="Deadline_Over."
10 msg1="本日の時刻の登録は17時45分(JST)で締め切っています。"
11 msg2="締切時刻以降にメールを送っても処理されません。"
12 msg3="よって、送られた時刻は破棄されました。" ;;
13 "E11" )
14 # 受付時刻より前のメール受信
15 msg0="Before_reception_start_time."
16 msg1="翌日の時刻の受付は20時00分(JST)からです。"
17 msg2="20時00分(JST)以降に再度送信してください。"
18 msg3="現在の時刻は‘date_+%H時%M分‘です。" ;;
19 "E12" )
20 # 日曜にメール受信
21 msg0="Today_is_Sunday."
22 msg1="今日は日曜日です。研究室は開いていない(はずの)日です。"
23 msg2="日曜日はメールの受付をしていません。"
24 msg3="よって、送られた時刻は破棄されました。" ;;
25 "E13" )
26 # 未来の日付を指定してメール受信の際、その日付が日曜だった
27 msg0="That_day_is_Sunday."
28 msg1="指定された日は日曜日です。研究室は開いていない(はずの)日です。"
29 msg2="日曜日にあたる日付の指定はできません。"
30 msg3="よって、送られた時刻は破棄されました。" ;;
31 "E14" )
32 # 翌日より1週間以上先の日付を指定
33 msg0="Date_specification_error."
34 msg1="翌日から1週間より先の日付は指定できません。"
35 msg2="日付を確認し、送り直してください。"
36 msg3="(‘date_—date_’tomorrow’_+%Y/%m/%d‘から‘date_—date_’tomorrow_7days’_+%Y
    /%m/%d‘まで指定可能です。)" ;;
37 "E15" )
38 # 過去の日付を指定

```

39 msg0=" Date_specification_error."

40 msg1=" 過去の日付は指定できません。"

41 msg2=" 日付を確認し、送り直してください。"

42 msg3=" ('date____date_'tomorrow' _+%Y/%m/%d' から 'date____date_'tomorrow_7days' _+%Y
 /%m/%d'まで指定可能です。)" ;;

43 "E16")

44 # 21時51分以降の時刻を指定

45 msg0=" Sent_time_was_too_late!"

46 msg1=" 21時51分(JST)以降の時刻は指定できません。"

47 msg2=" 21時50分(JST)までの時刻を書いて送り直してください。"

48 msg3=" よって、送られた時刻は破棄されました。" ;;

49 "E20")

50 # 締切以降の変更メール受信

51 msg0=" Deadline_Over."

52 msg1=" 本日の時刻の変更は17時45分(JST)で締め切っています。"

53 msg2=" 締切時刻以降にメールを送っても処理されません。"

54 msg3=" よって、送られた時刻は破棄されました。" ;;

55 "E21")

56 # ファイル不存在エラー(当日)

57 msg0=" 404_Today's_File_Not_Found!"

58 msg1=" 本日分の時刻リストファイルが存在しません。"

59 msg2=" まずは時刻の登録を行ってください。"

60 msg3=" よって、送られた時刻は破棄されました。" ;;

61 "E22")

62 # ファイル不存在エラー(未来の日付)

63 msg0=" 404_Since_tomorrow_File_Not_Found!"

64 msg1=" 指定された日付の時刻リストファイルが存在しません。"

65 msg2=" まずは時刻の登録を行ってください。"

66 msg3=" よって、送られた時刻は破棄されました。" ;;

67 "E23")

68 # 21時51分以降の時刻を指定

69 msg0=" Sent_time_was_too_late!"

70 msg1=" 21時51分(JST)以降の時刻は指定できません。"

71 msg2=" 21時50分(JST)までの時刻を書いて送り直してください。"

72 msg3=" よって、送られた時刻は破棄されました。" ;;

73 "E30")

74 # 締切以降のキャンセルメール受信

75 msg0=" Deadline_Over."

76 msg1=" 本日送られた時刻のキャンセルは17時45分(JST)で締め切っています。"

77 msg2=" 締切時刻以降にメールを送っても処理されません。"

78 msg3=" よって、キャンセルの処理は行われませんでした。" ; ;
79 " E32")
80 # 日曜にメール受信
81 msg0=" Today_is_Sunday."
82 msg1=" 本日は日曜日です。研究室は開いていない(はずの)日です。"
83 msg2=" 日曜日はメールの受付をしていません。"
84 msg3=" よって、キャンセルの処理は行われませんでした。" ; ;
85 " E33")
86 # 未来の日付を指定してメール受信の際、その日付が日曜だった
87 msg0=" That_day_is_Sunday."
88 msg1=" 指定された日は日曜日です。研究室は開いていない(はずの)日です。"
89 msg2=" 日曜日にあたる日付の指定はできません。"
90 msg3=" よって、キャンセルの処理は行われませんでした。" ; ;
91 " E34")
92 # ファイル不存在エラー(当日)
93 msg0=" 404_Today's_File_Not_Found!"
94 msg1=" 本日分の時刻リストファイルが存在しません。"
95 msg2=" まずは時刻の登録を行ってください。"
96 msg3=" よって、キャンセルの処理は行われませんでした。" ; ;
97 " E35")
98 # ファイル不存在エラー(未来の日付)
99 msg0=" 404_Since_tomorrow_File_Not_Found!"
100 msg1=" 指定された日付の時刻リストファイルが存在しません。"
101 msg2=" まずは時刻の登録を行ってください。"
102 msg3=" よって、キャンセルの処理は行われませんでした。" ; ;
103 " E40")
104 # 2038年問題への対応
105 msg0=" Year_2038_problem."
106 msg1=" 2038年1月19日より先の日付はシステム側で正常に読み取れません。(2038年問題)"
107 msg2=" 日付を確認し、送り直してください。"
108 msg3=" ('date_—date_'tomorrow' _+%Y/%m/%d' から 'date_—date_'tomorrow_7days' _+%Y/%m/%d' まで指定可能です。)" ; ;
109 " E41")
110 # ファイル不存在エラー(当日)
111 msg0=" 404_Today's_File_Not_Found!"
112 msg1=" 本日分の時刻リストファイルが存在しません。"
113 msg2=" まずは時刻の登録を行ってください。"
114 msg3=" ファイルが存在しない為、時刻の参照はできません。" ; ;
115 " E42")

