ISSN 0915-6372

NATIONAL INSTITUTE FOR FUSION SCIENCE

一時アカウントFTPサービスによるファイル交換システムの開発

Development of a File Exchange System Based on the FTP Service Using Temporary Accounts

山本孝志

T. Yamamoto

(Received - Mar. 1, 2007)

NIFS-MEMO-53 Mar. 2007

RESEARCH REPORT
NIFS-MEMO Series

TOKI, JAPAN

一時アカウント FTP サービスによる ファイル交換システムの開発

山本 孝志
核融合科学研究所 計算機・情報ネットワークセンター

Development of a file exchange system based on the FTP service using temporary accounts

Takashi YAMAMOTO
Computer and Information Network Center, National Institute for Fusion Science

Abstract

A file exchange system based on FTP service named cftp (Computer Center's FTP service) was developed for exchanging the files which is too large to send by e-mail on July, 2003. The user can privately exchange the files with this system which strictly separates the user's directories. This separation is made by the "guest user" function added by wu-ftpd, the famous FTP server software.

The account on cftp system is rapidly issued when the user sends a request e-mail to the system. The account is valid only for seven days for avoiding the security problems. This system comprises the UNIX account system, wu-ftpd, and Perl script.

This service has been offered to the community of *National Institute for Fusion Science* for three years and six months. On this report, the outlines of the script program for the system are explained. The usability and the next generation system are also discussed on the basis of the answers to a questionnaire.

keywords: private communication, Except module, network administration, automation

概 要

電子メールでは添付することができない大容量のファイルの送受信を目的としたファイル交換システム（Computer Center's FTP service, cftp）を 2003 年 6 月に FTP サービスを元にして開発した．利用者のファイルスペースを相互に独立させることを著名な FTP サーバソフトウェア wu-ftpd のゲストユーザ機能を用いて実現し，安全な個人間のファイル交換を実現させた．

本システムのアカウントは利用者の電子メールによる請求により随時に発行するが，恒久的なアカウントを発行するとセキュリティ上の懸念があるため，アカウントの有効期間は 7 日間としている．このシステムは UNIX OS 上に wu-ftpd をインストールし，その環境を Perl 言語によるスクリプトで生成している．

本報告では，システムの実装方法の説明の他，3 年 6 ヶ月にわたり核融合科学研究所で本サービスを提供してきた利用状況（利用者のアンケートを含む）や次世代のシステムについての考察を行う．

1 開発方針

1.1 背景

すでにインターネットは日常生活に浸透し，研究や業務で情報交換を行う際にはかかせなくなっている．書類を送る場合にも，以前ならば郵送で一日単位の時間がかかっていたが，現在では電子メールで送付することがほとんどであり，海外に対してもほぼ数分で届く．その際の送付にかかるコストもわずかである[*1]．電話も重要なコミュニケーションツールであるが，正確に情報を記録するには録音などを行う必要があり，また通話料も比較的高価であるため，情報を交換するツールとしては電子メールが優位である．

しかし，一般に電子メールで書類やデータを送付する場合には容量に上限があり，これを越えて送る場合は，送信者側で圧縮や分割などを行う必要がある．例えば，当研究所（核融合科学研究所）の代表メールサーバでは 1 通当り 10 MB という制限を行っている．制限する理由は，事故などより誤って非常に大容量のメールが送られた場合，メールサーバ全体に障害が起き，結果として他の利用者のメールの送受信が停止することを回避するためである．

そのような大容量のデータは組織内（敷地内）ならば，USB メモリなどで持参すればよいが，国内外に存在する研究者とのデータの送受信は，やはりインターネットの利用が簡便である．そこで，電子メールと共に歴史がある FTP サービス [*2] を活用したファイル交換システム cftp を開発した．このシステムの特徴は，利用者の使用可能なエリアは独立しており，利用者間でどのようなファイルがやり取りされているかがわからない点である．また，ファイルの送受信の際に用いられるアカウントは利用者の請求により自動発行され，一定期間後には自動削除されるまでの間，管理者による作業は発生しないという利点もある．

以下，本報告では，電子情報を交換する各種の方法を俯瞰した後にシステムの開発目標を述べ，第 2 章でシステム開発とその実装方法について述べる．第 3 章で利用状況と利用者アンケートを踏まえて今後のファイル交換システムについて考察を行い，最後にまとめを述べる．付録として，システム開発の補足とアンケートの回答を記す．

[*1] これが迷惑メール（スパム）の温床ともなっている．

[*2] File Transfer Protocol. RFC959.

