

大学でのウェブアプリケーションを用いた LifeHacks の検討

06H112 湯川正洋

目次

1	導入	1
2	LifeHacks の応用	2
3	実験概要	3
3.1	40:10 分時間割	3
3.2	朝会	4
3.3	daily 記録	4
4	実験内容	5
4.1	時間割	6
5	LifeHacks 用ウェブアプリケーションの開発	6
5.1	開発の目的	6
5.2	ウェブアプリケーションによるメリット	7
5.3	各機能	7
5.4	開発環境	10
6	実験結果	11
6.1	利点	11
6.2	欠点	14
7	結論	16
7.1	ツールからシステムへ	16
7.2	LifeHacks の教育的利用	16
付録 A	アンケート結果	19
A.1	良かったこと	19
A.2	良くなかったこと	19
A.3	なければどうなっていたか	19
A.4	今後どうすればよりよくなるか	19
A.5	付録 2 ソースコード	20

概要

近年生産性を向上させるテクニック群である LifeHacks が人気を博しており、ビジネス誌などでしばしば取り上げられている。しかし学生向けに情報があまり発信されておらず、学生が LifeHacks を自然と知る機会は多くない。そこで、今回大学の研究室に LifeHacks を導入し、学生に対してどのような効果があるか検討した。

本研究では LifeHacks における“記録”、“スケジュール管理”、“ウェブアプリケーションによる LifeHacks の効率化”の3要素に重点をおいた。これらの要素に時間割や研究室などの大学特有の制限を加味し、大学向けに応用した LifeHacks を考案し実験を行った。

実験は情報安全工学研究室に所属する、自身を含む10名の学生の協力によって行った。3か月間実験を行い、実験終了後学生にアンケートを行った。アンケートによると、多数の利点が挙げられる一方で、ツールとしての限界を感じさせる点も挙げられていた。

LifeHacks を積極的に導入した学生にはかなりの効果があったが、消極的にしか導入していなかった学生にはあまり効果が見られなかった。また LifeHacks を導入したことで、実験終了後積極的か消極的かに関わらず、学生に生産性を向上させることへの意識改革が起こった。スケジュール管理やモチベーションの維持といったことに興味を持つようになったのだ。大学を卒業して社会に出る前に LifeHacks を経験することは、学生の自己啓発を促す。本実験により大学での LifeHacks 導入には、教育的価値があることを見出すことができた。

1 導入

仕事場や私生活における生産性^{*1}を向上させる方法が、“LifeHacks”や“仕事術”など(以後本論文ではこれらの方法を LifeHacks で統一する)と呼ばれ近年人気を博している。LifeHacks の範囲は多岐に渡り、以下にいくつかの例を挙げる。

- アプリケーションやシェルスクリプトによるコンピュータでの自動処理
- 情報化社会において必要な情報を効率よく得るための整理
- 手帳や Web アプリケーションを用いたスケジュール管理
- 起床時間や体重を記録していくことによる私生活の改善
- 睡眠や運動による身体機能や脳機能の改善

このように LifeHacks では、文字通り生活全般の改善方法について広く取り扱っている。実践される場所もまた自宅、仕事場、通勤路などさまざまである。

しかし大学における LifeHacks の例^{*2}はあまり見ない。私はその理由を情報の不足と実践の制限との 2 つだと考えている。LifeHacks という分野は主にビジネス誌や社会人向けウェブサイトで取り扱われている。そのため学生向けの情報があまり発信されておらず認知度が低い。また学校は仕事場や自宅に比べて授業時間や個人のスペースの狭さなどの制限が多い。そのため個人で改善できる部分が少ないか、狭い範囲に限られてしまうのではないだろうか。とは言え、LifeHacks は今や多くの場で実践されており、大学でも実践する価値はあるはずである。

そこで LifeHacks を大学内の私が所属する研究室に導入することで、学生の生産性にどのような影響があるのか実験を行った。実験後、参加者に対してアンケートを行い、得られた結果を基に大学での LifeHacks 実践の価値を検討した。

^{*1} 本研究での生産性とは単位時間当たりの仕事量のこととする。

^{*2} 大学に限らない LifeHacks はのぞく。

2 LifeHacks の応用

実験を行うにあたって、大学向けに応用した LifeHacks を導入する必要がある。最大の効果を得るには研究室の内情に即したものが最も良い。しかし研究室は研究室ごとの独自色が強い。研究室の内情に個別に対応していくと、“大学での実践例”から大きくはずれる可能性がある。研究室によりすぎず、あくまでも大学生活一般において有用なものを導入すべきである。

そこで、学生一人々々に対応させるのではなく、研究室に所属する学生全体を 1 つのチームととらえることにした。チームでの生産性を上げる LifeHacks[1] を応用することで、研究室に所属する学生全体の生産性を上げることができるだろう。

LifeHacks を応用するにあたって、以下の要素を特に重要視した。

(a) 記録

その日 1 日の自分の行動や、成果などを記録する。記録することで、自分がどの作業に時間を取られたか、無駄な時間を把握することができる。

記録という手法は世間で幅広く使われており、自己の管理や成長に効果的である。スポーツではタイムやスコアを記録し、それを参考に訓練することはよく知られている。またスポーツに限らず、ダイエットや禁煙においても効果的であることが認められている。

(b) チームでの 1 日のスケジュール管理

チーム全体で、参加者の 1 日のスケジュールを把握できる環境作りを行う。チーム内でお互いのスケジュールやその中の作業を把握することで、お互いの作業を可視化できる。可視化することで、お互いの作業に対する助言や提言を行いやすくなる。チーム内でお互いに助言・提言を行うことで、自分だけでは気づかなかった点や無駄に気づくやすくなり、作業の効率化を行うことができる。

(c) ウェブアプリケーションの作業を用いた LifeHacks の効率化

(a) と (b) をウェブアプリケーションを用いて効率化を図る。コンピュータにより一部を自動的に処理 [2] することにより、LifeHacks を行う上での負担を軽減することができる。

具体的にはチーム内の 1 日のスケジュールをまとめ、同時に記録するウェブアプリケーションを開発する。チーム全員分のスケジュールを把握することは、非常に手間と時間がかかってしまう。そこでコンピュータを用いて、できるだけ人の手がかかる部分を排除する。

3 実験概要

大学内の研究室を対象として LifeHacks を実践するにあたり、以下の 2 つの条件を考慮しなければならない。これらの条件は今回実験を行う大阪産業大学工学部情報システム工学科を基準としているが、他の一般的な 4 年生大学と大差はない。

- 大学の時間割
- 学生ごとの履修状況の違い

大学の授業は時間割で区切られ行われている。そのため学生が大学でのスケジュールを組むときは、基本的に時間割ごとに時間を区切って行う場合が多い。授業が連続で 1 日中あるわけではなく、授業の間に自由な時間が取れることもある。

また学生ごとの履修状況も異なり、人によって授業がある時間とない時間が存在する。このためある時間は授業、ある時間は自由、というように単純に仕切れるものではない。

以上を踏まえて、前項で述べた要素を満たすために、今回以下の手法を用いることにした。次項より詳しい説明を行う。

- 40:10 時間割
- 朝会
- daily 記録

3.1 40:10 分時間割

この手法は 48:12 分仕事術 [3, 4] という手法を参考に、大学の時間割向けにに適用したものである。導入によってやる気・モチベーションの向上・維持を目指し、またわかりやすいスケジュール管理の目安とする。

48:12 分仕事術という手法は、1 時間を 8 割にあたる 48 分と 2 割にあたる 12 分とに分け、48 分を集中して仕事をする時間 (仕事時間)、12 分を休憩時間にあてるというものである。48 分間はあらゆる雑音や邪魔を排除して、仕事をひたすら片付ける。12 分間は 48 分間に取った行動とは違う行動を主とした休憩を取ることで、やる気・モチベーションの維持にあてる。

重要なのは 12 分間の休憩であり、必ず取らなければいけないことである。また 48 分間とは違う行動ということも重要である。例えば 48 分間デスクワークを行ったならば、12 分間はデスクから離れて体を動かす必要がある。そうすることで脳と身体を活性化し、モチベーションの低下を防ぐ [5, 6]。