```

116 # ファイル不存在エラー(未来の日付)
117 msg0=" 404_Since_tomorrow_File_Not_Found!"
118 msg1=" 指定された日付の時刻リストファイルが存在しません。"
119 msg2=" まずは時刻の登録を行ってください。"
120 msg3=" ファイルが存在しない為、時刻の参照はできません。";
121 "E43" )
122 # 来る時刻と帰る時刻が同一
123 msg0=" Lab_Cometime_&_Gohometime_are_Sametime!"
124 msg1=" 何らかの理由により、来る時刻と帰る時刻に同じ時刻が設定されています。"
125 msg2=" 同一の時刻は設定できません。"
126 msg3=" 時刻はすべて1バイト文字で書いてください。2バイト文字では処理できません。";
127 esac
128
129 sendmail -t << EOF
130 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
131 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
132 To: $2
133 Subject: ${msg0} ('date +%Y/%m/%d_%H:%M:%S' )
134
135 次の理由により、処理が実行されませんでした。
136
137 ----->&----->&----->&-----
138 $msg1
139 $msg2
140 $msg3
141 ----->&----->&----->&-----
142
143 ***本メールは自動で作成されたメールです。返信しないでください。***
144
145 ---
146 Tatsuya Koyabu
147 *****@a4w.ise.osaka-sandai.ac.jp
148 .
149 EOF

```

A.5 指定時刻処理

ソースコード 16 午前9時に開閉時刻表示用 Web ページの自動更新を行うスクリプト (comehtml0900)

```
1 #!/bin/sh
2 # 変数定義
3 closetime='cat /home/user/gohomelist | grep -o "[0-2][0-9]:[0-5][0-9]" | sort
   -r | head -1 2>/dev/null '
4 cdate='date +%Y/%m/%d'
5
6 opentime='cat /home/user/comelablist | grep -o "[0-2][0-9]:[0-5][0-9]" | sort |
   head -1 2>/dev/null '
7
8 otmp='cat /home/user/otextra 2>/dev/null '
9 ctmp='cat /home/user/ctextra 2>/dev/null '
10 if test "$opentime" != "" -a "$closetime" != "";then
11 # 午前9時現在でリストに時刻がある場合にお知らせメールを投げる (Open/Closing up
   Time Registered)
12 sh /home/user/cmail OCTR ${opentime} ${closetime}
13 fi
14 if test "$ctmp" = "" -o "$closetime" != "$ctmp";then
15 # 戸締り一時ファイルがなければ作る
16 echo "$closetime" > /home/user/ctextra
17 chown user:user /home/user/ctextra
18 fi
19
20 if test "$otmp" = "" -o "$opentime" != "$otmp";then
21 # オープン一時ファイルがなければ作る
22 echo "$opentime" > /home/user/otextra
23 chown user:user /home/user/otextra
24 fi
25
26
27 # 最終更新時刻表示用変数
28 ldate=" <i>Last updated </i>'date +%Y/%m/%d %H:%M%S' </i><br>"
29
30 # エラー処理
31 if test "$closetime" = "" -o "$opentime" = "";then
32 # ファイルがないなどの理由により変数が nullな場合の処理
33 otmsg=" <h2><b>オープン時刻取得失敗</b></h2>"
```