1.2 電子情報の送付方法の比較

個人間で電子情報を送付する方法について，おおよその送付サイズと所要時間を表 1 にまとめた．匿名 FTP サービスは個人間の送付を目的としないが，本システムの説明のために加えた．

表 1　電子情報を転送する各種方法の比較.

手法・媒体	送付サイズ	転送時間	必要なもの
郵便（電子メディア）	～数 10GB	1 日～数日	メディア（DVD-R, USB メモリ）
電子メール	～数 MB	～数分	メールクライアント
無料ファイル転送サービス	数 10MB～100MB	～ 十数分	Web ブラウザ
Anonymous FTP サービス	～数 100MB	～ 十数分	FTP クライアント（Web ブラウザ）

■郵便（外部メディア）　情報を USB メモリ, CD-R, DVD-R，磁気テープ，外付けハードディスク等にコピーした後に郵送や宅配便で相手先に送付する方法である．利点は確実に送れることがだが，欠点は物理的距離に比例して送料や配送にかかる時間が大きくなることである．特に，国際間の場合はそれらの増加が著しくなり，状況によっては行方不明になる可能性がある．また，送受信のいずれにおいても，使用する端末から外部メディアへの情報の出し入れに相応の時間がかかる．

■電子メール　電子メールによる送付は冒頭に述べたように，現在，最も普及している方法である．手軽に電子情報を送付することができるが，一般的に，一度に送付できるファイルサイズには制限があり，制限を越えたサイズのファイルを送付するためには，送信者側でファイルの分割等を行う必要がある．

■無料ファイル転送サービス　Web ブラウザを利用してファイルを送付できるサービスである．手軽であり，ファイルの公開を目的とする「アップローダー」と違い，送付情報等の秘密を第三者に対して守れるが，システム毎に利用者登録が必要であること，他の利用者との兼ね合いでダウンロード速度が極端に遅くなる場合があるなどの難点がある．

■Anonymous FTP サービス（匿名 FTP サービス）　FTP サービスは，その名の通りファイルの送受信を目的とした古くからあるサービスである．FTP サービスを利用するためには，専用の FTP クライアントが必要だったが，最近では，Web ブラウザなどでも利用できる．特に Anonymous FTP サービスは，アカウント名として `ftp` あるいは匿名を意味する `anonymous` を，パスワードとして**自分の電子メールアドレス**を入れると誰でも利用可能となる FTP サービスである[*3]．

ファイルのダウンロード（サーバから自分の端末への受信）は誰でもできるが，ファイルのアップロード（自分の端末からサーバへの送信）は特定のディレクトリのみに限られ，ファイルの公開にはサーバ管理者の介在が必要である．インターネット黎明期ではこの Anonymous FTP サービスが情報交換の中心であった．個人のサイトでも多くの Anonymous FTP サービスが提供されたが，アップロードしたファイルがそのまま第三者にもダウンロードできるような設定が残され，結果として違法なファイル交換所となったサイトが多数あった．

[*3] Anonymous FTP サービスのパスワードは任意でよいが，サーバ管理者に誰が使用したかを伝えるため自分の電子メールアドレスを入れるのがマナーである．

■その他　これらの Web サービスや FTP サービスを自分で管理する行う方法がある．自分が管理者であるため，事実上，ファイル容量に制限がなく，力量さえあれば高機能なサービスも提供できる．一方，組織内においては，そのセキュリティポリシーに従う必要があり，必ずしも対外的にそれらのサービスを公開できるとは限らない．また，セキュリティ対策などサーバを管理する手間が常にかかるため，あまり一般的ではない．

1.3　開発目標

これらの状況を踏まえ，セキュリティに考慮しながらも，利用者，管理者ともに使用する際の手間が負担にならないよう留意して本 cftp システムを開発した．以下にシステムの主な機能を利用者の観点から述べ，システムと利用者との関係を図 1 に示す．

