ＵＮＩＸ入門

１．目的

UNIX は、現在広く使われているオペレーティングシステムの一つである。コンパクトで移植性に優れており、パソコンからワークステーション、メインフレームにいたるまでいろいろな計算機に対し、シンプルで便利な計算機環境を同じように提供している。本演習では UNIX を自在に使えるようになることを目的とする。

２．ＵＮＩＸ入門

2.1 まず始めに

使用するワークステーション(WS)のホスト名(ikura または tara)、ユーザーＩＤ(i 学籍番号)を確認する。

2.2 ログイン

パソコンのデスクトップにある Tera Term アイコン 

をダブルクリックする。すると図1のような host 名選択画面が現れるので、Host:欄に接続したいホスト名(ikura または tara)を入力し、OK ボタンをクリックする。

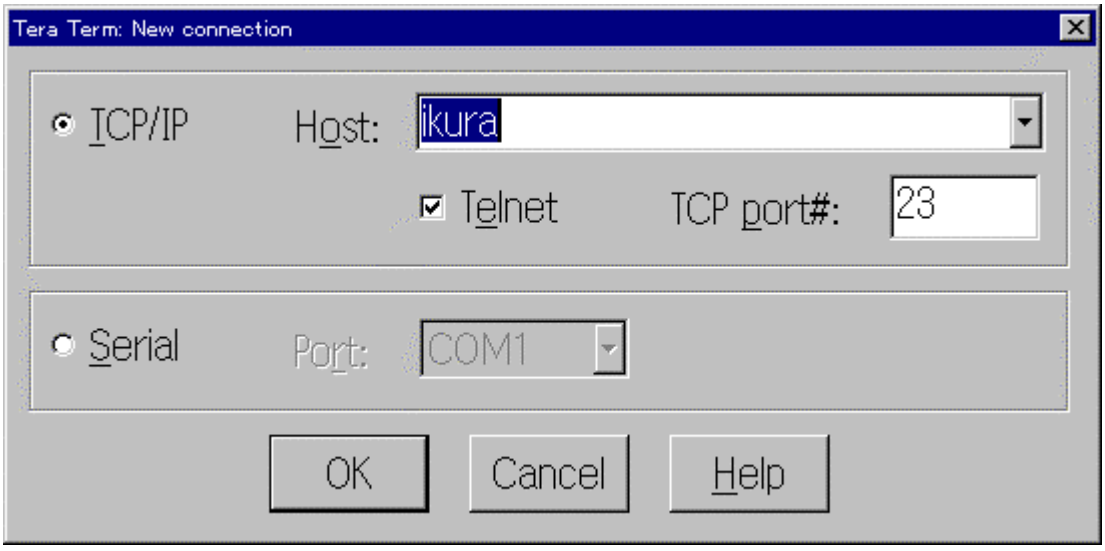

図 2 host 名選択画面

ログインを促すプロンプトが表示されますから、次のようにユーザー名とパスワードを入力します。

login : **i95100**　　--- ユーザー名の入力
password : ○○○○　　--- パスワードの入力

%　　--- ＷＳに入った。(csh が起動された)

2.3 パスワードの登録／変更

まだパスワードを登録していない場合や変更したい場合は以下の手順でパスワードを設定します。

```
% yppasswd
Changing NIS password for kousen on ikura.
Old password:          --- 今までのパスワードを入力します。
                           (初回は、必要ありません)
New password:          --- 新しいパスワードを入力します。
Retype new password:   --- もう一度、新しいパスワードを入力します。
NIS entry changed on ikura
%
```

これでパスワードが登録／変更できました。

```
% exit
```

これでログアウトしますから、もう一度ログインしてみて下さい。

2.4 ディレクトリの表示、ファイルのコピー、削除

```
% ls              --- ディレクトリの表示
                     （ファイルがないので何も表示されない）

% ls  -a          --- 隠しファイルも表示（隠しファイルは「. 」で始まる）
.         ..         .cshrc     .forward   .login

% cp    .cshrc     cshrc--- ファイルのコピー
       コピー元 コピー先

% ls                    --- 確認
cshrc                  --- ちゃんとコピーされている
% ls  -l               --- より詳しいディレクトリ情報を表示
total 3
  -rw-r--r--   1 kousen      st      2874   Mar 2 11:19   cshrc
     ↑            ↑          ↑        ↑          ↑          ↑
    属性     ユーザー名   グループ名 ファイルの   作成時間    ファイル名
                                     大きさ(Byte)

% ls  -al              --- より詳しいディレクトリ情報を隠しファイルも含
                           めて表示



% rm  cshrc            --- ファイルの削除
```

```
% ls                    --- 確認
                       --- ちゃんとなくなっている
% cp  .cshrc  cshrc     --- もう一度コピー
% ls
cshrc
```