```

34  ctmsg=" _____<h2><b>戸締り時刻取得失敗</b></h2>"
35  opentime=" Acquisition_failure."
36  closetime=" Acquisition_failure."
37  echo " `date +%Y/%m/%d %H:%M%S` "opentime_error." >> /home/user/ot.log
38  echo " `date +%Y/%m/%d %H:%M%S` "closetime_error." >> /home/user/ct.log
39  else
40  # 通常時処理
41  # 計算用変換式
42  ctime2='echo $closetime | sed s/://g`
43  # 21:45より遅ければメッセージを追加表示
44  if [ "2145" -le "$ctime2" ]; then
45      cmsg=" _____<h2><b>22:00までに大学の敷地外へ出ませう。</b></h2>"
46  #else
47      # cmsg=""
48  fi
49
50  # 22:00より遅い時刻があった場合は21:50を強制表示
51  if [ "2200" -le "$ctime2" ]; then
52      closetime=" 21:50"
53  fi
54  # エラーが発生しなければ各種処理後のオープン時刻を表示
55  otmsg=" _____<h2><b>研究室は_$opentime_に開く予定です。</b></h2>"
56  echo " `date +%Y/%m/%d %H:%M%S` "opentime_success._$opentime" >> /home/user
    /ot.log
57  # エラーが発生しなければ各種処理後の戸締り時刻を表示
58  ctmsg=" _____<h2><b>戸締り時刻は_$closetime_の予定です。</b></h2>"
59  echo " `date +%Y/%m/%d %H:%M%S` "closetime_success._$closetime" >> /home/
    user/ct.log
60  fi
61
62  cat << EOF > /home/user/keylocklab.html
63  <!DOCTYPE html>
64  <html>
65  <head>
66      <meta charset="UTF-8">
67      <title> 15706 $date Opentime $opentime / Closing up time $closetime</
        title>
68      <link rel="stylesheet" type="text/css" href="./comelab.css">
69  </head>
70  <body>

```

```

71     <h1>情報教育システム研究室 $cdate のオープン/戸締り時刻</h1>
72 $otmsg
73 $ctmsg
74     $cmsg
75     <hr>
76     $ldate
77     <i>Copyright&copy; 2018 T-Koyabu.</i>
78     <br>
79     <a href="list.html">Go List.</a>
80 </body>
81 </html>
82 EOF

```

ソースコード 17 午後 6 時に開閉時刻表示用 Web ページの自動更新を行い、施錠時刻を確定させるスクリプト (comehtml1800)

```

1  #!/bin/sh
2  # 変数定義
3  closetime='cat /home/user/gohomelist | grep -o "[0-2][0-9]:[0-5][0-9]" | sort
    -r | uniq | head -1 2>/dev/null '
4  cdate='date +%Y/%m/%d'
5
6  opentime='cat /home/user/comelablist | grep -o "[0-2][0-9]:[0-5][0-9]" | sort
    | uniq | head -1 2>/dev/null '
7
8  otmp='cat /home/user/otextra 2>/dev/null '
9  ctmp='cat /home/user/ctextra 2>/dev/null '
10 if test "$ctmp" = "" -o "$closetime" != "$ctmp";then
11 # 戸締り一時ファイルがなければ作る
12 echo "$closetime" > /home/user/ctextra
13 fi
14
15 if test "$otmp" = "" -o "$opentime" != "$otmp";then
16 # オープン一時ファイルがなければ作る
17 echo "$opentime" > /home/user/otextra
18 fi
19
20 # 現在時刻取得
21 nowtime='date +%H%M'
22 today='date +%Y/%m/%d'
23 # 最終更新時刻表示用変数
24 ldate=" <i>Last updated <code>date +%Y/%m/%d %H:%M%S</code>" </i><br>"

```