- 利用者は本システムへのアカウント開設の要求をメールで行い，システムで作成されたアカウント情報をメールで受け取る．
- 利用者はメールの返答に書かれたアカウント情報を元にファイルを `ftp` で転送する．
- 利用者は必要に応じて, アカウント情報をファイルの送付先に伝える．
- 7 日後にシステムは利用者のアカウント情報, ならびにアップロードしたファイル等すべての情報を削除する．
- 利用者は，他の利用者からその存在を知られない．すなわち，利用者は，他に利用者がいるかどうか，また，その利用者が扱っているファイルに関する一切の情報を知ることはできない．

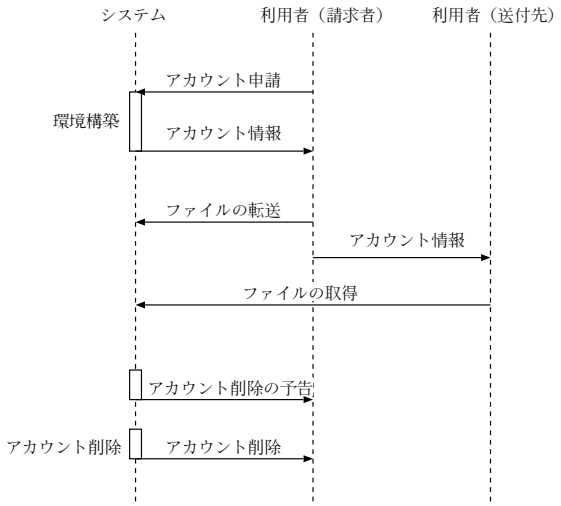

図 1. cftp システムの動作シーケンス図.

2　システム開発

2.1　システム概要

cftp システムを実現させるために，すでに FTP サーバとして稼動していた Sun ワークステーションに，外付け SCSI ハードディスク 1.6GB を加えた．FTP サーバソフトウェアとして wu-ftpd を利用し，約 460 行の Perl 言語 [1] によるスクリプトを開発した．なお，利用者毎に利用可能なディスク容量については特に管理しておらず，利用者全体でこの外付けハードディスクのディスク容量が制限となる．

cftp システムは受付用アドレスで利用者からのアカウント設置請求の電子メールを受領すると，FTP 用アカウントを UNIX のアカウント作成コマンド，ならびに Expect.pm モジュールを利用して作成し，その結果を電子メールで通知する．作成されたアカウントは FTP サービスのみに有効となるよう修正されている．その後，7 日間の有効期間を過ぎたアカウントは，同じく UNIX のアカウント削除コマンドにより消去する．

アカウント作成の際には，差出人アドレス（From 行のアドレス）を解析し，研究所のドメイン名である

nifs.ac.jp を含まない場合は処理を拒否する．これは，誰でもアカウントが作成できると不法なファイルの交換所になることが予想されるためである．[*4]．ファイルの送付先に関しては利用者に依存しており，利用者の判断により通知されたアカウント情報を伝えることになる．

なお．有効期間を更新する機能についても一時検討したが，システムをシンプルにするため実装しなかった．必要な場合は，再度，アカウントを作成した後にファイルのアップロード等を行う必要がある．

以下，主な機能と，その実装方法について述べる．

2.2 スクリプトの実行環境

cftp システムでは，アカウント請求の電子メールを受信すると電子メールプログラム（sendmail）より利用環境を整えるプログラムが起動される．UNIX のアカウント作成には管理者権限が必要であるが，一方，sendmail は一般ユーザ権限であり，メールから呼び出されたプログラムは管理者権限を得ることはできない．UNIX ではファイルに特殊なビット（SUID ビット）が付与されていれば，実行したユーザの権限ではなく，そのファイルの所有者の権限で実行できる機能がある．しかし，ここではセキュリティに配慮し，SUID ビットが付与された C 言語によるラッパー（wrapper）プログラムを経由して呼び出すことにした[*5]．参考のため，ラッパープログラムのファイル属性，ならびにソースコードを以下に示す．