48:12 分仕事術は 1 時間という区切りで行われているが、大学では時間割を考慮する必要がある。私が所属し、今回実験対象となっている大阪産業大学の授業は 1 時限につき 90 分間であり、間に 10 分間の休憩がある。そこで、授業の 90 分と休憩の 10 分を足して 100 分とし、それを 2 つの 50 分と考える。その 50 分を 48:12(4:1) で分けることで、40 分間の仕事時間と 10 分間の休憩時間にあてることにした。これにより大学の時間割と 48:12 分仕事術の応用が可能となる。

1 コマを 40 分 (+10 分) とし、2 コマで 1 時限分とする。また便宜上 1a・1b のような名称にし、対象時限数に a と b を付加することで、その 1 時限分の前半と後半とする。

1 日のスケジュールを立てる際には、この 1 コマを単位とし、1 コマごとの作業予定をあてるようにする。チーム全体でスケジュールを把握するにあたって、共通の単位を用いたスケジュールを立てることで、よりわかりやすくなる。

3.2 朝会

大学の1限目が始まる時間は09:00 JSTであるが、その30分前の08:30 JSTからチーム全員での会議(以下朝会)を20分間で行う。

朝会では毎日ランダムで議長を選出し、議長は朝会の司会とまとめを行う。ランダムで議長を選出することにより、メンバー全員がほぼ等しく議長になる可能性を得る。毎日ランダムで選ばれるということから、メンバー全員に緊張感と責任感を持たせることができる。また時間が限られているため、議長がいかに朝会を進めるかという点も重要で、時間を意識したファシリテーション技術が養われる。

またメンバーは朝会が始まるまでに、スケジュールを立てておく必要がある。スケジュールは40:10分時間割で設けた新たな時間割を用いる。

朝会で行うことは主に以下の2つである。

- 個人の主張
- スケジュールの把握

個人の主張部分では、個人がチーム内のメンバー全員に向かって自由な発言を行う。内容は問題提起から昨日あったイベントの話などさまざまである。時間がかかりそうな場合や、会議が必要な場合は、別に時間を取って続きを行う。時間を取る場合は緊急をのぞいて、別の日のがのぞましく、事前連絡をした方がよい[7]。当日の場合、は他人の既に立てられているスケジュールに影響を与えてしまうからだ。

スケジュールの把握部分では、議長が決めた順番に従ってその日1日のスケジュールを発表し、発表者以外のメンバーは助言・提言・質問を行う。こうすることで、自分の立てたスケジュールを客観的に見てもらうことができる。客観的に見てもらうことで、スケジュールの無駄の指摘や作業の助言・提言をしてもらえる可能性がある。にさまざまな知識や技術をもったメンバーが混在するチームで、公に他人の意見を聞ける場を設けることで、他人の知識・技術を有効利用することができる。

朝会が終わり次第、議長はチームで共有しているメーリングリストに各自のスケジュールと個人の主張のまとめをメールで送信する。これによってチーム内で他のメンバーのスケジュールと主張を共有することができ、またメールという形を用いているため整理や参照が容易である。

3.3 daily 記録

40:10分時間割で新たに設けた時間割を用いて、帰宅前に毎日の行った作業を記録する。自分がスケジュール通り作業を行うことができたかどうか確認できる。また毎日の記録を集計していくことで、実際に自分がどの程度のペースで作業を進められているかの把握が容易となる。

このdaily記録は日報という形でチーム内のメンバー全員に共有される。この時40:10分時間割の作業記録とは別に、その日の成果や反省、今後の目標を記録し共有する。メーリングリストに投稿するメールという形で行われるので、各メンバーが別々の場所においても携帯電話やコンピュータで把握することができる。

記録を取る際にはできるだけ具体的な方がよい。例えば1コマで読んだ本の文字数や、走ることの出来た距離などである[8, 9]。具体的であればあるほど今後の参考にしやすく、自己の成長に役立てることができる。またコマ単位で管理することで、時間帯や体調によってどれだけ影響を受けるかがわかりやすくなり、自分のことを知りやすくなる。

4 実験内容

実験は私の所属する情報安全工学研究室内で以下の条件で行なった。期間は 2009 年 8 月 1 日から 10 月 31 日までの 3 か月間である。

- 参加した学生は自身を含め 10 名
- 参加学生は全て 4 回生で情報安全工学研究室所属
- 履修状況に違いがある (授業のある学生とない学生がいる)
- 朝会や daily 記録の参加はあくまでも自主性に委ねる
- 40:10 分時間割のみ強制力を持つ

1 日の主なスケジュールを以下の図 1 に示す。

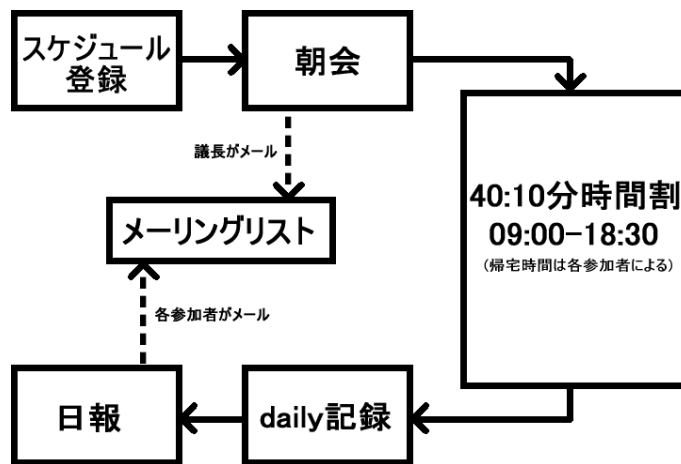


図 1 1 日のスケジュール

参加者は 08:30JST に始まる朝会までに 1 日のスケジュールを立てておく。その後朝会を行ない、議長の選出、個人の主張、スケジュールの把握を行なった後 08:50JST までには終える。09:00 JST 以降は研究室に合わせた 40:10 分時間割 (詳しい時間割は次項参照) にのっとりスケジュール通り行動し、帰宅前には daily 記録を行なった上で日報としてメーリングリストにメールを送信する。これを平日の月曜日から金曜日まで毎日繰り返す。

ここの改ページ不要

4.1 時間割

40:10 分時間割を研究室に合わせ、表 1 の時間割にする。時間は全て JST である。

適度な休憩を取るという観点と、リズムを作るという観点から、3 コマの仕事時間を 1 つのセクションとし、各セクションの間には 1 時間以上の休憩時間を用意した。

表 1 研究室に合わせた 40:10 分時間割

朝会	08:30 – 08:50
1a 限	09:00 – 09:40
1b 限	09:50 – 10:30
2a 限	10:40 – 11:20
昼休憩	11:20 – 12:50
3a 限	12:50 – 13:30
3b 限	13:40 – 14:20
4a 限	14:30 – 15:10
自由時間	15:10 – 16:10
5a 限	16:10 – 16:50
5b 限	17:00 – 17:40
6a 限	17:50 – 18:30
自由時間	20:00 – 21:50

ここは改ページ必要

5 LifeHacks 用ウェブアプリケーションの開発

今回の LifeHacks の一環として、各 LifeHacks を効率化するためのウェブアプリケーションを開発した。実験と同時進行で開発とアップデートを行なったため、こちらは 3 ヶ月間全体での運用にはならなかった。しかし基礎となるシステム自体は実験 2 ヶ月目には完成しており、その後細かい点でアップデートを行なったものの、概ね 2 ヶ月間運用することができた。

また今回は実験ということもあり、研究室内のコンピュータからのアクセスしか受け付けないようにした。一部ユーザー認証が必要であるが、基本的に研究室からでは誰でも使用することができる。

5.1 開発の目的

主な目的はスケジュール把握と daily 記録部分の一部自動化による、作業の効率化である。

当初朝会后にチーム全体のスケジュールを共有する際に、ML にメールで送信するという方法を取っていた。しかし問題があり、議長が一括してスケジュールをまとめたメールを作成し、送信するのでは議長 1 人にかなりの負担を与えてしまう。ただでさえ毎回ランダムで選ぶようにしているので、議長になった場合 1a 限のスケジュールに影響を与えかねなかった。また参加者全員が 1 人 1 人自分のスケジュールをのせたメールを作成し、送信するのではメールを開く手間がかかってしまう。