```

25
26 # エラー処理
27 if test "$closetime" = "" -o "$opentime" = "";then
28 # ファイルがないなどの理由により変数が nullな場合の処理
29 otmsg=" .....<h2><b>オープン時刻取得失敗</b></h2>"
30 ctmsg=" .....<h2><b>戸締り時刻取得失敗</b></h2>"
31 opentime=" Acquisition_failure."
32 closetime=" Acquisition_failure."
33 echo " `date +%Y/%m/%d %H:%M%S` `_opentime_error.`" >> /home/user/ot.log
34 echo " `date +%Y/%m/%d %H:%M%S` `_closetime_error.`" >> /home/user/ct.log
35 else
36 # 通常時処理
37 # 計算用変換式
38 ctime2=`echo $closetime | sed s/://g`
39 # 21:45より遅ければメッセージを追加表示
40 if [ "2145" -le "$ctime2" ]; then
41     cmsg=" ..<h2><b>22:00までに大学の敷地外へ出ませう。</b></h2>"
42 else
43     cmsg=""
44 fi
45
46 # 21:50より遅い時刻があった場合は21:50を強制表示
47 if [ "2150" -le "$ctime2" ]; then
48     closetime=" 21:50"
49 fi
50 # エラーが発生しなければ各種処理後のオープン時刻を表示
51 otmsg=" .....<h2><b>研究室は`_opentime`に開く予定です。</b></h2>"
52 echo " `date +%Y/%m/%d %H:%M%S` `_opentime_success.`_opentime" >> /home/user
    /ot.log
53 # エラーが発生しなければ各種処理後の戸締り時刻を表示
54 ctmsg=" .....<h2><b>戸締り時刻は`_closetime`の予定です。</b></h2>"
55 echo " `date +%Y/%m/%d %H:%M%S` `_closetime_success.`_closetime" >> /home/
    user/ct.log
56 fi
57 diffclose=`echo "1800" `
58 cat << EOF > /home/user/keylocklab.html
59 <!DOCTYPE html>
60 <html>
61 <head>
62 <meta charset="UTF-8">

```

```

63     <title> 15706 $cdate Opentime $opentime / Closing up time $closetime</
        title>
64     <link rel="stylesheet" type="text/css" href="./comelab.css">
65 </head>
66 <body>
67     <h1>情報教育システム研究室 $cdate のオープン/戸締り時刻</h1>
68 $otmsg
69 $ctmsg
70 $cmsg
71     <hr>
72     $ldate
73     <i>Copyright&copy; 2018 T-Koyabu.</i>
74     <br>
75     <a href="list.html">Go List.</a>
76 </body>
77 </html>
78 EOF
79
80 if test "$closetime" = "Acquisition_failure.";then
81 # 戸締り時刻取得失敗時にエラーメッセージ入りメールを送信
82 sendmail -t << EOF
83 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
84 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
85 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
86 Subject: Closing up time Confirmed. $today
87
88 情報教育システム研究室の戸締り予定時刻をお知らせします。
89 —————>&—————>&—————>&—————>&—————
90 本日の戸締り予定時刻はメールが一通も来なかったか、取得に失敗しているため、表示
されません。
91 —————>&—————>&—————>&—————>&—————
92
93 返信は17時45分で締め切りました。
94
95 ***本メールは自動で作成されたメールです。返信しないでください。***
96
97 —
98 Tatsuya Koyabu
99 *****@a4w.ise.osaka-sandai.ac.jp
100 .

```

```
101 EOF
102 elif test "$ctime2" -le "$diffclose";then
103 # 18:00より前に戸締り予定時刻が設定されている場合の処理
104 sendmail -t << EOF
105 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
106 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
107 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
108 Subject:Closing up time Confirmed. $today
109
110 情報教育システム研究室の戸締り予定時刻をお知らせします。
111 ----->&----->&----->&----->&----->&-----
112 本日の戸締り予定時刻は $closetime ですので
113 すでに閉まっている可能性があります。
114 ----->&----->&----->&----->&----->&-----
115
116 返信は17時45分で締め切りました。
117
118 ***本メールは自動で作成されたメールです。返信しないでください。***
119
120 ---
121 Tatsuya Koyabu
122 *****@a4w.ise.osaka-sandai.ac.jp
123 .
124 EOF
125 else
126 # 通常時のメール送信
127 sendmail -t << EOF
128 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
129 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
130 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
131 Subject:Closing up time Confirmed. $today
132
133 情報教育システム研究室の戸締り予定時刻をお知らせします。
134 ----->&----->&----->&----->&----->&-----
135 本日の戸締り予定時刻は $closetime です。
136 ----->&----->&----->&----->&----->&-----
137
138 返信は17時45分で締め切りました。
139
140 ***本メールは自動で作成されたメールです。返信しないでください。***
```