/var/cftp/bin

```
-rwxr-xr-x   1 yama     staff      14016 Feb  1 23:20 cftp.pl*
-rwsr-xr-x   1 root     other      23672 Jun  9  2003 wrapper*
```

wrapper.c

```
/* p. 414, Programming Perl Second Edition, Japanese Edition */
#define REAL_FILE "/var/cftp/bin/cftp.pl"

#include <stdio.h>
#include <unistd.h>

main(int argc, char *argv[]) {
  execv(REAL_FILE, argv);
}
```

2.3 アカウントの作成

2.3.1 アカウントの設定; Expect

UNIX のアカウント情報はパスワード情報を除きテキストファイルで保存されているため，直接書きかえることも容易に行えるが，ここでは UNIX コマンドの useradd を用いて省力化をはかった．アカウントを作成した直後はパスワードが定められておらずログインできない状態であるため，予めパスワードの設定を行う必要がある．パスワードを管理しているファイル/etc/shadow に直接暗号化したパスワードを書きこむ方法もあるが，アカウント作成と同様に UNIX コマンド passwd を用いて省力化をはかった．ただ，この passwd コ

[*4] メールのヘッダ行は一般に書き替え可能であり，所員以外の第三者が From 行のアドレスを nifs.ac.jp に含むアドレスに偽装できるが，その場合は偽装したアドレス，すなわち，研究所のアドレスに届くため，第三者へアカウント情報が届くことはない．

[*5] Perl に限らずスクリプト言語全般について SUID ビットによる管理者権限の実行は UNIX カーネルのバグを引きおこす可能性がある．詳細については参考文献 [1] の 6.3.1.3 節 「セキュリティのバグ」，ならびに [2] を参照のこと．

マンドはパスワードを対話的に入力する必要があり，通常のスクリプトでは処理することはできない．そこで，対話的処理を Expect.pm モジュールを用いることにより実現した [3]．その概要を示すため，以下に初期パスワードを設定するサブルーチンを示す．

cftp システムより抜粋

```
sub init_passwd {
# Expect module
# cf. Managing Multiplatform Enviroments with Perl, Japanese Edition.
#     D. N. Blank-Edelman, O'Reilly Japan, 2002, page 95
  my ($account, $passwd) = @_;

  # コマンド（$passwd_cmd に $account を引数として与える）を起動する．
  my $command = Expect->spawn($passwd_cmd, $account)
    or die "Can't start program... $!\n";

  # コマンドからの出力を標準出力へ表示しない．
  $command->log_stdout(0);
  # コマンドから "New password:"と返答されることを 10 秒間は待つ．
  $command->expect(10, "New password:");
  # コマンドへ $passwd を入力する．
  print $command "$passwd\n";
  # コマンドから "Re-enter new password:"と返答されることを待つ．
  $command->expect(10, "Re-enter new password:");
  # コマンドへ $passwd を入力する．
  print $command "$passwd\n";
  # コマンドの終了．
  $command->soft_close();
}
```

このように Except.pm を用いると他のマシンへ自動的にログインを行うスクリプトも容易に作成できるため，異機種混在下での自動処理に非常に威力を発揮する．

2.3.2 システムのアカウント管理

UNIX のアカウントを作成する他に，アカウント毎に有効期限や連絡先アドレスを管理する必要がある．非常に頻繁に使われるシステムならばデータベースで管理する必要があるが，実際の需要は多くて一日に数件と推測されたためテキストファイルで管理している[*6]．

アカウント管理表 account.tab

# 有効期限	ユーザ名	通知メール先	消去済み
2006/12/07 23:59:59	foo091	foo@nifs.ac.jp	expired
2006/12/08 23:59:59	bar092	bar@nifs.ac.jp	expired
2006/12/09 23:59:59	baz093	baz@nifs.ac.jp	

