CS214 - Advanced UNIX

Lecture 5
Processes. Format strings. Crontab

Ymir Vigfusson

Processes

A process is a program that is being executed.

- Processes can *fork* to create new processes.
- The new process is called a *child* and the old one is its *parent*.
- Gives a hierarchical structure of programs being executed concurrently (*multi-tasking*).
- Each process has a *process identifier* (PID) associated with it.

Processes

Sometimes we wish to change the state of a resident process.

- This is done by sending a *signal* to that process.
- The bash command to do so is called `kill`.
- There are quite a few different signals, see `kill -l` for a list.
- Most commonly used are HUP, INT, KILL, STOP, CONT and TERM.

Processes

- Processes have handlers for the signals they support, except for KILL which cannot be blocked (and will terminate the process).
- When we send a signal (and we have permission to do so), the handler code gets executed.
- For example, HUP is often used for making processes reread a configuration file, or a database.
- TERM is used for making processes terminate gracefully.

Processes

- STOP pauses a process (cannot be blocked), and CONT will resume it.
- When you process ^Z (ctrl+Z) in bash, it will send TSTP to pause the foreground process. Similar to STOP, but can be blocked.
- Example: `kill -KILL 5312` will terminate process with PID 5312 not-so-gracefully.
- Use `ps` to get a list of processes, and `ps auxef` will give you more than enough information about what's running on your system.

Processes

- If you wish to send a signal to a bunch of processes, use `killall` and specify the name instead of the PID.
- Example: `killall -HUP mysqld` will send SIGHUP to all programs whose name includes mysqld.
- You can actually use regular expressions by using `-r`.
- To find the PID of a running program in a simple way, use e.g. `pidof mysqld`.

Processes

- In bash, new processes go in foreground per default.
- You can put them in background by using & in the end of the line.
- Example: `find / -name "cs214*" &` will be run in the background.

Processes

- You can list background processes by using `jobs`.
- To resume a stopped process in foreground type e.g. `fg 3` where 3 is the identifier you saw in `jobs`.
- To resume it in the background, use `bg`.

Processes

- Example: To produce a list of files ending in `.tex` in the background and redirecting all errors to *the black hole* `/dev/null`, use

```
find / -name "*.tex" 1> texf 2> /dev/null &
```

- You can join the error messages and the regular output by using `2>&1`.
- Example: `./script > output.txt 2>&1`.

Format strings

The `printf` function from C exists in bash and gawk, among other places.

- `printf "Test"` prints `Test` but without any newline.
- `printf "Test\n"` prints `Test` in the a separate line.

Format strings

The `%` is a special directive that tells `printf` to convert the next argument into a modified string.

- `printf "%d, %d and %x\n" 1 2 255` prints `1, 2 and ff`.

Format strings

This is especially convenient for reformatting floating point numbers.

- `printf "%06.3f%%\n" 5.66666` prints `05.667%`.
- The ’0’ means to pad with zeroes, ’6’ is the minimum width of the string that is output, 3 is the number of digits after the period, `f` specifies a floating point number, and %% prints a literal %.

Format strings

- The directive `%s` prints a string, and `%.20` will print at most 20 characters of the string.
- Use `%x` and `%X` for lower-case and upper-case hexadecimal.
- Example: `%#02X` with argument 15 will print `0x0f`.

Crontab

Crontab is a service that allows us to schedule scripts to be run periodically.

- A privileged cron *daemon* (or a resident program) is responsible for scheduling the requests, security, and sending the output to the user.
- Use `crontab -e` to edit your user's cronjobs.

Crontab

Here is an example of an entry in crontab.

```
# Run every 2 hours on MTWF.
0 */2 * * Mon-Wed,Fri   echo "Feed the cat" |mai
```

- The format is *minute, hour, day of month, month, day of week*.
- Each field allows for list and ranges.
- The `*` is the entire range, `/` is a step value (i.e. `*/2` could be every two hours).