```

141
142 —
143 Tatsuya Koyabu
144 *****@a4w.ise.osaka-sandai.ac.jp
145 .
146 EOF
147 fi

```

ソースコード 18 午前 9 時と午後 6 時に時刻リスト表示用 Web ページの自動更新を行うスクリプト (grepmember0900)

```

1  #!/bin/sh
2  # 変数定義
3  comefiles='echo "$(grep _H_ " _/home/user/comelablist_*)" '
4  gohomefiles='echo "$(grep _H_ " _/home/user/gohomelist_*)" '
5
6  cltmp='cat /home/user/clextra '
7  ghtmp='cat /home/user/ghextra '
8  if test "$ghtmp" = "" -o "$gohomefiles" != "$ghmp";then
9  # 帰る時刻一時ファイルがなければつくる
10  echo "$gohomefiles" > /home/user/ghextra
11  chown user:user /home/user/ghextra
12  fi
13
14  if test "$cltmp" = "" -o "$comefiles" != "$cltmp";then
15  # 来る時刻一時ファイルがなければ作る
16  echo "$comefiles" > /home/user/clextra
17  chown user:user /home/user/clextra
18  fi
19
20 # エラー処理
21 if test "$comefiles" = "grep: _/home/user/comelablist_*: _No_such_file_or_
    directory" -o "$comefiles" = "";then
22
23  comefiles='echo "取得失敗" '
24  echo " `date +%Y/%m/%d %H:%M%S` " _cometimelist_error." >> /home/user/cclist.
    log
25  else
26  echo " `date +%Y/%m/%d %H:%M%S` " _cometimelist_success." >> /home/user/
    cclist.log
27  fi
28  if test "$gohomefiles" = "grep: _/home/user/gohomelist_*: _No_such_file_or_
    directory" -o "$gohomefiles" = "";then

```

```

29
30 gohomefiles='echo "取得失敗"'
31 echo "date +%Y/%m/%d %H:%M%S" "_gohometimelist_error." >> /home/user/
    ghlist.log
32 else
33 echo "date +%Y/%m/%d %H:%M%S" "_gohometimelist_success." >> /home/user/
    ghlist.log
34 fi
35
36 sedname()
37 {
38 cat - |
39 sed -e "s/$/<br>/g" |
40 # フルパス削除
41 sed -e "s/\/home\/user\/\/g" |
42 sed -e "s/comelablist_/g" |
43 sed -e "s/gohomelist_/g" |
44 # 置換処理(a4w)
45 sed -e "s/ITO.Hirofumi:/49H039_|_/ " |
46 sed -e "s/INO.Tadataka:/49H041_|_/ " |
47 sed -e "s/UTAGAWA.Hiroshige:/49H045_|_/ " |
48 sed -e "s/KATSUSHIKA.Hokusai:/49H055_|_/ " |
49 sed -e "s/KINOSHITA.Toukichiro:/49H061_|_/ " |
50 sed -e "s/KOBAYASHI.Ichizo:/49H106_|_/ " |
51 sed -e "s/SHOTOKU.Taishi:/49H086_|_/ " |
52 sed -e "s/NAKATOMINO.Kamatari:/49H106_|_/ " |
53 sed -e "s/HATTORI.Hanzo:/49H114_|_/ " |
54 sed -e "s/MATSUO.Basho:/49H133_|_/ " |
55 # 置換処理(ge)
56 sed -e "s/s\([0-9]\)\([0-9]\)\([a-z]\)\([0-9]\)\([0-9]\)\([0-9]\):/1\2\U
    \3\4\5\6_|_/g"
57 }
58 if test "$comefiles" = "取得失敗";then
59 comelist='echo "$comefiles"'
60 else
61 comelist='echo "$comefiles" | sedname | sort | uniq'
62 fi
63
64 if test "$gohomefiles" = "取得失敗";then
65 gohomelist='echo "$gohomefiles"'

```

