Lab 2

- ## Executing shell commands in Perl:

Backtick Operator:

```
#!/usr/bin/perl

@files = `ls -l`;

foreach $file (@files) {
   print $file;
}
```

System Function:

```
system( "ls -l")
```

- ## Multiprocessing in Perl:

The fork function:

The fork() system call creates a new process, called a child process. What it effectively does is that it creates a clone of the current process. The fork() function returns the child pid to the parent process, 0 to the child process, or undef if the fork is unsuccessful.

The exec function:

The exec() system call is used to assign a different code/program to execute for a child process.

```
if(!defined($pid = fork())) {
  # fork returned undef, so unsuccessful
  die "Cannot fork a child: $!";
}
elsif ($pid == 0) {
  print "Printed by child process\n";
  exec("date") || die "can't exec date: $!";
}
else {
  # fork returned 0 nor undef
  # so this branch is parent
  print "Printed by parent process\n";
  while (wait() != -1) {}
  print "Completed all child processes\n";
}
```

- ## Version Control (Git) :

  Version control is a system that records changes to a file or set of files over time so that you can recall specific **versions** later.

  ### Basic Git Commands

  To initiate a new git repository in a directory:
  git init

  To add files to staging:
  git add filename

  To check status of repository:
  git status

  To commit changes:
  git commit -m "Comment"

  To create a new branch:
  git branch branch_name

  To switch to a branch:
  git checkout branch_name

  To merge a branch into current branch:
  git merge branch_to_merge

To push to a remote repository:

```
git remote add name_of_address link_to_git_repository
git push name_of_address name_of_branch
```

To pull changes from remote repository:

```
git pull name_of_branch name_of_address
```

To clone a repository:

```
git clone remote_url
```

To check commit history:

```
git log
```

or

```
git log --oneline
```

To switch to a previous commit, fIrst, check the hash of commit from git log, then:

```
git checkout hash_of_commit
```

or

```
git checkout ref_of_commit
```