そこでその手間がかかってしまう部分をアプリケーションで一部自動化することにした。具体的には、参加者各人がアプリケーション側に 1 日のスケジュールを登録すると、アプリケーション側が自動で全員分をまとめる。

議長はそのまとめ部分を引用し、朝会で変更のあった部分を修正して ML にメールで送信するだけにした。

同時にこのスケジュール登録機能は daily 記録に応用することができる。帰宅前にスケジュール部分を実際の行動に置き換えて登録することで、自動的にコマごとに作業項目をまとめ集計する。集計することで自分が今までにどの作業を多く行なったかや、無駄を確認でき、今後のスケジュールに活かすことができる。

また生産性を上げるという観点から、ユーザーにとって使いやすく、対話的なシステムを目指した [10]。

5.2 ウェブアプリケーションによるメリット

デスクトップアプリケーションではなくウェブアプリケーションを開発した利点は以下の 2 点である。

- インターネット環境とウェブブラウザを持つ端末があればどこからでも使用可能
- バグ修正やアップデートが容易

チーム全体のスケジュールを改めて把握しようとする際に、特定のコンピュータのデスクトップアプリケーションを起動しなくてはいけないとなると大変不便である。しかし、ウェブアプリケーションの場合はインストールが不要で、インターネット環境とウェブブラウザを持つ端末からならばどこからでも使用可能で便利である。

また参加者からの要望があれば、実験と同時進行でアプリケーション自体を随時アップデートする必要がある。ウェブアプリケーションならば 1 度サーバー上のアプリケーションをアップデートすれば済む。しかしデスクトップアプリケーションの場合、アップデートが必要な度にユーザー側でアップデートする必要があり、アプリケーションをインストールしているコンピュータの数だけ手間がかかってしまう。ウェブアプリケーションとデスクトップアプリケーションを比べてみると、前者の方がアップデートが容易であると言える。

5.3 各機能

主に 4 つの機能を実装した。次項より各機能の解説を行なう。スケジュール登録の部分のみユーザー認証が必要である。

1. ユーザー認証
2. スケジュール登録
3. スケジュールまとめ
4. 記録の自動集計によるデータ集

全体の概念図を図 2 に示す。

5.3.1 ユーザー認証

スケジュールまとめやデータ集は誰でも見る事が可能であるが、スケジュール登録のみはユーザー認証が必要である。故意や事故による誤登録を防ぐためである。

ユーザー認証にはユーザー名とパスワードが必要である。ユーザー名は単純に参加者の苗字のアルファベットとしている。パスワードは各人自由にいつでも設定できるようにしており、認証時には MD5 というハッシュ関数を用いている。MD5 によって得られたハッシュ値と、データベースに登録されているユーザー情報のハッシュ値とで照合し、ユーザー認証を行なう。研究室内向けの実験用ウェブアプリケーションであるので、ユーザー登録などのユーザー管理部分は今回作成しなかった。人数に変動がなく、10 名という数なので何か変更があれば、私だけで十分対応が可能だからだ。図 3 にユーザー認証画面の例を示す。

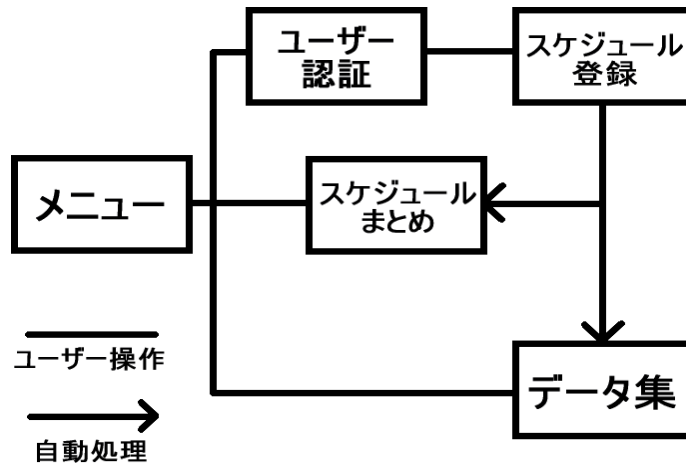


図 2 全体の概念図



図 3 ユーザー認証の例

5.3.2 スケジュール登録

40:10 分時間割に従ってスケジュールを登録する。基本的にその日の入力しか受け付けないようになっており、ここで登録されたスケジュールは、チーム全員で共有されるスケジュールまとめに登録される。1 度登録されれば終わりというわけではなく、いつでも変更や削除が可能である。

スケジュール登録と同時に、簡単な主張部分としてつぶやき機能を搭載しており、簡単なメッセージをチャットのように送信・共有することができる。図 4 にスケジュール登録画面の例を示す。

5.3.3 スケジュールまとめ

スケジュール登録によって得られた各人のスケジュールをまとめて表示する。全てテキストベースであり、表やグラフなどは一切表示しない。これはこのスケジュールまとめをメールにコピーして送信するからである。朝



図 4 スケジュール登録とつぶやきの例

会に参加しなかった者も表示するようになっており、朝会に出席したか欠席したかわかりやすいようになっている。図 5 にスケジュールまとめ画面の例を示す。



図 5 スケジュールまとめの例

5.3.4 記録の自動集計によるデータ集

コマごとの作業内容をまとめ、割合を表示する。記録開始日から最後に記録した日までの期間*3も表示している。項目数のしきい値を上昇させることで、項目数の少ない項目をその他にまとめることもできる。図 6 にデータ集画面の例を示す。

各人のデータ集

#しきい値:2
indexへ

uemori
各項目のまとめ(2009-09-28~2009-12-25)

項目名	コマ数	比率
卒研	105	28.4%
卒論	93	25.1%
江双	41	11.1%
読書	40	10.8%
論文	37	10.0%
プレゼン修正	8	2.2%
卒研など	8	2.2%
週報	6	1.6%
論文修正	5	1.4%
デバッグ的なもの	4	1.1%
スライド制作	3	0.8%
おかえり	3	0.8%
プレゼン思案	3	0.8%
プレゼン	3	0.8%
おでかけ	3	0.8%
発表練習	3	0.8%
その他	5	1.4%

takenaka
各項目のまとめ(2009-09-24~2009-12-18)

項目名	コマ数	比率
卒論	49	35.0%
卒研	26	18.6%
日報	15	10.7%
バイト	12	9.2%

図 6 データ集の例

5.4 開発環境

以下の環境で開発を行なった。

- ハードウェア (IBM-PC/AT 互換機)
 - HDD.....80GB
 - CPU.....Intel® Core™2 1.66GHz[x86 Family 6 Model 14 Stepping 8]
 - ロジックボード.....YNRC-BR
 - チップセット.....Radeon Xpress 200M
 - メモリ.....1GB
- OS.....Vine Linux 4.0
- 開発言語.....PHP 5.2.6
- データベース.....MySQL 5.0.27

*3 実験終了後も使い続けているため、最後に記録した日は人によってまばらである

6 実験結果

今回結果をまとめるにあたって、参加者にアンケートを行った。アンケートは以下の条件で行った。アンケート結果自体は項目ごとに簡単にまとめ付録としている。

- 対象は参加者 10 名の内自身を除いた 9 名。
- 個別面談方式で、それぞれ約 20 分程。
- 主なアンケート項目は以下の通り。
 - － 良かったこと
 - － 良くなかったこと
 - － LifeHacks を行っていないかどうなっていたと思うか
 - － 今後の課題
- その他個別質問も交えた。

このアンケートで得た回答をもとに、利点と欠点を以下のように分けた。次項より詳しく説明する。

1 利点

- 1.1 計画・行動・結果によるフィードバック
- 1.2 スケジュールと優先順位の明確化
- 1.3 作業の可視化による助け合い
- 1.4 チームでのモチベーション維持
- 1.5 休憩時間の有効利用
- 1.6 ウェブアプリケーションによる効率化