```

66 else
67 gohomelist='echo "$gohomefiles" | sedname | sort | uniq '
68 fi
69 nowday='date +%Y/%m/%d '
70 ldate=" <i>Last updated </i> 'date +%Y/%m/%d %H:%M%S" '</i><br>'
71 cat << EOF > /home/user/list.html
72 <!DOCTYPE html>
73 <html>
74 <head>
75 <meta charset="UTF-8">
76 <title> 15706 $nowday Lab Come/GoHome time List.</title>
77 <link rel="stylesheet" type="text/css" href="./comelab.css">
78 </head>
79 <body>
80 <h1>情報教育システム研究室 $nowday 時刻リスト</h1>
81 <table>
82 <tr>
83 <th><h2>来る 予定時刻</h2></th>
84 <th><h2>帰る 予定時刻</h2></th>
85 </tr>
86 <tr>
87 <th><h2> $comelist </h2></th>
88 <th><h2> $gohomelist </h2></th>
89 </tr>
90 </table>
91 <hr>
92 $ldate
93 <i>Copyright&copy; 2018 T-Koyabu.</i>
94 <br>
95 <a href="index.html">Go Toppage.</a>
96 </body>
97 </html>
98 EOF

```

ソースコード 19 時刻の集計処理を行うスクリプト (todayreports)

```

1 #!/bin/sh
2 # 変数定義
3 today='date +%Y/%m/%d '
4 comelist='cat /home/user/comelablist 2>/dev/null '
5 gohomelist='cat /home/user/gohomelist 2>/dev/null '
6

```

```

7 # 来る時間カウント関数
8 cometimecount()
9 {
10 echo "#来る予定の時刻"
11 for i in `seq -w 7 21`
12 do
13 echo "$i時台: `cat /home/user/comelablist_* | grep "$i" : | wc -l` 人"
14 echo "$i時台: `cat /home/user/comelablist_* | grep "$i" : | wc -l` 人" >> /home/
    user/report/cometimereport `date +%Y%m%d`
15 done
16 # 来る予定時刻レポートのオーナー変更
17 chown user:user /home/user/report/cometimereport `date +%Y%m%d`
18 }
19 # 帰る時間カウント関数
20 gohometimecount()
21 {
22 echo "#帰る予定の時刻"
23 for i in `seq -w 7 21`
24 do
25 echo "$i時台: `cat /home/user/gohomelist_* | grep "$i" : | wc -l` 人"
26 echo "$i時台: `cat /home/user/gohomelist_* | grep "$i" : | wc -l` 人" >> /home/
    user/report/gohometimereport `date +%Y%m%d`
27 done
28 # 帰る予定時刻レポートのオーナー変更
29 chown user:user /home/user/report/gohometimereport `date +%Y%m%d`
30 }
31 # 来る時間人数カウント関数
32 cometimememcount()
33 {
34 echo "今日メールを送ってきたのは `ls /home/user/comelablist_* | wc -l` 人"
35 }
36
37 # 帰る時間人数カウント関数
38 gohometimememcount()
39 {
40 echo "今日メールを送ってきたのは `ls /home/user/gohomelist_* | wc -l` 人"
41 }
42 # レポートのパーミッション変更
43 #chown user:user /home/user/cometimereport `date +%Y%m%d`
44 #chown user:user /home/user/gohometimereport `date +%Y%m%d`

```

```

45 # メール配信(取得エラー時)
46 if test "$comelist" = "" -a "$gohomelist" = "";then
47 sendmail -t << EOF
48 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
49 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
50 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
51 Message-Id: <'date +%Y%m%d'.TimeRep.'date +%s'@*****.a4w.ise.osaka-sandai.ac.
    jp>
52 Subject:Send time Report. $today
53
54 Lab Come/Go Home Time Reports.
55 ----->&----->&----->&-----
56 本日の
    Reportはメールが一通も来なかったか、取得に失敗しているため、表示されません。
57 ----->&----->&----->&-----
58
59 このメールは、本日送られてきたメールに記載されていた時刻の統計をお知らせするも
    の
60 です。
61
62 ***本メールは自動で作成されたメールです。返信しないでください。***
63
64 —
65 Tatsuya Koyabu
66 *****@a4w.ise.osaka-sandai.ac.jp
67 .
68 EOF
69 elif test "$comelist" != "" -a "$gohomelist" = "";then
70 # メール配信(帰る時間取得エラー時)
71 sendmail -t << EOF
72 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
73 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
74 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
75 Message-Id: <'date +%Y%m%d'.TimeRep.'date +%s'@*****.a4w.ise.osaka-sandai.ac.
    jp>
76 Subject:Send time Report. $today
77
78 Lab Come/Go Home Time Reports.
79

```

```

80  _____>&_____>&_____>&_____
81  'cometimecount '
82  _____>&_____>&_____>&_____
83  'cometmemcount '
84
85  _____>&_____>&_____>&_____
86  本日の帰る予定時刻
      Reportはメールが一通も来なかったか、取得に失敗しているため、表示されません。