2.5 ディレクトリの作成、移動、削除

ディレクトリは棚のようなもので、関連したファイルを同じ棚にまとめて入れておけば、後で必要なファイルを早く探し出すことができます。またカレント(現在の)ディレクトリを work に移動するということは、work という棚の前に立つことに対応します。従って ls すると、work の中に入っているファイルだけが表示されます。

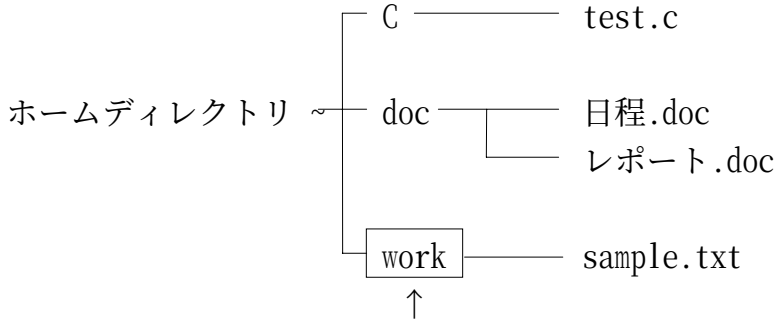

図2 ディレクトリの例

```
% mkdir  work          --- work というディレクトリを作成
% ls  -l               --- 確認
total 4
  -rw-r--r--  1 kousen         2874 Mar  2 11:19 cshrc
  drwxrwxr-x  2 kousen          512 Mar  2 11:20 work
  ↑
```

dは、ディレクトリであることを示しています。

```
% ls  -F               --- 便利なディレクトリ表示形式
cshrc   work/          --- ディレクトリには、最後に / がつく。

% cd  work             --- ディレクトリの移動
% ls  -a               --- 確認（移動したか良くわからない）
.     ..
% pwd                  --- 現在のディレクトリ位置の表示
/usr/home/kousen/work

% cd  ..               --- 親ディレクトリ（もとのディレクトリ）に移動
```

```
% pwd                    --- 確認
/usr/home/kousen

% cp  cshrc  work        --- ファイル（cshrc）を work ディレクトリにコピー
% ls  work               --- work ディレクトリにコピーされていることを確認
cshrc

% cp  cshrc  work/sample  --- ファイル（cshrc）を work ディレクトリに
                             sample という名前でコピー
% ls  work               --- work ディレクトリにコピーされていることを確認
cshrc    sample

% rmdir  work            --- work というディレクトリの削除を試みる
rmdir: work: Directory not empty      --- ディレクトリの中が空でないた
                                          め削除できない
% rm work/*              --- ディレクトリ内のファイルを全て削除
% rmdir  work            --- 再度 work というディレクトリの削除
% ls  -F                --- 確認
cshrc                   --- ちゃんとディレクトリが削除されている

% mkdir  work            --- もう一度 work というディレクトリを作成
% ls  -F                --- もとに戻ったか確認
cshrc    work/
```

2.6 テキストエディタの使用、ファイル内容の表示

```
% aemacs t.c             --- テキストエディタ aemacs を起動
```

[最低限覚えてほしい aemacs の操作]
^X ^C … 強制終了　　ESC-Z … セーブ後終了　　ESC-? … HELP
↑
コントロールキーを押しながらＸを押すことを示しています。

```
% ls                     --- t.c ができたか確認
cshrc   t.c   work

% cat  t.c               --- ファイル内容の表示
#include    <stdio.h>
void main(void){
    printf("---- test ------\n");
}
```

2.7 Ｃのコンパイル

```
% gcc  -o  t  t.c      --- Cのコンパイル
                              (-o t で実行ファイルを t に指定)

% ls  -F               --- 確認(t という実行ファイルができている)
cshrc   t*    t.c     work/
         ↑
```
＊は、実行可能ファイルであることを示している

```
% t                    --- 実行
---- test ------
```