2 欠点

- 2.1 ツールとしての限界

6.1 利点

6.1.1 計画・行動・結果によるフィードバック

朝会のためのスケジュールリングと日報のための daily 記録により、1 日の自分の計画と行動の結果がよくわかるようになった。そのため自分がどれだけ計画を達成できたかや、達成できなかったかが一目瞭然となった。

実験前までスケジュールを組んで行動していなかった参加者の変化が特に顕著だった。スケジュールを組まず、記録も取っていない場合、1 日の自分の行動があいまいにしか把握できなくなる。つまり自分がどれだけ作業に時間をかけているかや、無駄な時間を過ごしているかを正確に把握できていない。具体的に問題点をとらえられなければ、具体的な解決策に取り掛かれるはずもなく、更にいくら問題に取り組んでも結果がわかりにくい。そこで結局、的確な解決策に乗り出せていない参加者が多かった。

しかし今回の実験で、あいまいになっている部分を参加者自ら具体的に認識することにより、改善のための具体的な努力をすることができるようになった。

参加者の中には、参加者自身が立てたスケジュール通りに動けなかった場合「悔しい」気持ちになったという者もいた。悔しい気持ちになった次の日は、前日よりもより努力しようという行動に繋がり、結果としてより自制心がついたとのことだった。

また成果が記録されることで、モチベーションの維持にもつながった。これは当初より想定されていた利点で

あるが、実際にしてみたことで、より効果を実感することができた。ある参加者は単位時間あたりの読書量*4を記録することで、自分の読書スピードの改善に利用していた。実際に効果があったようで、実験前に比べて実験後は格段に読書量が増えていたようだ。毎日の小さな改善が、3か月という期間を経て、大きな成果に繋がったと言える。

6.1.2 スケジュールと優先順位の明確化

実験1ヵ月目はまだ不慣れな参加者が多く、スケジュール通り行動することができないことが多かった。しかし2ヵ月以降はスケジュールをもとに行動することに慣れたようで、スケジュールを意識した行動を取るようになっていた。

スケジュールを40:10分時間割に合わせて40分単位で立てることにより、時間を意識した動きがしやすくなっていた。多くの参加者は、実験前までは「午前中」や「大体何時まで」というような曖昧な時間を取り方をしていた。ところが40分という小刻みに明確化された時間で取り組まなければならなくなったため、40分で作業をこなすという意識が参加者に芽生えた。時間内の作業をこなすことを意識することで、他の誘惑に負けやすくなったとのことだった。

小刻みなスケジュールを意識しだすと、優先順位もまた意識しやすくなる。自分の中でしっかりと「緊急か緊急でないか」「重要か重要でないか」「先が見えているか見ていないか」などを意識するようになり、自然と洗練されたスケジュールを立てるようになっていった。更に朝会で他の参加者から意見をもらうことで、客観的に判断してもらえることも、優先順位の明確化において役にたった。

例えばある参加者は、明確化され朝会で宣言し確定したスケジュールに従って行動している間は、いくら他人からの頼みごとであってもすぐには行動せず、しっかりと自分のスケジュールと優先順位を考えてから行動するようになった。どのような場合においても、必ず優先順位を意識することで、目標達成に向けて着実に作業を進められるようになっていた。

スケジュールを明確化することで、スケジュールを意識しやすくなる。同時に他の要因に惑わされないように、優先順位を守ろうとする気持ちが働き、結果としてより洗練されたスケジューリングができるようになるのだ。

6.1.3 作業の可視化による助け合い

朝会と日報によって他人の詳しい作業内容がわかるようになり、自然とチーム内でお互いに助け合う姿勢が見られるようになった。

実験前までは、お互いの作業内容や進捗状況をほとんど把握できていなかった。特に今回実験を行った情報安全工学研究室は、今年度の卒業研究内容が1人1人違う。基本的に1人で始めから最後まですることになり、内容自体も専門化していくので、お互い手が出しにくかった。結果として、お互いの卒業研究の進捗状況がわからず、そもそも積極的にわかるという姿勢がなかった。

しかし朝会と日報によって他人の行動を詳しく把握できるようになり、自然とお互いの進捗状況がわかるようになった。すると参加者同士でお互いのアドバイスをし合うようになり、問題解決のためにさく時間が劇的に少なくなった。また複数の参加者の知恵を出し合うことで、より良いアイデアや結果を得ることができるようになった。

特に卒業研究を行うにあたっては、専門知識やアイデアが要求される。しかし大学の4年生ともなると各自得意分野が違うので、うまくチーム全体の知恵や知識として活用することで、全く違ったアプローチや効率的な方法などを得ることができる。各自の卒業研究内容と得意分野が違うからこそ、チームで助け合える環境というのは非常に有効であった。

*4 文字数 ÷ 読書時間 によって導き出すことができる。

6.1.4 チームでのモチベーション維持

チームを意識することで、モチベーションの維持に繋がる。これは LifeHacks に限ったことではなく、資格勉強やジムでのトレーニングなど一般的なことにも通じることである。複数人で同じことにあたることで、「しなければいけない」という義務感が発生し、安易に引き下がらないようになるのだ。本実験の場合は主に朝会への参加やスケジュールにおいて効果が現れた。

朝会への参加は強制ではないが、基本的には参加することが、参加者の中で暗黙の了解となっていた。普段朝会に参加している参加者が連絡なしに休んだ場合などは、後になってどうして来なかったのかと参加者同士で尋ねられるようになっていた。また日頃から朝会への参加率が低い参加者は、朝会に参加しないことを非難されることがあった。方法の是非はともかく、参加者は朝会に参加しなければならないという意識を持つようになり、次第に積極的に参加するようになっていたのだ。実験開始から1ヵ月が過ぎたころには、朝会への参加はごく自然なものとなり、私が欠席している日でも通常通り朝会は行われていた。

またスケジュールにおいては、朝会后チーム全体に ML を通して送信されることが参加者に対して義務感を生じさせていた。朝会時に確定したスケジュールをチーム全体に対して宣言することで、スケジュールを達成しなければならないという気持ちが強く働いたようである。またチーム全体で各自のスケジュールを共有しているので、とある参加者がスケジュール通り行動できていない時は、他の参加者から指摘されるようになっていた。

どちらも当初は指摘されるのが嫌などという気持ちからの義務感であったが、日を重ねるに連れて次第にごく当たり前のようになっていた。参加者同士で良くないところは指摘し合うことで、お互いの改善にも繋がり、また団結力が生まれた。チームとしてよくまとまることで、一丸となって各自の目標達成に向けて行動するようになっていた。チームをより意識することで、結果的にチーム全体のモチベーション維持に繋がっていた。

6.1.5 休憩時間の有効利用

40:10 分時間割を導入するまでは、各自が好きな時間に休憩を取っていた。休憩時間は主に読書、睡眠、会話、ゲームなどに費やされていた。チームが個室の中で好きな時間に休憩を取ると、下図のような休憩時間のずれが生じる。またそのチーム内の親密度が高いほど、相手の行動や言動が気になるということから、休憩時間のずれはチーム全体の生産性低下を招いてしまう。(図7)

この問題は実験前から研究室内で発生しつつあったが、40:10 分時間割によって休憩時間が全員同じなることで、作業時間中に誰かが休憩を取ることがなくなった。結果として40分間の仕事時間により集中することが可能となり、生産性の向上に繋がった。(図8)