87  _____>&_____>&_____>&_____
88
89  このメールは、本日送られてきたメールに記載されていた時刻の統計をお知らせするも
      の
90  です。
91
92  ***本メールは自動で作成されたメールです。返信しないでください。***
93
94  —
95  Tatsuya Koyabu
96  *****@a4w.ise.osaka-sandai.ac.jp
97  .
98  EOF
99  elif test "$comelist" = "" -a "$gohomelist" != "";then
100 # メール配信(来る時間取得エラー時)
101 sendmail -t << EOF
102 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
103 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
104 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
105 Message-Id: <'date +%Y%m%d'.TimeRep.'date +%s '@*****.a4w.ise.osaka-sandai.ac.
      jp>
106 Subject:Send time Report. $today
107
108 Lab Come/Go Home Time Reports.
109
110 _____>&_____>&_____>&_____
111 本日の来る予定時刻
      Reportはメールが一通も来なかったか、取得に失敗しているため、表示されません。

112 _____>&_____>&_____>&_____
113

```

```

114 ----->&----->&----->&-----
115 'gohometimecount '
116 ----->&----->&----->&-----
117 'gohometimememcount '
118
119 このメールは、本日送られてきたメールに記載されていた時刻の統計をお知らせするも
120 の
121
122 ***本メールは自動で作成されたメールです。返信しないでください。***
123
124 ---
125 Tatsuya Koyabu
126 *****@a4w.ise.osaka-sandai.ac.jp
127 .
128 EOF
129 else
130 # メール配信(通常処理)
131 sendmail -t << EOF
132 From: hogefuga-request@*****.a4w.ise.osaka-sandai.ac.jp
133 Reply-To: Nobody@*****.a4w.ise.osaka-sandai.ac.jp
134 To: hogefuga@*****.a4w.ise.osaka-sandai.ac.jp
135 Message-Id: <'date +%Y%m%d'.TimeRep.'date +%s '@*****.a4w.ise.osaka-sandai.ac.
136 jp>
137
138 Subject:Send time Report. $today
139
140 Lab Come/Go Home Time Reports.
141 ----->&----->&----->&-----
142 'cometimecount '
143
144 ----->&----->&----->&-----
145 'gohometimecount '
146 ----->&----->&----->&-----
147 'gohometimememcount '
148
149 このメールは、本日送られてきたメールに記載されていた時刻の統計をお知らせするも
150 のです。

```

```
151 ***本メールは自動で作成されたメールです。返信しないでください。***
152
153 —
154 Tatsuya Koyabu
155 *****@a4w.ise.osaka-sandai.ac.jp
156 .
157 EOF
158 fi
```

ソースコード 20 日付変更時に処理を行うスクリプト (listytdmv)

```
1  #!/bin/sh
2
3  # 前日分のリストを日付をつけて保存
4  mv /home/user/comelablist /home/user/list/comelablist `date --date '1 day ago'
   +%Y%m%d`
5  mv /home/user/gohomelist /home/user/list/gohomelist `date --date '1 day ago' +%
   Y%m%d`
6  mkdir /home/user/list/member/`date --date '1 day ago' +%Y%m%d`
7  mv /home/user/comelablist_* /home/user/list/member/`date --date '1 day ago' +%
   Y%m%d`/
8  mv /home/user/gohomelist_* /home/user/list/member/`date --date '1 day ago' +%Y
   %m%d`/
9  # 前日分のログを日付をつけて保存
10 mv /home/user/ct.log /home/user/log/ct `date --date '1 day ago' +%Y%m%d`.log
11 mv /home/user/ot.log /home/user/log/ot `date --date '1 day ago' +%Y%m%d`.log
12 mv /home/user/cclist.log /home/user/log/cclist `date --date '1 day ago' +%Y%m%d
   `.log
13 mv /home/user/ghlist.log /home/user/log/ghlist `date --date '1 day ago' +%Y%m%d
   `.log
14 mv /home/user/clrecieve.log /home/user/log/clrecieve `date --date '1 day ago'
   +%Y%m%d`.log
15 mv /home/user/ghrecieve.log /home/user/log/ghrecieve `date --date '1 day ago'
   +%Y%m%d`.log
16 mv /home/user/cachefree.log /home/user/log/cachefree `date --date '1 day ago'
   +%Y%m%d`.log
17 mv /home/user/vm.log /home/user/log/vm `date --date '1 day ago' +%Y%m%d`.log
18 # 前日分の一時ファイルを削除
19 rm -f /home/user/[co]textra
20 rm -f /home/user/clextra
21 rm -f /home/user/ghextra
22 # 本日分のリストを作成
```