2.8 ファイル名の変更、移動

```
% mv  t  test1         --- ファイル名の変更
  (mv [旧ファイル名] [新ファイル名])

% ls  -F               --- 確認
cshrc    t.c     test1*   work/

% mv  t.c  test1.c
% ls  -F               --- 確認
cshrc    test1.c   test1*    work/

% mv  cshrc  work      --- ファイルの移動
                          (cshrc を work ディレクトリに移動)
% ls  -F               --- 確認
test1.c   test1*    work/

% mv  te*  work        --- ファイルの移動
(te で始まるファイルすべて[test1.c と test1]を work ディレクトリに移動)
% ls  -F               --- 確認
work/
% ls  -F  work         --- work ディレクトリの中にあるファイル名を表示
cshrc    test1*   test1.c
```

2.9 オンラインマニュアルの利用

```
% man  csh             --- csh の説明がでてくる
% man  -k  gnu         --- gnu に関するコマンド等が表示される
```
(注1)もしオンラインマニュアルがうまく表示されないときは **cp ~kane/.cshrc ~** を実行し、再度ログインしなおして下さい。

(注2)漢字が化けている場合は、Tera Term の Setup メニューの中の Terminal を選

択し、Kanji(Recieve)と Kanji(Transmit)を EUC に変更して下さい。

2.10 終了

% **exit**

2.11 整理（MS-DOS コマンドとの比較）

UNIX	MS-DOS	備考
ls	dir	ディレクトリ情報の表示
cp	copy	ファイルのコピー
rm	del	ファイルの削除
mv	ren	ファイル名の変更，移動(UNIX のみ)
cd	cd	ディレクトリの移動
pwd	cd	ディレクトリ位置の表示
mkdir	md	ディレクトリの作成
rmdir	rd	ディレクトリの削除
more	more	一画面ごとにファイル等を表示
exit		終了
^S	^S	一時停止
^Q	any key	一時停止の解除
^C	^C	プロセスの強制終了

2.12 リダイレクト

% **aemacs file1** --- 10 行程度何か入力する
% **aemacs file2** --- 10 行程度何か入力する

% **cat file1** --- file1 の内容を確認

% **cat file1 > rel** --- 標準出力が rel に切り替わり、結果が rel に出力される。このとき以前に rel に何か書き込んであっても以前の内容は消去される。
% **cat rel** --- file1 と同じ内容であることを確認

% **cat file2 > rel** --- 標準出力が rel に切り替わり、結果が rel に出力される。このとき以前に rel に何か書き込んであっても以前の内容は消去される。
% **cat rel** --- file2 と同じ内容であることを確認

% **cat file1 >> rel** --- 標準出力が rel に切り替わり、結果が rel の最後に追加される。

% **cat re1** --- file2+file1 と同じ内容であることを確認

% **cat file2 file1 > re2** --- file2 と file1 の内容が re2 に出力される。
% **cat re2** --- file2+file1 と同じ内容であることを確認

% **cat < file1** --- 標準入力が file1 に切り替わり file1 の内容が表示される

% **cat < file 1 > re1** --- 標準入力が file1 に、標準出力が re1 に切り替わり、file1 の内容が re1 に出力れる。
% **cat re1** --- file1 と同じ内容であることを確認

2.13 フィルターコマンドとパイプ

% **head -5 file1** --- file1 の先頭５行を表示
% **tail +5 file1** --- file1 の５行目から表示
% **tail -3 file1** --- file1 の最後の３行を表示

% **tail -40 .cshrc | more** --- .cshrc の最後の 40 行を１画面ごとに表示

% **ls -l /** --- ルートディレクトリのファイル情報を詳しく表示
% **ls -l / | colrm 1 10** --- ルートディレクトリの詳しいファイル情報の各行について、１から 10 文字目までを削除

このように

% プログラム１| プログラム２

と入力すると、プログラム１の結果（標準出力）がプログラム２の入力（標準入力）に渡されます。
これは、

% プログラム１ > 結果１
% プログラム２ < 結果１

によっても同一の結果が得られます。ここで「<」は、プログラム２の標準入力（通常はキーボード）をファイル（結果 1）に切り替える働きをします。

% **ls -l / > result1**
% **colrm 1 10 < result1**--- **ls -l / | colrm 1 10** と同じ結果が得られる。

［検索］

% **grep scale .cshrc** --- .cshrc 中の scale を含む行を検索

% **grep printf work/*.c** --- work ディレクトリにある c ファイル中の printf を含む行を検索

[ソート]

