



## Using Regular Expressions to Create User Accounts

For each exercise where you write a program, keep the original program from each exercise and modify a copy for the next exercise.

### 1 Background

There are many ways of creating accounts on a Linux system; your program could select the next available user and group IDs, manually edit the `/etc/passwd`, `/etc/group` and `/etc/shadow` files, create the user directories, copy the login scripts and other basic account files from `/etc/skel`, and change the ownership to the new account, but a simpler and more portable way is to use the standard `useradd` program that you learned about last year. We can call that from our own Perl programs using the built-in `system` function.

Today we will write a program that will create a user account for each student listed in the artificial student registration data.

#### Procedure:

1. Download the artificial student data from <http://nicku.org/snm/lab/regular-expressions/artificial-student-data.txt>. This file is in the old format of the student registration system, but contains no real data about any student.
2. Write a Perl program that can read all the lines of this file when it is given as a command line parameter, and print the student numbers only, one to each line.
3. Make a copy of your program, and modify it so that it prints only the names of the students, one to each line, with no extra spaces either at the beginning or end of each name.
4. Using the manual page for the `useradd` program as a guide, modify your previous programs so that your program prints one `useradd` command for each student, using the student ID as the login ID, the Hong Kong ID as the password, and the name of the student as a comment.

Note that our Linux systems require that the user/group names must start with a letter, and may not contain colons, commas, newlines or any non-printable characters. The maximum length of the user ID is 32 characters; that of the group ID is 16. To meet this requirement, make the user ID be the first letter (made lower case) of the family name, followed by the student ID.

5. Modify this last program further so that it uses the built-in Perl function `system` to execute the `useradd` command as well as print the command. Type `perldoc -f system` to read about this important function.
6. Modify your last program so that it reports an error message if the execution of any `useradd` command is unsuccessful.