Assignment 2: Creating bulk user accounts in Linux and Windows — Solutions

Submission: by 8pm, Sunday, 30 November 2003

Where: Online at `http://nicku.org/perl2/submit.cgi`. A paper submission is not required (save the trees!)

Do not submit an RAR archive. I will just delete it; you will get zero marks.

Put your *email address* in a comment right near the top of your assignment so that I can return the marked assignment to you. Assignments with no email address will not be returned.

Important: All programs you write must start with these lines:

```
#! /usr/bin/perl

use warnings;
use strict;
```

and should compile without errors.

Cheating: Your work *must* be original. Copying will be *severely* dealt with.

1 Background

1.1 On Windows

On the Windows system, you may call the external program `net user`, which can be called with syntax like the following:

```
net user user_id plain_text_password /add /fullname:"full user name"
```

You would probably also need to call the `cacls` program to set the permissions.

Alternatively, you could use the program in the lab worksheet: `http://nicku.org/snm/lab/win32-accounts/win32-accounts.pdf`. This uses Perl modules, and is based on an extract from *Perl for System Administration*, by David N. Blank-Edelman, O'Reilly, 2000.

1.2 On Linux in the College

Because we use a network directory for user accounts here in the College, you will find that creating accounts will take too long unless you do the following:

- Create a base directory for local accounts, as described in the document `http://nicku.org/snm/lab/setting-up-local-accounts/setting-up-local-accounts.pdf` and change `useradd` to use this as the default for new home directories:

  ```
  $ sudo mkdir /home2
  $ sudo /usr/sbin/useradd -D -b /home2
  ```

  Note that you only need to ever do this once.

- Boot the computer into *single user mode* (See section 4.7.1 on page 14 of the document `http://nicku.org/ossi/lab/grub/grub.pdf` for how to do this.) Then run your program. Alternatively, temporarily turn off LDAP authentication, as described in the document mentioned above about setting up local accounts.

1.3 On both Linux and Windows

There is a built-in Perl variable that indicates the operating system that the program is running on. The variable `$^O` holds the value `MSWin32` on a Windows system.

How do you use the `Win32::Lanman` and `Win32::Perms` modules presented in `http://nicku.org/snm/lab/win32-accounts/win32-accounts.pdf`? The idea is to test the operating system, and then to `require` each module if the OS is Win32. See the code in the book *Perl for System Administration*, by David N. Blank-Edelman, which you can download from the O'Reilly web site, or view it here: `http://nicku.org/ftp/perl/example-code-and-oreilly-books/PSA-code/PSA-Chapter02.txt`.

What I have done is put all my windows subroutines into a separate file (I called the file `windows_routines.pm`). Note that your file name *must* end in "`.pm`"

The lines

```
require Win32::Lanman;
require Win32::Perms;
```

appear at the top of that file, and do not appear in the main file.

The last line of the file is:

```
return 1;
```

Then in my main program file, I have the following line:

```
if ( $^O eq "MSWin32" ) {
    require windows_routines;
}
```

To allow Perl to find this file, you can either put it in a directory in @INC:

```
$ perl -e 'print join "\n", @INC'
/usr/lib/perl5/5.8.1/i386-linux-thread-multi
/usr/lib/perl5/5.8.1
/usr/lib/perl5/site_perl/5.8.1/i386-linux-thread-multi
/usr/lib/perl5/site_perl/5.8.1
...
```

Or better still, put this at the top of your main file:

```
use FindBin qw($RealBin);
use lib "$RealBin/";
```

See `perldoc FindBin`.

You may wish to use the standard `Getopt::Long` Perl module as a way to pass options to your software. See `perldoc Getopt::Long`.

Assignment Requirements

1. Write a program that can read the data provided at `http://nicku.org/snm/lab/regular-expressions/artificial-student-data-2003.txt`, and generate an account for each student on both Linux and also on Windows.

2. The program should also provide an option to delete the accounts, which works on both operating systems.

3. There should be an option to remove the home directories when deleting accounts. If this option is not specified, the home directories should remain unchanged.

4. The same program must run on both operating systems with no change.

5. The username should be the student number of the student, with the first letter of their family name prepended in lower case.

> For example, for a student with the name "YEUNG, Hoi Man," and the student number 915367894, the user name that the person logs in with is `y915367894`.

6. The password shall be their Hong Kong ID, with letters in *lower* case, but `not` including the parentheses or the check digit. So for a student with the Kong Kong ID A456789(A), their password should be `a456789`. This means that there will be seven characters in each password.

7. You should create a home directory for each user, with appropriate access permissions.

8. You should include the full name of each student in their account information stored in both the Linux and the Windows computer systems.

9. More marks will be awarded for solutions that are more general (i.e., can easily be adapted to use other input data, can provide network account options, can provide more information into each account...)

10. Also, with the soft copy, provide a soft copy of the following from your Linux system:

- password file `/etc/passwd`, and also
- `/etc/shadow` from your Linux system after creating the accounts. *Include only* the lines for the accounts you created with your assignment; do not provide the information about other accounts on your system.
- Include a long listing of the home directories:

```
$ ls -l /home > ~/assignment-2-perl/home-directories-list.txt
```

11. From your Windows system, provide evidence that the accounts have all been successfully created on Windows. At least provide the following:

- The accounts should be shown as output from `net user`, and redirected into a file, to be included as a soft copy with your assignment.
- Use the `cacls` program to show the permissions on the home directories you have created, by redirecting the output of `cacls` into a file, perhaps like this:

  ```
  D:\home> for /d %i in (*) do @cacls %i >> \temp\cacls-out.log
  ```

  Then include that file in the zip file you submit.

  There is no need to include a screen dump from the User Manager for Windows.
- Include a listing of the home directories:

  ```
  D:\home> dir E:\home > assignment-2-perl\home-directories-list-windows.txt
  ```

12. Your program should display an informative error message for each account that fails to be created or deleted.

13. You should be prepared to demonstrate your program in the laboratory.