% **sort .cshrc | more** --- .cshrc の各行をソートし１画面毎に表示

% **sort .cshrc | uniq | more** --- .cshrc の各行をソートし、重複行をなくし、１画面毎に表示

[文字変換]

% **tr a-z A-Z < .cshrc > cshrc.tr** --- .cshrc 中の小文字を大文字に（a-z を A-Z に）変換し、cshrc.tr に結果を入れる

% **more cshrc.tr** --- 結果を確認

% **tr -d '\012' < file1 | more** --- file1 中の改行（8 進で\012)を削除して、１画面毎に表示

[カウント]

% **wc file1** --- file1 中の行数、単語数、および文字数を表示

[相違点を調べる]

% **tr z @ < .cshrc > cshrc.tr** --- .cshrc 中の z を@に変換し結果を cshrc.tr に格納

% **diff .cshrc cshrc.tr** --- .cshrc と cshrc.tr の違いを表示

[文字列の置換]

% **man sed | grep sed > file3** --- sed のオンラインマニュアルの中で sed を含む行のみ file3 に出力

% **sed 's/sed/stream editor/g' file3** --- file3 中の sed を stream editor に置換

sed 's/検索文字列/置換文字列/g'

最後の g をつけないと各行の変換は１回のみとなる。

% **sed 's/^ //g' file3** --- file3 中の行頭の５つの空白を削除して表示 '^'は、行頭であることを表している

% **sed 's/$/> file/g' file3** --- file3 中の行末($)に"> file"を付加

% **sed 's/:.*$//' /etc/passwd** --- /etc/passwd 中の':'以降の文字列を削除

ここで、'.'は任意の１文字を意味し、'*'は直前の文字の繰り返しを意味する。

よって、".*"は、任意の文字列を意味する。また'$'は行末を意味するので、":.*$"は":(任意の文字列)(行末)"すなわち':'以降の文字列を意味する。

[特定フィールドを抜き出す]

% **ypcat passwd | more**--- NIS(Network Information Service)が動作している状態で password 情報を表示

% **ypcat passwd | awk -F: '{print $1, $5}' | more**
--- １番目と５番目のフィールドを表示。-F に続く':'は区切り文字を指定している。

% **ypcat passwd | awk -F: '{printf "%8s\t%s\n", $1, $5}' | more**
--- 見やすく表示

2.14 スクリプト

% **aemacs test.sh** --- editor を起動し以下の内容を test.sh に保存

```
echo ディレクトリ`pwd`には以下のファイルがあります。
ls -F
```

% **qkc -e test.sh** --- test.sh の漢字コードを EUC に変換

% **chmod +x test.sh** --- test.sh を実行可能にする

% **test.sh** --- test.sh を実行

このように一連の操作をファイルに記述したファイルをシェルスクリプトといいます。一度スクリプトを書いておけば、次回からの操作が楽になります。例えば、以下のようなスクリプト cck を書いておけば、cck test と入力するだけで test.c をコンパイルし、実行ファイル名を test にすることができます。

% **aemacs cck** --- editor を起動し以下の内容を cck に保存

```
gcc -o $1 $1.c -lm
```

ここで、$1 はスクリプトに与える１番目の引数(test)です。このシェルスクリプトを使用し、test.c をコンパイルするには **cck test** とすれば良いわけです。

[C-shell]
スクリプトの先頭に
#! /bin/csh
と書いておくと、C-shell が起動されます。 C-shell は、C ライクな構文と高度な会話型機能をもつシェル(コマンド・インタープリタ)です。以下のような if 文、foreach 文、while 文などが使用できます。

```
if ( 条件式１ ) then
    処理１
else if ( 条件式２ ) then
```

```
    処理2
else
    処理3
endif

foreach i ( 1 2 3 a bc def )
    処理
end

while ( 条件式 )
    処理
end
```

条件式：

```
    $変数名 == "string"    --- 変数の内容がstringに一致？
    $変数名 != "string"    --- 変数の内容がstringに一致しない？
    $変数名 =~ "*.c"       --- 変数の内容が"なんとか.c"?
    $変数名 !~ "*.c"       --- 変数の内容が"なんとか.c"でない?
    -e test.c              --- test.cというファイルが存在?
    -r test.c              --- test.c が読み出し可能?
    -w test.c              --- test.c に書き込み可能?
    -x test.c              --- test.c が実行可能?
    -z test.c              --- test.c のファイルサイズが0?
    -d work                --- workというディレクトリが存在?
```

