CMPUT 379
Lab 7: Daemon Processes

Chris Solinas
solinas@ualberta.ca
Based on slides by Arnamoy Bhattacharyya (arnamoy@ualberta.ca)

What is a daemon?

What is a daemon?

- Program that runs in the background

What is a daemon?

- Program that runs in the background
- (normally) no user interaction

What is a daemon?

- Program that runs in the background
- (normally) no user interaction
- Some Examples:
  - sshd
  - crond

How to write a Daemon

1. Fork a child process and exit parent
2. Unmask any inherited file modes
3. Open logs
4. Create new session id
5. Change working directory
6. Close standard file descriptors

How to write a Daemon

1. Fork a child process and exit parent

How to write a Daemon

1. Fork a child process and exit parent

```
pid = fork();
if (pid < 0){
     printf("fork failed!\n");
     exit(1);
}

if (pid > 0){
     printf("pid of child process %d
\n", pid);
     exit(0);
}


/* Child process execution */
```

How to write a Daemon

2. Unmask any inherited file modes

- Clone of parent (including file permissions)
- Should open, read, write to files we need to

How to write a Daemon

2. Unmask any inherited file modes

```
umask(0);
```

- Clone of parent (including file permissions)
- Should open, read, write to files we need to

How to write a Daemon

3. Open logs

- Can’t rely on users watching stdout
- Closing stdout anyway
- Syslog is often the best choice but not for assignment 2

How to write a Daemon

3. Open logs

```
fp = fopen ("mylog.log", "w+");
```

- Can’t rely on users watching stdout
- Closing stdout anyway
- Syslog is often the best choice but not for assignment 2

How to write a Daemon

4. Create new session id

- Parent forked, then exited without waiting
- Child process looks like an orphan
- Create our own process group

How to write a Daemon

4. Create new session id

- Parent forked, then exited without waiting
- Child process looks like an orphan
- Create our own process group

```
pid_t sid;

sid = setsid();
if (sid < 0) {
   fprintf(fp, “Could not create process
group”);
   exit(EXIT_FAILURE);
}
```

How to write a Daemon

5. Change working directory

- Working directory inherited from parent could get unmounted

How to write a Daemon

5. Change working directory

- Working directory inherited from parent could get unmounted

```
if ((chdir("/")) < 0) {
    fprintf(fp, “Could not change working
directory);
    exit(EXIT_FAILURE);
}
```

How to write a Daemon

6. Close standard file descriptors

- STDOUT, STDIN, STDERR
- Not useful without user interaction

How to write a Daemon

6. Close standard file descriptors

- STDOUT, STDIN, STDERR
- Not useful without user interaction

```
close(STDIN_FILENO);
close(STDOUT_FILENO);
close(STDERR_FILENO);
```

How to write a Daemon

That’s it!

```
// do something useful
```

Demo

Another approach

daemon(3)

- Standardized function for creating daemon processes
- **Implemented using concepts seen earlier**
- See man page for more

```
#include <unistd.h>

daemon(0, 0)
```

Important

- Do not leave daemonized processes running on the lab machines when you log out
- Sysadmins **WILL** find out and you may get docked marks