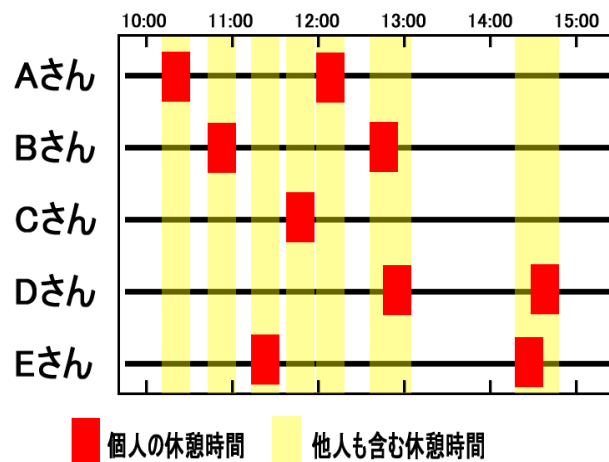


図7 実験前の休憩図

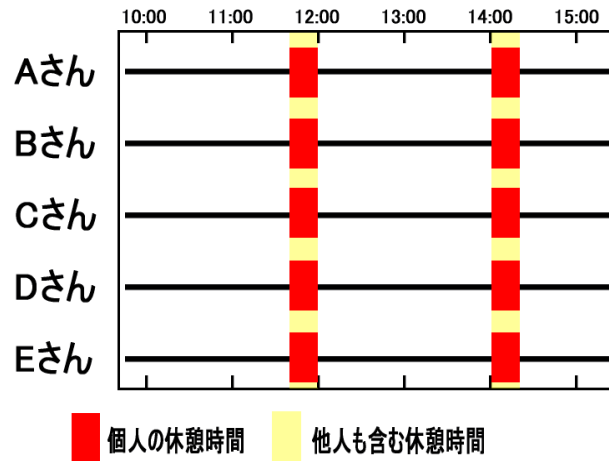


図8 実験後の休憩図

6.1.6 ウェブアプリケーションによる効率化

朝会と daily 記録の負担軽減のために専用のウェブアプリケーションを開発・実装したが、当初想定していた通りの効率化に成功した。時間の節約や参加者の負担軽減に繋がったことは、実際に参加者から高い評価を得ることができた。

一時期朝会と daily 記録をウェブアプリケーションなしに行っていた。しかし 10 名の参加者ともなると、各自の予定をまとめ ML にメールを送信するだけでも、朝会の議長にとっては大きな負担となっていた。場合によっては 1a 限 (40 分) の予定に支障をきたすこともあり、議長がランダムに決まるということと相まって、参加者全員のスケジュールを乱す可能性を内包していた。

それがウェブアプリケーションを実装してからは、全員の負担が大きく軽減された。議長が朝会后に ML にメールを送信する作業が、実装前で約 10 分程かかっていたのだが、実装後は 1、2 分もあれば完了していた。実装前までは完全に議長が予定を入力していたのに対して、実装後はまとめページをコピーして貼り付けるだけになっていたからだ。議長自身が入力する必要があるのは、個人の主張の部分だけである。

また daily 記録を入力するだけで、自動でまとめてくれるというのは、記録をまとめるという作業部分をなくせることになり、こちらも時間節約と負担軽減に繋がった。ある参加者は、わざわざまとめる時間を取る必要がなく、それでいて一見して自分の記録がまとめられていてわかりやすいというのは、自己改善をする上で役に立っていたと言っていた。

6.2 欠点

6.2.1 ツールとしての限界

今回の実験はあくまでも協力という形を取ったため、ほとんど強制力を持たなかった。企業で行う場合と比べると、この点が最も大きな違いだと言える。40:10 分時間割のみは全員の協力が必要なため、全員に協力してもらったが、他は基本的に自主性に委ねることになった。

自主性に委ねることによって、積極的に利用した学生 (積極的學生) と消極的利用にとどまった学生 (消極的學生) とに分かれることになった。結果から見ると、卒業研究の進捗度と完成度から見たこの 2 者の間の差は大きかった。積極的學生が実験終了時 (2009/10/31) の段階でほとんど卒業研究が終わっていたのに対し、消極的學生は卒業研究に難航しているかほとんど手を付けられていなかった。

この消極的學生にはいくつかの以下のような共通点があった。

- 朝会参加率が低い
- スケジュールや daily 記録があいまい
- 休憩時間をうまく利用できていない

朝会参加率が低い

積極的學生と消極的學生とを比べて、最も顕著であったのが朝会参加率である。実験期間 3 ヶ月の内、朝会参加数が 10 回に満たない参加者は、2009 年 12 月に入ってもいまだ卒業研究を完成することができていない。逆に 7 割以上参加している参加者は全員卒業研究を完成させることができています。朝会へ参加できていない主な理由は意識の低さである。朝会は 08:30JST からだが、今回の実験参加者の中で物理的にこの時間に研究室に來れない参加者はいない。また長期的に参加できない特別な理由があるわけでもなかった。

スケジュールや daily 記録があいまい

今回の実験ではスケジュールや daily 記録の明確化具体化に重きをおいているが、消極的學生はいずれもあいまいな表現にすることが多かった。例えば「午後は卒業研究します」のようなもので、時間的な制約がゆるく、具体性に欠けている。結局これでは他の参加者から見ると、何をしているのかわからず、助言のしようもない状態となる。

あいまいさの部分は、朝会に参加することである程度他の参加者から指摘されるので、改善される場合がある。しかし前述の通り消極的學生は朝会にほとんど参加せず、多くの場合指摘を受けることがない。

休憩時間をうまく利用できていない

40:10 分時間割にそって動いているにも関わらず、休憩時間をうまく利用することができていなかった。作業をだらだらと続けている場合や、休憩時間外に休憩をとっている場合^{*5}があった。

これらの共通点は本実験がツールとしての枠を出ず、万人に効果があるわけではないことを物語っている。つまりシステムではなくツールでしかないということだ。積極的學生はツールとしてうまく利用し、前述した利点を受けることができた。一方、消極的學生はツールをうまく利用していなかった。消極的學生でも強制的に利用させることができるのであれば、利点のある程度は受けることができたはずである。しかし大学においての実践という限りにおいては、授業という形態をとって教員が直接指導するなどの場合をのぞいて、強制力を持たせることは難しいだろう。

*5 他の参加者に迷惑をかける行動は自粛していた。

7 結論

実験環境 (大学内研究室) における LifeHacks は多くの利点があることがわかった。しかし同時に、前述した消極的學生は LifeHacks をツールとしてうまく使えず、実験前と比べても生産性の向上はあまり見られないこともわかった。参加者に等しく効果を望むならば、ツールではなくシステム化する必要があるだろう。

また実験終了後、実験前と比べて、参加者自身の生産性に対する意識の向上が見られた。大学生が社会に出る上で、LifeHacks は良い経験になったようで、LifeHacks に教育的価値がある可能性を見出すことができた。

7.1 ツールからシステムへ

消極的學生に対しては前項で、強制力を持たせればうまくいく可能性を示唆したが、それ以前にもまだ改善できるはずであると考えている。例えば事前に LifeHacks 実践における説明や、もっと自然と LifeHacks に取り組めるようなシステム作りを行えば良かったのではないかと。

今回の実験は実験期間 (3 か月) 確保のために足早に開始してしまった経緯があり、事前の説明が不足していたと私自身感じている。実際に消極的學生は LifeHacks の効果や意義を十分に理解しておらず、あまり前向きに取り組んでいたとは言えない事前に十分な時間をとり、参加者に LifeHacks の方法や意義、効果について (例えばプレゼンテーションのような方法を用いて) 詳しく説明できれば、参加者のモチベーション向上や維持に繋がったのではないだろうか。

また、もっと簡単に LifeHacks に取り組めるシステム作りも有効だろう。LifeHacks はうまくシステム化しなければ負担増になることがある。今回の実験でも前述した通り、ウェブアプリケーションを導入していなければ朝会や daily 記録は参加者の負担になっていた。負担をいかに軽減し、簡単に LifeHacks を実行できるシステムを作らなければ、LifeHacks は長続きしないだろう。

今回の実験では LifeHacks はかなりツール寄りになってしまい、効果は参加者の意識や積極性にゆだねられてしまっていた。今後は参加者の意識に左右されにくい、誰でも簡単に使えて効果のあるシステム作りを念頭におく必要があるだろう。

7.2 LifeHacks の教育的利用

積極的學生の多くが、実験開始後から生産性について興味を持ち始めた。これは実際に積極的學生が LifeHacks による生産性向上の効果を実感したからだろう。アンケートにも、自己改善のための LifeHacks に興味を持ったとの意見が見られた。

大学生は大学院進学などをのぞいて、そのほとんどが就職などで社会に出ることになる。その社会では初めに述べたように、LifeHacks の有用性が認められており、実際に職場で使われているケースが多数ある。それを踏まえると、大学生が社会に出る前に LifeHacks を学び、実践することは教育 (Education^{*6}) という意義に沿っており、汎用的な職業訓練に相当する可能性を秘めている。