以下にいくつかの例を挙げるが、詳細はオンラインマニュアルを利用して調べること。

（1）変数とif文の練習（if.sh)

```
#! /bin/csh -f
set abc = mojiretsu             # 変数abcに"mojiretsu"を代入

if ( $abc == "mojiretsu" ) then # 変数abcの内容が"mojiretsu"と一致すれば
    echo "一致"                 #   "一致"と表示
else                            # 一致しなければ、
    echo "不一致"               #   "不一致"と表示
endif
```

（2）foreach文の練習（foreach.sh)

```
#! /bin/csh -f
foreach i ( 1 2 3 a bc def )    # ( )内の値を変数iに代入して順次ループ
```

```
    echo $i                         # 変数 i の内容を表示
end
```

（3）C シェル組み込みコマンド"@"と while 文の練習 (while.sh)

```
#! /bin/csh -f
# C シェル組み込みコマンド"@は右辺の計算を行い、
# 計算結果を左辺の変数に代入します。
set no = 1                          # 変数 no を 1 に初期化
set sum = 0                         # 変数 sum を 0 に初期化
while ( $no <= 10 )                 # 変数 no が 10 以下の間、
    @ sum = $sum + $no              #     sum <-- sum + no
    @ no = $no + 1                  #     no <-- no + 1
end
echo 1 から 10 までの和=$sum
```

（4）引数の練習 (arg)

```
#! /bin/csh -f
echo "与えられた全引数：" $argv
echo "引数の数：" $#argv
echo "1 番目の引数：" $argv[1]
echo "2 番目の引数：" $argv[2]
echo "3 番目の引数：" $argv[3]
```

（5）2+3+4+…+10 の計算結果を表示(add)

```
    #! /bin/csh -f
    if ( $#argv != 2 ) then         # もし引数の数が２つでなければ
        echo "使用例："             #     使用例を表示
        echo "add 2 10"
        echo "2+3+4+…+10 の計算結果を表示"
        exit                        #     終了
    endif

    @ no = $argv[1]                 # 第１引数を変数 no に代入
    @ sum = 0                       # 変数 sum を初期化
    while ( $no <= $argv[2] )       # 変数 no が第２引数以下の間、
        @ sum = $sum + $no          #     sum <-- sum + no
        @ no ++                     #     no ++
    end
    echo $argv[1]から$argv[2]までの和=$sum
```

（6）不要ファイルの削除(clean)

```
    #! /bin/csh -f
    echo "----- 以下のファイルを消去します. -----"
    ls  .*.bak  .*~  *.bak  *~  #*
```

```
    echo -n "-----    よろしいですか？(y/n)      -----"
    if ( "$<" == "y" ) then
        rm  .*.bak  .*~   *.bak  *~  #*
        ls  -AFC
    endif
```

（7）ファイル属性の表示(atrib)

```
    #! /bin/csh -f
    if ( $#argv != 1 ) then          # もし引数の数が１つでなければ
        echo "使用例："              #      使用例を表示
        echo "atrib dir_name"
        echo "dir_name 中のファイルの属性を表示"
        exit                         # 終了
    endif

    echo $argv[1]"ディレクトリ中の"
    foreach fn ( `ls $argv[1]` )     # 第１引数で指定したディレクトリ内のすべてのファイルに対してループ
        echo -n $fn"は"
        if ( -r $argv[1]/$fn ) echo -n "、読み込み可能"
        if ( -w $argv[1]/$fn ) echo -n "、書き込み可能"
        if ( -x $argv[1]/$fn ) echo -n "、実行可能"
        if ( -d $argv[1]/$fn ) echo -n "、ディレクトリ"
        echo "です。"
    end
```

2.15 ディスク容量のチェック

```
    % df                 --- ディスクの空き容量を表示
    % du                 --- カレントディレクトリ以下のディスク容量を表示
```

2.16 プロセス

```
    % ps  ux             --- 現在どのようなプロセスが動いているか表示
    % aemacs  test
```

