Processes, Services and Apache – Overview Sheet

Processes

A process is simply a runing program, each process has a unique process-id (pid) and is also running under a user and group which give it permissions to access certain parts of the system. Processes can be killed manually and monitored using the command line outside of the programs execution flow.

Services and Daemons

A service is a process which provides background resources via some form of bus or via the network. A service will listen on network ports or api buses (i.e. dbus) and respond to queries from client programs. A service is usually always running and can provide a diverse array of features from printing queue management to web page services.

A daemon is a background process and can be said to be another word for a service program.

HTTP Web services

Web site services are provided by web servers, these services listen to port 80 by default and supply files (normally html) over the network using tcp with http headers followed by the content. An HTTP header is plain text, name value pairs which describe the following content, type, size, date requested, caching details and so on. The HTTP content can be binary or text based and should be of the type specified in the header.

There are different kinds of HTTP server from which to pick from but for our practical session we will be using the apache2 web server.

Services Monitoring

You can monitor a process using the `ps` and `top` commands, ps is a tool for listing all processes and has many options to change the output view and selection of shown processes, the given example shows as much information and all processes running on the system, but you should get familiar with the manual for the ps command. The top command is an interactive running program, which refreshes constantly a list of top resource consuming processes, as well as giving a good overview of memory, processor and io useage on the system.

```
ps aux   # Show me all running processes, from any user
top      # Interactive display of resource use
```

Lots of service daemons will save a pid lock files in /var/, this file contains the process-id for the running process and can be used to kill or check if the process is running.

Most daemons will create logs in /var/log/, these logs are constantly updated with new information and previous days logs will be separated out and after a time automatically deleted. Keeping the logs current. To view these logs you can use the `cat`, `head` and `tail` commands:

```
cat /var/log/dmesg     # print entire log file to screen
head /var/log/dmesg    # Show the first few lines of the file
tail -f /var/log/dmesg # Show the last few lines and follow changes
```

Controlling Services

The most direct way of controlling a service is using the init.d controls, each service will have a script in /etc/init.d which can be executed with a start/stop/restart argument. These scripts control the process id directly and can place a pid file in /var/ to limit the number of processes to one.

```
sudo /etc/init.d/apache2 start
sudo /etc/init.d/apache2 stop
sudo /etc/init.d/apache2 status
```

You can run from the command line any daemon service that you normally run in the background, this can give you more error information if things are going wrong. For instance with apache2 we can run `apache2`, although this is not the recommended way of running services normally, since the processes will be killed as soon as you exit the shell command line.

To kill or stop a process manually without using the init.d script, you can use `kill` and `killall` commands:

```
kill [process-id]
killall [process-name]
```

Apache2 Web Service

The apache service has a set of configuration files in /etc/apache2/ which should be configured for optimal performance and security. These configurations allow you to enable and disable each individual website as well as all the available features (plugins) to apache. It's also important to make sure the apache service is running by checking the status from the init.d script as shown above.

Basic Configuration

Each website is a virtual host which allows a single server to host multiple websites and it's configuration file should specify the document root (where files can be found), the server admin (email address), logging and the access options for each directory or sub-directory in the document root. (see example configuration provided)

Each website should have a configuration file in /etc/apache2/sites-available/, then when this website is enabled it should be symbolically linked (using ln -s) to /etc/apache2/sites-enabled/ and the apache2 process restarted for the new website to become available.

```
cd /etc/apache2
gedit sites-available/mysite.conf
ln -s sites-available/mysite.conf sites-enabled/10-mysite.conf
sudo /etc/init.d/apache2 restart
```

Viewing the Result

To view the resulting website, you only have to open up your favourite web browser such as Firefox or Lynx and browse to the hostname or ip-address of your computer:

Address: http://192.168.10.60/
Address: http://myhostname/

Extensions and Sites Management

Apache has a method of increasing it's functionality using plugins. These can be enabled or disabled using the supplied tools, look in the /etc/apache2/mods-available/ directory for a list of available plugins. Using the same method above for symbolically linking the website configuration, link each of the modules you want to enable to /etc/apache2/mods-enabled/. Make sure to link both the .load file and the .conf file for any given module plugin you want to enable.

```
cd /etc/apache2
ln -s mods-available/alias.conf mods-enabled/
ln -s mods-available/alias.load mods-enabled/
sudo /etc/init.d/apache2 restart
```