そういった意味では今回の実験環境に限らず、學生が LifeHacks の意義や方法などを学び、チームや個人などを問わず実践の機会を与える価値はあると言える。教育的利用で言うと大学に限らず、高校や小中学校でも LifeHacks を勉強に応用させ実践させると、生徒の自己改善に繋がるのではないだろうか。

LifeHacks の教育的利用は、成績や受験のためだけではない、職業訓練に近い汎用性の高い教育になるだろう。また自己改善に伴う人間性の向上は未成年者や學生にとっても、有意義なものとなるはずである。本実験により、

^{*6} 社会に出て自分でご飯を食べる力をつけさせること

LifeHacks の教育的利用の価値を実際に見出すことができた。

謝辞

情報安全工学研究室での実験を進める上で、担当教員である大垣斉講師には大変な協力・ご指導をいただきました。また実験参加者である情報安全工学研究室所属の9名の学生、実験のきっかけを作っていただいた先輩方も含め、感謝の意を表します。

参考文献

- [1] 小室淑恵. 6時に帰る チーム術. 2008, 日本能率協会マネジメントセンター.
- [2] Neal Ford. プロダクティブ・プログラマー—プログラマのための生産性向上術. 2009, オーム社. 島田浩二訳.
- [3] 堀 E. 正岳. 情報ダイエット仕事術. 2008, 大和書房.
- [4] 堀 E. 正岳 and others. Life Hacks PRESS, vol. 2 , pp. 66–68. 2008, 技術評論社.
- [5] Andy Hunt. リファクタリングウェットウェア. 2009, オーム社. 武舎広幸, 武舎ゆみ訳.
- [6] Tom Stafford and Matt Webb. MIND HACKS—実験で知る脳と心のシステム. 2005, オーム社. 夏目大訳.
- [7] Mark Forster. マニャーナの法則 明日できることを今日やるな. 2007, ディスカバー・トゥエンティワン. 青木高夫訳.
- [8] 堀正岳 and 佐々木正悟. iPhone 情報整理術—あなたを情報“強者”に変える 57の活用法!. 2009, 技術評論社.
- [9] 橋本大也. 情報力. 2009, 翔泳社.
- [10] Chris Crawford. クロフォードのインタラクティブデザイン論. 2004, オーム社. 安村通晃訳.

付録 A アンケート結果

アンケート結果を項目ごとにわけたものである。意味が重複しているものや、似ているものはまとめてある。

A.1 良かったこと

- 朝会参加への義務感によって、意識にポジティブな変化が起こった。
- 40:10分時間割によって、短期的な行動力や集中力が得られた。
- 自分の行動を他の参加者に宣言することで、より達成しようという気になった。
- 他の参加者の行動が見えるようになったので、わずらわされづらくなった。
- 記録を取っていくことで、スケジューリングなどの反省をしやすくなった。
- チームを意識することで、モチベーションの維持に役立った。
- メーリングリストをうまく使うことで、助言を得られた。
- 時間にケジメをつけた行動ができるようになった。
- コマ割りによって予定をたてやすくなった。
- 時間を意識しだすので、他の誘惑に負けづらくなった。
- 作業の優先順位を考えるようになった。
- 目標を明確にすることができた。
- 以前と比べて計画性を持てるようになった。

A.2 良くなかったこと

- 朝会にあまり参加しない参加者がおり気になった。
- 実験終盤でもスケジュールを守れていない参加者がいた。
- 行動記録があいまいな参加者がいた。
- 休憩時間中もずっと作業をしてしまっていた。
- 仕事時間中に時折他人の邪魔をする参加者がいた。
- 仕事時間中に1人が2人が崩れると、全員が崩れがちだった。

A.3 なければどうなっていたか

- 誘惑に負けていたらと卒業研究をするはめになっていた。
- 留年してしまっていたかもしれない。
- 生産性やスケジュールを考えることがなかった。
- できる人とできない人の差がひどかったと思う。

A.4 今後どうすればよりよくなるか

- 朝会に全員参加するように参加者を指導する。
- スケジュールをより具体化させる。
- スケジューリングや朝会の事前指導をする。
- 全員にさせたいなら、うまくシステム化する必要がある。

A.5 付録2 ソースコード

A.5.1 メニュー (index.html)

```
<html>
<head><title>team.andrew Hacks!!</title>
<link href='./asainD.css' rel='stylesheet' type='text/css'>
</head>
<body>
<h1>team.andrew Hacks!!</h1>
<p>ここは team.andrew Hacks!!のページでございます。卒研も含まれるでございます。</p>
<br>
<h2> ニュー</h2>
<form><input type="button" onClick="location.href='./login.php'"
  value="記録とつぶやき" style="width:230pt"><br><br>
<input type="button" onClick="location.href='./asainDm.php'"
  value="本日のまとめ" style="width:230pt"><br><br>
<input type="button" onClick="location.href='./asaintubu.php'"
  value="つぶやきログ" style="width:230pt"><br><br>
<input type="button" onclick="location.href='./usersum.php'"
  value="各人のデータ集" style="width:230pt"><br><br><br>

<h2>その他</h2>
<input type="button" onClick="location.href='./gasu2.gif'"
  value="プログラマをぶっ飛ばす" style="width:230pt"><br><br><br>

</form>
<h2>更新情報</h2>
<h4>2009/10/01</h4>
  - 同上ボタンをログインした段階でチェック状態にデフォルトで出来るようにしました。

<h4>2009/09/28</h4>
  - データ集部分のデータが上書きされてしまうバグを修正。<br>
  - データ集部分で、データが入力されている期間を表示するようにしました。<br>

<h4>2009/09/26</h4>
  - ユーザー名とパスワードによるユーザー認証ができるようになりました。<br>
  - パスワード変更ができるようになりました。<br>
  - ユーザー認証ができたので、構成が少し変わりました。<br>
  - 入力ページの入力欄に、既に入力されているデータが自動で反映されます。<br>

<h4>2009/09/25</h4>
  - 簡易データ集完成。今後も更新予定。<br>
```

- コイキングがギャラドスに進化しました。

- 更新情報を index.html に移しました。

<h4>2009/09/18</h4>

- つぶやき機能表示部分のミスを修正しました。

- 今までのつぶやきを表示するページができました。

<h4>2009/09/17</h4>

- つぶやき機能がつかしました。今後強化していきます。

<h4>2009/09/04</h4>

- オートコンプリートを無効化しました。

<h4>2009/09/03</h4>

- 同上自動化機能を修正しました。

- 意味のないボタンができました。コイキングのように役に立ちません。

<h4>2009/09/02</h4>

- 同じ項目を入力する手間を省くため、同上を自動化するチェックボックスをつけました。

- 失敗した場合などのために、項目を破壊するチェックボックスをつけました。

- まとめページでの表示がおかしくなるバグを修正しました。

<h4>2009/08/31</h4>

- 更新したい行のみ入力して送信すると、その部分のみ更新するようになりました。

- 送信後の出力が送信データではなく最新テーブルを参照するようになりました。

<h4>2009/08/28</h4>

- システムが一応の完成。

</body>

</html>

A.5.2 ログイン (login.php)

```

<html>
<head>
<title>hacks!ログイン</title>
<link href='./asainD.css' rel='stylesheet' type='text/css'>
</head>
<body>
<h1>ログイン</h1>
<form action="asainD.php" method="post">

<?php
require_once("info/info.php");
$ms=mysql_connect($SERVER,$DBUSER) or die("接続失敗");
mysql_select_db($DATABASE);
$gyo = 10;
$i = 0;
?>

```



```
ユーザー名<input id="intext" type="text" name="user" size="15"
  autocomplete='off'><br><br>
```

```
パスワード<input id="intext" type="password" name="pass"
  size="15"><br><br>
```

```
<input type="submit" value="ログイン" size="40"><br><br>
<br>
```