F10(ファンクションキー）を押す。

```
    % ps  ux
USER       PID %CPU %MEM   SZ  RSS TT STAT START  TIME COMMAND
kousen    7945  0.0  0.6  300  432 p1 S    08:55   0:01 -tcsh (tcsh)
kousen   13755  0.0  0.8  204  576 p1 T    14:23   0:00 aemacs test
kousen   13756  0.0  0.7  252  500 p1 R    14:23   0:00 ps ux

    % kill  -9  13755      --- aemacs のプロセス（プロセス番号 PID 13755)を強制終了させる
```

% **ps ux** --- aemacs のプロセスがなくなったことを確認

３．実験

(課題１)

２節の説明に従い演習し、UNIX 基本操作を習得せよ。

(課題２)

以下の問いに答えよ。

(1) パスワードを変更するには どのような操作をすればよいか。ただし NIS が動作しているものとする。

(2) カレントディレクトリのファイル名を隠しファイルも含めて詳しく表示するにはどの様なコマンドを入力すればよいか記せ。

(3) ./work ディレクトリにある test という名前のファイルを削除するにはどの様なコマンドを入力すればよいか記せ。

(4) ./work ディレクトリにある test という名前のファイルをカレントディレクトリに test2 という名前のファイルとして コピーにはどの様なコマンドを入力すればよいか記せ。

(5) カレントディレクトリに work_dir という名前のディレクトリを作成するにはどの様なコマンドを入力すればよいか記せ。

(6) カレントディレクトリにある work_dir という名前のディレクトリにカレントディレクトリを移動するにはどの様なコマンドを入力すればよいか記せ。

(7) カレントディレクトリの位置を確認するにはどの様なコマンドを入力すればよいか記せ。

(8) カレントディレクトリにある work という名前のディレクトリを削除するにはどの様なコマンドを入力すればよいか記せ。

(9) カレントディレクトリにある work という名前のディレクトリにある test.c という名前のファイルの内容を表示するにはどの様なコマンドを入力すればよいか記せ。

(10) カレントディレクトリにある test.c という名前のＣのソースファイルをコン

パイルし, test という名前の実行ファイルを作成するにはどの様なコマンドを入力すればよいか記せ。

(11) カレントディレクトリにあるファイルで, ファイル名の最後の文字が d であるファイル全てをカレントディレクトリにある work という名前のディレクトリに移動するにはどの様なコマンドを入力すればよいか記せ。

(12) C の関数で、printf()に関する説明を表示させるにはどの様なコマンドを入力すればよいか記せ。

(13) パソコンからワークステーション（ホスト名 ikura）にログインしていました。本来ならログアウトしてからパソコンの電源を切るべきところを誤ってログアウトせずにパソコンの電源を切ってしまいました。正常な状態に戻すにはどのような操作を行えばよいか、記せ。

(14) ファイルシステムの残り容量を表示するにはどの様なコマンドを入力すればよいか記せ。

(15) カレントディレクトリ以下のファイル全体の大きさを調べるにはどのようなコマンドを入力すればよいか記せ。

(16) file1 と file2 を結合して、file3 を作りたい。どうすればよいか。

(17) file1 と file2 の違いを調べたい。どうすればよいか。

(18) カレントディレクトリにある file3 名前のファイルの先頭５行を表示するには、どのようなコマンドを入力すればよいか記せ。

(19) カレントディレクトリにある file3 名前のファイルの最終５行を表示するには、どのようなコマンドを入力すればよいか記せ。

(20) カレントディレクトリにある file3 の中から print という文字列を含む行を表示したい。どのようなコマンドを入力すればよいか記せ。

(21) カレントディレクトリにある file3 という名前のファイル中の小文字の半角英字を大文字に変換し、結果を out という名前のファイルに出力するには、どの様なコマンドを入力すればよいか記せ。

(22) シェルスクリプトを用いて、shell.tst infile と入力すると、infile という名

前のファイル中に含まれる単語の一覧表が作られるようにするにはどうすればよいか。

(23) n の階乗を求めるシェルスクリプトを作成せよ。

４．考察

（１）init の働きについて説明せよ。
（２）inetd の働きについて説明せよ。
（３）新しいユーザーアカウントを作成するにはどのようにすればよいか説明せよ。
（４）新しいホスト名を登録するにはどのようにすればよいか説明せよ。
（５）以下の略語等について、詳しく説明せよ。

NIS	DNS	IP アドレス	ゲートウェイ IP アドレス
プロキシー	ファイアウォール	SMTP	NNTP
RFC	PPP	DHCP	WINS