有効期限はアカウントを作成した際に算出している．アカウントを削除する処理では単純に現在の時刻と比較するよう設計した[*7]．

[*6] 情報をテキストテキストで管理することは古き良き UNIX の伝統でもある．

[*7] メールアドレスの foo, bar, baz 等は実在しないアドレスである．

2.3.3 wu-ftpd の設定

■ゲストユーザ機能 wu-ftpd には，ゲストユーザという機能がある．これはもともと Anonymous FTP サービスの概念を受け継いだものであり，ゲストユーザの対象となった利用者は，ftp で接続しても所定のディレクトリより上位のディレクトリに移動等のアクセスができなくなる．この機能により，システムの設定ファイルや他の利用者のファイルなどの存在を隠すことができる．

また，UNIX には，実行プログラムの最上位ディレクトリ（ルートディレクトリ）を強制的に変更する chroot コマンド（システムコール）がある．wu-ftpd のゲストユーザ機能もこれを利用しているが，システム関係のファイルもアクセスできなくなるために，FTP サービスに必要となる ls コマンド等もそのディレクトリの下に配置する必要がある．そのため，ここでは関連するダイナミックライブラリ等をも含めコマンド一式をアカウント毎にコピーしている．

wu-ftpd におけるゲストユーザの設定方法は，inetd からの起動時に，

/etc/inetd.conf
```
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd          in.ftpd -l -a
```

と，-a オプションを付与することにより，設定ファイル/etc/ftpaccess の読込みを指示する[*8]．ftpaccess では，次のように利用者名（アカウント名）を列挙して指定すればよい．

/etc/ftpaccess
```
guestuser foo091 bar092 baz093
```

今回のシステムの様に頻繁にユーザ名が変更される場合は， ゲストグループの機能を使用すると便利である．

/etc/ftpaccess
```
guestgroup cftp
```

この設定により，UNIX アカウントのグループが cftp に所属するユーザすべてに対しゲストユーザ機能が有効になる．

■ルートディレクトリの指定 具体的にどのディレクトリがその利用者のルートディレクトリになるかを，/etc/passwd ファイルにて決めることができる．

/etc/passwd
```
foo091:x:20091:20000:ftp user:/public/foo091/./pub:/bin/false
```

このコロン ":" で区切られた第 6 カラム目の "." より前がルートディレクトリとなり，"." 以降のディレクトリがホームディレクトリとなる．この例では，foo091 が ftp による接続した直後のディレクトリは /public/foo091/pub であり，他のディレクトリへ移動を試みてもルートディレクトリ/public/foo091 より上位へは移動することはできない．

2.3.4 利用者への通知

利用環境の整備が終了すると，cftp システムはアカウント情報を利用者へメールで通知する．

[*8] wu-ftpd は第 6 引数にあるよう tcpwapper（TCP/IP レベルで柔軟に接続許可を制御とするプログラム．これにより inetd から起動されるサービスでも簡単に接続先を所内のみなどと限定できる）を経由して起動されている．また，-l オプションは ftpd のやりとりを OS のログ（syslog）に記録することを指示する．

利用者への通知

```
Your request is accepted.
        hostname: ○○○.nifs.ac.jp
        username: foo091
        password: xxxxxxxx.yyyyy

        expiration: 2006/12/07 23:59:59 JST (+0900)

Your account is valid for 7 days.
All uploaded files will be automatically removed after the valid period.
```

2.4 アカウントの削除

有効期限が切れたアカウントの削除は，アカウントの作成処理とは独立に UNIX の cron から毎日深夜 1 時に起動されて処理を行っている．

crontab

```
# cftp expiration
0 1 * * * /var/cftp/bin/cftp.pl --expire | mail foo@nifs.ac.jp 2>&1
```

その際，アカウント管理表（account.tab）を参照し，消去済みのマーク（第 5 項目 expired）がついていないアカウントに対し現在の時刻と比較して，有効期限が切れているアカウントを UNIX の usrdel コマンドを呼び出して削除する．その際，usrdel コマンドには-r オプションを指定し，アップロードされたファイルも合わせて消去する．削除済みのマークの記入はアカウントの削除後に行っている．これにより停電など何らかの理由でアカウントの削除ができなくても，後日の処理の際に未処理のアカウントを検出して確実に削除が可能となる．