デフォルトでパスワードはぬるぼ状態になってます。

```
<br>
```

つまりパスワードに何もいれなくてログインできてしまいます。

```
<br>
```

私は変更したほうが良いと思います。

変更は入力ページの最下部にあります。

```
</form>
```

```
</body>
```

```
</html>
```

A.5.3 記録の入力 (asainD.php)

```
<?php
```

```
  session_start();
```

```
?>
```

```
<html>
```

```
<head>
```

```
<link href='./asainD.css' rel='stylesheet' type='text/css'>
```

```
<?php
```

```
  require_once("info/info.php");
```

```
  $ms=mysql_connect($SERVER,$DBUSER) or die("接続失敗");
```

```
  mysql_select_db($DATABASE);
```

```
  $uname = $_POST["user"];
```

```
  $pass = $_POST["pass"];
```

```
  $user = "asa".$_POST["user"];
```

```
  $passhash = hash('md5',$pass);
```

```
  $nin = mysql_query("select pass,dojo,chmod from asauser where name='$uname'");
```

```
  $ninsyo = mysql_fetch_array($nin);
```

```
  $dojoflg = $ninsyo[1];
```

```
  $chmod = $ninsyo[2];
```

```

$gyo = mysql_fetch_array(mysql_query("select id from asauser order by
asc limit 1"));

if($passhash==$ninsyo[0] && $uname!=''){
    session_unset();
    session_register("s_user");
    session_register("s_pass");
    $HTTP_SESSION_VARS["s_user"] = $uname;
    $HTTP_SESSION_VARS["s_pass"] = $passhash;
?>
<title>
<?php print $uname; ?>
's 記録</title>
</head>
<body>
<form action="asainDr.php" method="post">
<?php

$jigen = 11;
$i = 0;
$dayt = date("Y-m-d",time());

$time=array("1a","1b","2a","昼休み","3a","3b","4a","おやつ","5a","5b","6a");

print("<input type='hidden' name='jigen' value='$jigen'><h1>");
print date("Y/m/d",time());
print("</h1>データとしてまとめるので、項目名は統一するようにしませう。<br>");

$date = date("Y/m/d",time());

print("<input type='hidden' name='user' value='$user'>");
print("<input type='hidden' name='uname' value='$uname'>");
print("<input type='hidden' name='pass' value='$pass'>");

for($i=0;$i<$jigen;$i++){
    $default = mysql_query("select label,text from $user where day='$dayt'
and time='$time[$i]'");
    $def = mysql_fetch_array($default);

    print("<input type='hidden' name='id$i' value='$i'>");
    print("<input id='timetext' type='hidden' name='a$i' size='5' value='$time[$i]'
readonly>");
    print("<label>$time[$i]</label>");

```

```

    print("<input id='intext' type='text' name='b$i' size='10' value='$def[0]',
autocomplete='off'>    ");
    print("<input id='intext' type='text' name='c$i' size='45' value='$def[1]',
autocomplete='off'>");
    print("<input type='checkbox' name='d$i' value='1'>");
    print("<br>");
}

```

```

print("<p><input type='checkbox' name='dojo' value='1' $dojoflg>
項目を何も無い状態で送ると、上にある項目と同じにする。<br>
#{昼休み|おやつ|帰宅時間}には自動入力しません。
</p>");
print("<p>各行右端のチェックボックスにチェックをいれると、
その行のデータを<b><u>破壊</b></u>します。</p>");
?>

```

```

<input type="submit" size="30" value="送信">
<input type="reset" size="30" value="やり直し"><br><br>
<input type="button" onClick="location.href='./asainDm.php'"
value="本日のまとめページ" style="width:150"><br><br>
<input type="button" value="index ^" onClick="location.href='./index.html'"
style="width:150">

```

```

<?php
print("<br><br>本日のつぶやき（入力ページでは最新の10件のみ表示します）<br>");
$tubu=mysql_query("select tubuyaki,user,RIGHT(time,8)
from asatubu where time like '$dayt%' order by id desc limit 10");
mysql_query("delete from asatubu where tubuyaki='' or user=''");

while($tubuyaki=mysql_fetch_array($tubu)){
    $tuma[$i][0] = $tubuyaki[0];
    $tuma[$i][1] = $tubuyaki[1];
    $tuma[$i][2] = $tubuyaki[2];
    $i++;
}

for($j=0; $j<10; $j++){
    $i--;
    if($tuma[$i][0]!=''){
        print "「";
        print $tuma[$i][0];
        print "」...";
        print $tuma[$i][1];
    }
}

```

```

    print "(";
    print $tuma[$i][2];
    print "<br>";
}
}
print("<input id='intext' type='text' name='t' size='45' autocomplete='off'>");
?>
<input type="submit" value="つぶやく">
<input type="button" onClick="location.href='./asaintubu.php'"
value="今までのつぶやき"><br>
</form>

<h2>注意事項</h2>
<form>
-時限 (変更不可)>項目>内容で入力。 <br>
-SQL インジェクションやタグ挿入を想定していません。 ->
<input type="button" value="プログラマをぶっ飛ばす"
onClick="location.href='./gasu1.gif'"><br>
-2 つ目の項目スペースに文字をいれないと表示しません。 <br>
-'今日' 送ったデータしかまとめページで表示されません。
<br><br>
</form>

<form action="pass.php" method="post">
パスワード変更<br>
<?php print "<input type='hidden' name='pass' value='$pass'>"; ?>
変更前<input id='intext' type='password' name='passhen' size='15'><br>
変更後<input id='intext' type='password' name='passhen1' size='15'><br>
確認用<input id='intext' type='password' name='passhen2' size='15'><br>
<input type='hidden' name='uname' value=<?php print $uname; ?>>
<input type="submit" name="passhenkou" value="変更する">
</form>

<?php
print("</body>");
}else{
print "<title>認証失敗</title></head>";
print "パスワードをうっかり間違えてしまったか、ユーザー名を入れ忘れたようです。
<br>私はそのようなミスをするを想定していなかったなので、認証に失敗しました。";
}
?>

</html>

```

A.5.4 記録の処理 (asainDr.php)

```
<?php
    session_start();
?>
<head>
<link href='./asainD.css' rel='stylesheet' type='text/css'>
</head>
<?php

//mysql を開く
$ms=mysql_connect('localhost','yukawa') or die("mysql への接続失敗");
mysql_select_db('yukawa');

$dt = date("Y/m/d",time());
$dayt = date("Y-m-d",time());
$user = $_POST["user"];
$dojo = $_POST["dojo"];
$t = $_POST["t"];
$username = $_POST["uname"];
$password = $_POST["pass"];
print "<form action='asainD.php' method='post'>";
print "<input type='hidden' name='user' value='$username'>";
print "<input type='hidden' name='pass' value='$password'>";

//前日までのレコードを（一応）削除
//mysql_query("delete from $user where day<'$dt'");

if($user != 'notuser'){
//データ取得
echo $dt;
echo "の記録<br>#時間は全て JST<br>";
$tu = $_POST["t"];
//つぶやき挿入
mysql_query("insert into asatubu(time,user,tubuyaki) values(NOW(),
SUBSTRING('$user',4,12),'$tu')");
for($i=0;$i<$_POST["jigen"];$i++){
    $a = $_POST["a$i"];
    $b = $_POST["b$i"];
    $c = $_POST["c$i"];
    $del = $_POST["d$i"];
    $id = $_POST["id$i"];
    $j = $i - 1;
```

```

//テーブルへの挿入と更新
$dbl = mysql_fetch_array(mysql_query("select time from $user where time='$a'
and day='$dayt'"));
mysql_query("update $user set id='$id' where time='$a' and day='$dayt'");

if($b==''){ //何も入っていない場合の処理
    if($dojo==1 && $j>=0 && $id!=11 && $id!=3 && $id!=7){ //同上処理
        if($a==$dbl[0]){
            mysql_query("update $user set day='$dt' where time='$a' and day='$dayt'");
            mysql_query("update $user set label='$dbstock' where time='$a'
and day='$dayt'");
            mysql_query("update $user set text='$dcstock' where time='$a'
and day='$dayt'");
        }else{
            mysql_query("insert into $user values('$id','$dt','$a','$dbstock',
'$dcstock')");
        }
    }
    if($del==1){
        mysql_query("delete from $user where time = '$a' and day='$dayt'");
    }
}else{ //何か入っている場合の処理
    if($id!=3 && $id!=7){
        $dbstock = $b;
        $dcstock = $c;
    }
    if($a==$dbl[0]){
        mysql_query("update $user set day='$dt' where time='$a' and day='$dayt'");
        mysql_query("update $user set label='$b' where time='$a' and day='$dayt'");
        mysql_query("update $user set text='$c' where time='$a' and day='$dayt'");
    }else{
        mysql_query("insert into $user values('$id','$dt','$a','$b','$c')");
    }
}
}

//表示
if(!$HTTP_SESSION_VARS["s_user"]){
    print "何もあらまへん<br>";
}
print $HTTP_SESSION_VARS["s_user"];
$outdata = mysql_query("select time,label,text from $user where day='$dt'
order by id asc");