```

23 touch /home/user/comelablist
24 touch /home/user/gohomelist
25 # 前日受付分のリストの名前変更
26 rename s/'date +%Y%m%d'// /home/user/'date +%Y%m%d'comelablist*
27 rename s/'date +%Y%m%d'// /home/user/'date +%Y%m%d'gohomelist*
28 # 本日分のログを作成
29 touch /home/user/ct.log
30 touch /home/user/ot.log
31 touch /home/user/cclist.log
32 touch /home/user/ghlist.log
33 touch /home/user/clrecieve.log
34 touch /home/user/ghrecieve.log
35 touch /home/user/cachefree.log
36 touch /home/user/vm.log
37 # Owner変更処理
38 chown user:user /home/user/comelablist
39 chown user:user /home/user/gohomelist
40 chown user:user /home/user/log/ot'date —date '1 day ago' +%Y%m%d'.log
41 chown user:user /home/user/log/ct'date —date '1 day ago' +%Y%m%d'.log
42 chown user:user /home/user/log/cclist'date —date '1 day ago' +%Y%m%d'.log
43 chown user:user /home/user/log/ghlist'date —date '1 day ago' +%Y%m%d'.log
44 chown user:user /home/user/*.log
45 chown -R user:user /home/user/list/
46 # listのPermission変更
47 chmod 644 /home/user/comelablist
48 chmod 644 /home/user/gohomelist
49 # ファイル初期化
50 : > /home/user/comelablist
51 : > /home/user/gohomelist
52 : > /home/user/ct.log
53 : > /home/user/ot.log
54 : > /home/user/cclist.log
55 : > /home/user/ghlist.log
56 : > /home/user/clrecieve.log
57 : > /home/user/ghrecieve.log
58 : > /home/user/cachefree.log
59 : > /home/user/vm.log
60 # html更新処理
61 # 日付取得
62 cdate='date '+%Y/%m/%d' '

```

```

63 # 最終更新時刻表示用変数
64 ldate=" <i>Last updated 'date'+%Y/%m/%d %H:%M:%S'</i><br>"
65 ctmsg=" <h2><b>毎日 9:00 JST に最初の更新を予定</b></h2>"
66 cat << EOF > /home/user/keylocklab.html
67 <!DOCTYPE html>
68 <html>
69   <head>
70     <meta charset="UTF-8">
71     <title> 15706 $cdate Closing up time is Unacquired</title>
72     <link rel="stylesheet" type="text/css" href=" ./comelab.css">
73   </head>
74   <body>
75     <h1>情報教育システム研究室 $cdate のオープン/戸締り時刻</h1>
76     $ctmsg
77     <hr>
78     $ldate
79     <i>Copyright&copy; 2018 T-Koyabu.</i>
80   </body>
81 </html>
82 EOF
83
84 cat << EOF > /home/user/list.html
85 <!DOCTYPE html>
86 <html>
87   <head>
88     <meta charset="UTF-8">
89     <title> 15706 $cdate Lab Come/GoHome time List.</title>
90     <link rel="stylesheet" type="text/css" href=" ./comelab.css">
91   </head>
92   <body>
93     <h1>情報教育システム研究室 $cdate 時刻リスト</h1>
94     $ctmsg
95     <hr>
96     $ldate
97     <i>Copyright&copy; 2018 T-Koyabu.</i>
98   </body>
99 </html>
100 EOF

```

A.6 Web ページ装飾

ソースコード 21 時刻表示用 Web ページの装飾を行うためのコードを記したスタイルシート (comelab.css)

```
1 /* Copyrights 2018 T-Koyabu. */
2 body {
3     background-color: #000;
4     color: #28FE14;
5 }
6 h1 {
7     /*border-left: double #000000 ; */
8     border-bottom: solid #28FE14;
9 }
10 hr {
11     background-color: #28FE14;
12     color: #28FE14;
13     border: none;
14     height: 3px;
15 }
16 a {
17     color: #28FE14;
18 }
19
20 table {
21     border: 1px #28FE14 solid;
22 }
23
24 tr {
25     border: 1px #28FE14 solid;
26 }
27
28 th {
29     border: 1px #28FE14 solid;
30 }
31
32 #lists {
33     height: 100%;
34     padding: 1em;
35 }
36
```

```
37 iframe {
38 border:none;
39 width:100%;
40 height:100%;
41 padding:0;
42 margin:0;
43 }
```