処理の最後に，管理者へアカウントの管理状況を電子メールで報告する．

管理者への報告

```
Expire              Username       E-mail                  Status   Rest
-------------------------------------------------------------------------
2006/12/07 23:59:59 foo091         foo@nifs.ac.jp          expired
2006/12/08 23:59:59 bar092         bar@nifs.ac.jp          expire  -0.04
2006/12/09 23:59:59 baz093         baz@nifs.ac.jp          alive    0.96
2006/12/10 23:59:59 qux094         qux@nifs.ac.jp          notice   1.96

* The notice is sent 2 days before.
```

3 利用状況と今後の課題

3.1 利用状況

cftp システムは 2003 年 6 月にサービスを開始して以来，安定した稼動を続けている．表 2 に 2003 年 6 月から 2006 年 12 月末までにおける年間利用者数を示す．平均では 55 件/年（4〜5 件/月）の需要が続いている．

利用状況や問題点を探るために，2006 年 3 月に利用者アンケートを行った．付録に設問，ならびに，回答を示す．回答は総じて好意的な意見が多い．特に利用方法が簡単だという意見が特徴的であり，本システムを開発する上での目標が十分に達成されていることが示されている．また，研究者のみならず管理部など研究所

表 2　cftp 年間利用者数（2003 年 6 月〜2006 年 12 月）

年	2003 年	2004 年	2005 年	2006 年	合計
利用数	24	59	58	57	198

重複利用を含む．平均 55.3 件/年.

の多くの部署で活用されていることも推測される．一方，パスワードの不具合などの改善点の指摘も寄せられており，一部については対応することができた．

3.2　新システムへの課題

cftp システムは運用開始から 3 年半が過ぎ，ハードウェアの能力[*9]が，現在ではより安価に購入できる PC よりも劣るようになっており，特にハードウェアの更新が必要である．

ソフトウェア的には，ベースとして現在の FTP サービスを用いるか，Web サービスを用いるかの選択がある．また，現在のシステムのアカウントの作成は，電子メールによる申請をトリガーとしているが，最近は Web ブラウザによる動的な操作が好まれている[*10]．一方，アカウント作成の通知方法や消去の予告等は現行の電子メールによる方法が，利用者の手元に残るという観点から好ましいと思われる．ソフトウェア内部の留意点としては，現在はテキストファイルで管理しているアカウントの管理方法がある．データベースの管理というと最近では MySQL や PostgreSQL 等のフリーウェアの SQL データベースが良く用いられるが，前節に示したよう使用頻度が月に 4 件ほどであるため，この観点からはデータベースを用いなくても問題がないように思える．ただ，クラスター構成などシステムの信頼性を上げる場合には，結果としてこれらのソフトウェアを用いた方が構築は容易かもしれない．

アンケートでの指摘にもあるよう，現在のシステムではファイル名に日本語を用いると，いわゆる文字化けが発生することがある．これは日本語の漢字コードが OS により異なることによる[*11]．しかし，最近は，すべての文字コードを一つの体系とした Unicode が普及して来たため，新システムを Unicode 対応とすれば解決する問題と考えられる．

最後に，本システムにより発行されたアカウント情報をファイルを送る相手側にネットワークの盗聴に対しても安全に伝える方法について考察する．アカウント申請者へアカウント情報をシステムから安全に伝える方法は比較的容易で，利用者の Web ブラウザへデータを SSL（Secure Socket Layer）により暗号化して送ればよい．一方， 送付の相手方に安全にアカウント情報を送る簡単な方法は今のところ見当らず，別のメディア（電話，FAX, 封書）によるか，利用者間で暗号化メール（S/MIME, PGP）を使うなど，いずれにせよ費用や手間などのコストがかかる．元々，本システムは電子メールで送られない大容量のファイルを送付することが目的であるため，相手方に送付できたことが確認されたら速やかにアカウントを無効にするのがよく，この機能を備える方がより現実的であろう．さらに，非常に重要なデータは，別途暗号化して USB メモリ等で送付し，その暗号のパスワードは本人に電話などで直接伝えるべきだと考える．

[*9] 現在のハードウェアは機種名 Fujitsu S-4/5H model 170, CPU TurboSPARC 170MHz, メモリ 64MB，cftp 用ハードディスク 1.8GB, ネットワークインターフェース 10BaseT．しかし，ワークステーションだけあって，PC より故障率が低く感じる．

[*10] Web ブラウザから動的に処理を勧める方法として Asynchronous JavaScript + XML (Ajax) と呼ばれる技術がある．

[*11] マイクロソフト社 Windows は Shift-JIS, UNIX は EUC.

