Programming Tools

http://www.cs.chalmers.se/~kemp/teaching/programming_tools/

Aims

To introduce a variety of programming tools on a technical level.

To demonstrate how simple data processing and presentation tasks can be achieved using these tools on their own, or in combination.

To introduce some common application programs.

Objectives

At the end of this course, students should:

— understand some basic concepts of the UNIX operating system, Java and Perl;

— be able to perform a variety of data processing and presentation tasks;

— know how to use some common application programs.

What is an Operating System?

An operating system is the program that controls all the other parts of a computer system — both the hardware and the software. Most importantly, it allows you to make use of the facilities provided by the system. Every computer has an operating system.

The UNIX operating system has three important features; a kernel, the shell and a filesystem.

Amongst the functions performed by the kernel are:

- managing the machine's memory and allocating it to each process.
- scheduling the work done by the CPU so that the work of each user is carried out as efficiently as is possible.
- organising the transfer of data from one part of the machine to another.
- accepting instructions from the shell and carrying them out.
- enforcing the access permissions that are in force on the file system.

Programming Tools

Course structure

UNIX, etc.	4 lectures	Graham Kemp
Perl, etc.	4 lectures	Graham Kemp
Java, etc.	3 lectures	Daniel Dalevi

Exercises

Course exercises will give you an opportunity to develop your skills in using the programming tools introduced in the lectures.

Assessment

This course will be assessed through a series of **individual assignments**.

Shells

The shell is a program that interprets your commands, passes them on to the kernel and then displays the result of this operation.

Several different shells are available for UNIX:

tcsh	C shell with file name completion and command line editing
sh	standard shell and command interpreter (Bourne shell)
csh	shell command interpreter with a C-like syntax
bash	GNU Bourne-Again SHell
ksh	KornShell, a standard/restricted command and programming language

UNIX file system

Ordinary files
— used to store information (e.g. text, data, images, etc.)

Directories
— a file that holds other files and other directories

The UNIX file system is organised as a hierarchy of directories starting from a single directory called root which is represented by a / (slash).

Home directory (~) and current directory (.)

Pathnames
— absolute (start with /)
— relative

Some UNIX commands (1)

exit	- exit from the shell
passwd	- change login password
date	- write the date and time
whoami	- display the effective current username
who	- who is on the system
ls	- list contents of directory
cat	- concatenate and display files
more	- browse through a text file
pwd	- return working directory name
cd	- change working directory
mv	- move file or directory
cp	- copy file
rm	- remove file
mkdir	- make directory
rmdir	- remove directory
man	- find and display reference manual pages

UNIX system directories

/bin	executable binary files for some commands and utilities
/dev	special files used to represent real physical devices such as printers and terminals
/etc	commands and files used for system administration
/home	contains a home directory for each user of the system (/users at Chalmers)
/lib	libraries used by various programs and languages
/tmp	a "scratch" area where any user can store files on a temporary basis
/usr	system files and directories that you share with other users

Some UNIX commands (2)

printenv	- display values of environment variables
setenv	- assign value to environment variable
which	- locate a command; display its pathname or alias
alias	- create a pseudonym for a command
history	- process command history list
source	- read and execute commands from a file
quota	- display a user's file system disk quota and usage
du	- summarise disk usage
compress	- compress files
uncompress	- uncompress files
zcat	- display compressed text
zmore	- browse compressed text
wc	- count lines, words and characters in a file
head	- display first few lines of files
tail	- display the last few lines of a file
grep	- search file(s) for a pattern

Entering UNIX commands

Type its name followed by any options and arguments.

Options modify the way that a command works. They usually consist of a hyphen followed by a single letter.

Redirecting standard input and output

<	redirect standard input so that it comes from a file
>	redirect standard output so that it goes to a file
>>	append the standard output from a command to a file

Connecting commands with pipes

e.g. command1 | command2 | command3

Using regular expressions with the grep command

c	any non-special character represents itself
\c	turns off the meaning of any special character
^	beginning of a line
$	end of a line
.	matches any single character except a newline
[abc]	matches any of the enclosed characters
[^abc]	matches any character that is not enclosed
[a-m]	matches any character in this range
*	matches any number of the preceding character

Always quote the regular expression. This prevents the shell from interpreting the special characters before it is passed to the grep command.

Matching file names with regular expressions

You can use the following metacharacters within any shell to create regular expressions that match file names.

?	matches any single character
*	matches any number of any characters
[abc]	matches any of the enclosed characters
[a-m]	matches any character in this range

```
unix> rm temp*
unix> uncompress *.Z
unix> ls -l .??*
unix> ls -l */*
```


Editing streams with sed

```
[ address [ , address ] ] command [ arguments ]
```

address:

- a line number
- $
- /regular experssion/

common commands:

- d
- s/regular expression/replacement/flags