```

```

while($dbo=mysql_fetch_array($outdata)){
    echo $dbo[0];
    echo ":";
    echo $dbo[1];
    echo "。 ";
    echo $dbo[2];
    echo "<br>";
}
echo "<br>つぶやき:";
echo $t;

}else{ //ユーザーが選択されていなかった場合の処理
print("<font color='red'>非常に申し訳ないのですが、このページを正しく
表示することは私にはできません。<br>私には、一体誰が項目や内容を入力して
くださったのかがわからなかったのです。<br>大変面倒をおかけしますが、
私の下にあるボタンによって1つ前のページに戻り、<br>
再度ユーザーを選択してから送信していただいてもよろしいでしょうか?<br>
ご迷惑おかけしますが、よろしく願いいたします。
<br>-----V-----
-----<br></font>");
print("<img src='./tousan.jpg' alt='ごめんなさい'>");
}
//mysql を閉じる
mysql_close($ms);

?>
<br><br>
<input type="submit" value="こちらからお戻りくださいませ">
<br><br><br>
<input type="button" value="プログラマをぶっ飛ばす"
onClick="location.href='./gasu.gif'">

</form>

```

A.5.5 1日のまとめ (asainDm.php)

```

<?php
    session_start();
?>

<head>
<link href='./asainD.css' rel='stylesheet' type='text/css'>
<title>本日のまとめ</title>

```

```

</head>
<?php

$ms=mysql_connect('localhost','yukawa') or die("接続失敗");
mysql_select_db('yukawa');

$gyo=mysql_fetch_array(mysql_query("select id from asauser
order by id desc limit 1"));
$day=date("Y/m/d",time());
$dayt=date("Y-m-d",time());
?>
<form><input type="button" value="index ^"
onClick="location.href='./index.html'" style="width:230"></form>
<?php
print $day;

print "<br>本日のつぶりやき<br>";
$tubu=mysql_query("select tubuyaki,user,RIGHT(time,8)
from asatubu where time like '$dayt%'");
while($tubuyaki=mysql_fetch_array($tubu)){
print "「";
print $tubuyaki[0];
print "」...";
print $tubuyaki[1];
print "(";
print $tubuyaki[2];
print "<br>";
}

print "<br>各人の予定<br>";

for($i=1;$i<$gyo[0]+1;$i++){
$user=mysql_fetch_array(mysql_query("select name from asauser where id='$i'"));
$au=mysql_fetch_array(mysql_query("select tb from asauser where id='$i'"));

print $user[0];
print "<br>";
$flg = 0;

$data=mysql_query("select time,label,text from $au[0]
where day='$day' and label not like'' order by id asc");
$format = "%6s";
while($hyo=mysql_fetch_array($data)){

```



```

printf($format,$hyo[0]);
print ":";
print $hyo[1];
print "。 ";
print $hyo[2];
print "<br>";
$flg = 1;
}

if($flg == 0){
print "{欠席|朝会までにデータを送れなかった}<br>";
}
print "<br>";
}

mysql_close($ms);
?>

```

A.5.6 パスワード変更処理 (pass.php)

```

<head><title>パスワード変更</title>
<link href='./asainD.css' rel='stylesheet' type='text/css'>
</head>
<?php

require_once("info/info.php");

//mysql を開く
$ms=mysql_connect($SERVER,$DBUSER) or die("mysql への接続失敗");
mysql_select_db($DATABASE);

$pm = $_POST["pass"];
$p = $_POST["passhen"];
$p1 = $_POST["passhen1"];
$p2 = $_POST["passhen2"];
$user = $_POST["uname"];

if($p1 == $p2 && $pm == $p && $p1 != ''){
    $p1 = hash('md5',$p2);
    mysql_query("update asauser set pass='$p1' where name='$user'");
    print "パスワードを変更しました。 ";
}else{
    print "確認用パスワードが違います。<br>そうでなければ恐らくあなたは

```

何も入力せずに変更ボタンを押したのでしょう。

```
<br>さすがにそれは私の想定範囲を超えています!";  
}
```

```
//mysql を閉じる  
mysql_close($ms);
```

```
?>  
<br><br>  
<form><input type="button" value="index へ" onClick="location.href='./index.html'"  
  style="width:180">  
<br><br>  
<input type="button" value="プログラマをぶっ飛ばす" onClick="location.href='./gasu.gif'">  
  
</form>
```

A.5.7 つぶやきまとめ (asaintubu.php)

```
<head>  
<link href='./asainD.css' rel='stylesheet' type='text/css'>  
<title>今までのつぶやき</title>  
</head>  
<?php  
  
$ms=mysql_connect('localhost','yukawa') or die("接続失敗");  
mysql_select_db('yukawa');  
  
print "<h1>今までのつぶやき</h1>";  
?>  
<form><input type="button" value="index へ"  
  onClick="location.href='./index.html'" style="width:230"></form>  
<?php  
$count = 0;  
$date=mysql_query("select left(time,10) from asatubu  
  group by left(time,10) order by time desc");  
  
while($days=mysql_fetch_array($date)){  
  print "<br>$days[0]<br>";  
  $tubu=mysql_query("select tubuyaki,user,time,LEFT(time,10)  
  from asatubu where left(time,10)='$days[0]' order by time asc");  
  while($tubuyaki=mysql_fetch_array($tubu)){  
    print "「";  
    print $tubuyaki[0];
```

```
print "「 ...";
print $tubuyaki[1];
print "(";
print $tubuyaki[2];
print "><br>";
}
}
mysql_close($ms);
?>
```

A.5.8 全体のCSS(asainD.css)

```
body{
  line-height: 19pt;
  letter-spacing: 1px;
  background-color: #AAAAAA;
  color: #000000;
}
label{
  display:block;
  text-align: right;
  float: left;
  width: 60px;
  padding-right: 2px;
  text-align: left;
}
option{
  color: #000000;
  background-color: #FFFFFF;
}
h1{
  font-size: 120%;
  border-bottom: 1px dotted #000000;
  letter-spacing: 1px;
}
h1:first-letter{
  font-size: 1.5em;
}
h2{
  font-size: 110%;
  border-bottom: 1px dotted #000000;
  letter-spacing: 1px;
}
```

```

h2:first-letter{
  font-size: 1.5em;
}
h3{
  font-size: 105%;
  border-bottom: 1px dotted #000000;
  letter-spacing: 0.6px;
}
h3:first-letter{
  font-size: 1.5em;
}
#intext{
  height: 22px;
  border-top: none;
  border-right: 2px #000000;
  border-left: none;
  border-bottom: 2px #000000;
  background-color: #FFFFFF;
  padding: 3px 5px;
  color: #000000;
}
#timetext{
  height: 22px;
  border: none;
  padding: 3px 5px;
  background-color: #FFFFFF;
  font-align: right;
  color: #000000;
}
input[type="text"],
textarea{
  -moz-border-radius: 5px;
  -webkit-border-radius: 5px;
  height:10px;
}
input{
  border-style:double;
  border-width:3px 0px;
  border-color:#000000;
  background-color:#FFFFFF;
  color:#000000;
}
input[type="password"],

```

```
textarea{
  -moz-border-radius: 5px;
  -webkit-border-radius: 5px;
  height:10px;
}
input{
  border-style:double;
  border-width:3px 0px;
  border-color:#000000;
  background-color:#000000;
  color:#FFFFFF;
}
```