4 まとめ

電子メールで送ることができない大容量のファイルを送付するために，一般的な UNIX 環境のアカウントシステムと wu-ftpd サーバプログラムの特徴を生かしたファイル交換システム cftp を作成した．このシステムの特徴は，利用者は他の利用者からの影響を受けることがなく，どのようなファイルを交換しているかを第三者に知られないことである．

cftp システムの運用開始から 3 年半が経過するが，大きな障害に遭遇することなく，また，利用者も連綿と途切れることなく安定したサービスの提供を続けることができた．さらに新システムに備えるべき要件を，これまでに寄せられた改善案や今回のアンケート結果を元に考察した．今後も使い易いサービスを提供する予定である．

謝 辞

cftp システムの利用者対応に当って下さった計算機・情報ネットワークセンターのネットワークスタッフの皆様に感謝します．また，ご多忙の中，利用者アンケートに答えて下さった所員の皆様に感謝します．

参考文献

[1] Larry Wall, Tom Christiansen, and Randal L. Schwartz,「プログラミング Perl 改訂版」, オライリー・ジャパン, 1997.

[2] Simson Garfinkel, and Gene Spafford, “Practical UNIX and Internet Security, Second Edition”, O’Reilly, 1996. SUID など UNIX におけるセキュリティ全般について有益な情報が記されている．

[3] David N. Blank-Edelman,「Perl によるシステム管理」, オライリー・ジャパン, 2002．もともと Expect は tcl というスクリプト言語の拡張機能として作成された．その機能を Perl に移植したのが Expect.pm である．

付録 A　開発したシステムに関する補足

A.1　アカウントの作成規則

利用者の入力の負担を減らすために，cftp システムが自動発行するユーザ名とパスワードは，以下の規則を用いている．

■ユーザ名　ユーザ名は，電子メールの from アドレスに，本システムが持つ通し番号を付与して生成している．ユーザアドレスがフルネーム[*1] である場合は，ドットの前を採用している．

例） `foo@nifs.ac.jp` → `foo001`，`foobar.baz@nifs.ac.jp` → `foobar002`

■パスワード　パスワードは意味のないランダムな文字列とせずに，UNIX に備えられている辞書[*2] より予め「5 文字の単語」，ならびに，「7〜9 文字の単語」をそれぞれを保存し，これらより 1 単語ずつ選び「区切り子」で連結することにより生成している．区切り子は 10 種類の記号より 1 文字が選ばれる．また，最初に来る単語が「5 文字」であるか，「7〜9 文字」であるかも乱数により選んでいる．詳細な計算は省くが，この方法により約 5 億通りのパスワードを発行することができる．

A.2　ProFTPD によるゲストユーザの指定

wu-ftpd 以外の著名な FTP サーバとして ProFTPD[*3]がある．豊富な機能，柔軟な設定方法が可能であり近年の主流となっている．ProFTPD 自身で `chroot` 時の `ls` 機能を実現でき，ゲストユーザの機能と同等の設定を次の一行で行うことができる．

proftpd.conf
```
DefaultRoot ~/pub
```

この設定により，接続したユーザは自分のホームディレクトリの `pub` ディレクトリより上位へは進めなくなる．

現在 wu-ftpd の開発は事実上凍結されていることもあり，今からシステム開発を行う場合は ProFTPD を用いる方が望ましい．しかし，cftp システムが稼動している Solaris 2.5.1 では `chroot` 時の `ls` 機能が働かず，wu-ftpd と同様にユーザ毎にコマンドをコピーする必要があったため，当時の時点では実績が豊富だった wu-ftpd を使用した．

[*1] 当研究所では，所員全員にフルネームを元にした代表メールアドレスを発行しており，ドットの前が姓となる．

[*2] `/usr/dict/words`，あるいは`/usr/share/dict/words` などに存在する．パスワードの候補として使用される辞書は，目視によりタブーワードを除いてある．

[*3] http://www.proftpd.org/

付録 B　利用者アンケート

2006 年 3 月に利用者アンケートを行った．その時点までの利用者は重複を除くと 37 名であり全員に回答を依頼した．回答は 8 件であり回収率は 22% と低いが貴重な意見を得ることができた．

B.1　設問

「cftp:一時アカウントによる FTP サービス」のアンケート

Q1. 使ってみて便利でしたか？（不便，普通，便利，忘れた）
Q2. 使ってみて簡単でしたか？（困難，普通，簡単，忘れた）
Q3. 機会があればまた使いますか？（使わない，場合による，使う，わからない）
Q4. 良かった点を御記述して下さい．（　　）
Q5. 悪かった点を御記述して下さい．（　　）
Q6. その他，ご意見がございましたら御記入下さい（欲しい機能など）．（　　）

B.2　回答

■Q1.〜3.　選択設問 Q1〜3 について，表 3 にまとめた．ほぼすべてが肯定の回答であった．

表 3　アンケート結果

設問	肯定	中立	否定	忘却
便利	6	0	0	0
簡単	5	1	0	0
再度使うか	6	0	0	0

■Q4. 良かった点

- 詳細を忘れてしまいましたが，特にストレスを感じた記憶が無いので，それがよかった点と言えるでしょう．
- アカウントの依頼をして直ちに発行できる．面倒な手続きが必要でない．普通のブラウザで簡単に利用できる．ユニックスコマンドを使うことなく簡単に巨大なファイルを送受信でき，便利に使わせていただきました．管理部の人にも使えるようになっていただいたため（FTP サイトを開設できるシステムが存在すること，その使い方を最初に一度だけ教えました），より一層便利でした．国際会議や研究会などのプレゼンテーションファイルをあらかじめ受信しておくことができるので，共同研究などの運営上便利でした．
- 大量のデータを簡単に送ることができました．（4 件）

■Q5. 悪かった点

- パスワードをメールでやり取りするのがちょっと気持ち悪いですね．まぁ，問題はないと思いますが．
- 初心者向けの使用マニュアルのようなものはあればもっと便利であったと思います（日本語と英語とで）．たとえば，ファイル名に日本語が使えないという制限なども，どこかに記述されているとよい．
- セキュリティが強固な外部の機関からはアクセスできなかった．
- ファイルを添付（保存）したと聞いて，確認したら，添付されていなかった．2，3 回同席して確認しながら進めましたが，添付したはずが,...
- パスワードなしでも一応ログインできてしまいます[*4]．また，ftp で自分が置くファイルやフォルダ以外にも（システム関係の）ディレクトリが見えてしまうため，初めて利用すると戸惑ってしまいます．できれば，アップロードしたファイル以外は見えないようにしていただけるとうれしく思います[*5]．

■Q6. その他

- プラズマ・核融合学会の論文投稿等にも利用できるようにしてはどうかと思います．
- 今後も継続してほしい．現在 Macintosh で利用されているブラウザ サファリは，FTP サービスをサポートしていないようです．マックユーザ向けにインストラクションも欲しいところです．シェアウェアの Fetch を使えるようにしておくのも一つの手かもしれません．
- 大変便利に使わせて頂きました．
- パスワードに ":" が含まれているとき，Web ブラウザの ftp で使用できない[*6]．
- 使用できる期間がフレキシブルになれば良いと思います．
- 一週間しか使えないというのが短すぎる様な，... 安全面をみたらその方が良いかも知れないが，...

[*4] ユーザ名，パスワードなしで接続すると同じサーバに稼動している Anonymous FTP に接続される．あるいは，パスワードを間違った場合でも FTP クライアントが終了しないことを指しているの可能性がある．

[*5] 対応済み．

[*6] 対応済み．ブラウザから FTP サーバにアクセスする際の URL 構文は ftp://username:passwd@server/となるため，パスワードに ":" が含まれていると構文エラーとなる．なお，マイクロソフトのセキュリティ情報 MS04-004 に対するセキュリティ修正プログラムを適応すると，http(s) でのユーザネーム・パスワード情報を埋め込む構文が無効になるが，FTP サービスでは依然有効である．