**Operating Sys Lab 1**

**Problem 1.**

1. The child process is able to kill it’s parent process. When this happens, it is turned into an orphan process. One the process is orphaned it is immediately adopted by the init process.

2. A zombie process or defunct process is a process that has completed execution but still has an entry in the process table. It can not kill it’s parent process as a zombie process has already completed its execution.

3.Any process can become a zombie if it has not be cleaned(or reaped) properly and has not been taken of the process table. Therefore yes, a daemon can become a zombie.

4. An orphan can run as long as it needs to after being orphaned. This is due to it being immediately re-parented and this is a common way of creating daemon processes.

**Problem 2.1**

**#include <stdio.h>**

**#include <time.h>**

**#include <unistd.h>**

**int** **main**(**void**)

{

**while**(1**==**1)

{

**time\_t** current\_time;

**char\*** c\_time\_string;

current\_time **=** time(NULL);

c\_time\_string **=** ctime(**&**current\_time);

(**void**) printf("Current time is %s", c\_time\_string);

**FILE** **\***file;

file **=** fopen("/tmp/lablog.txt","a+");

**if** (file **!=** NULL)

{

fprintf(file,"%s",c\_time\_string);

fclose(file);

}

sleep(10);

}

}

**Problem 2.2**

**#include <sys/types.h>**

**#include <sys/stat.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <fcntl.h>**

**#include <errno.h>**

**#include <unistd.h>**

**#include <syslog.h>**

**#include <string.h>**

**#include <time.h>**

**int** **main**(**void**) {

*/\* Our process ID and Session ID \*/*

**pid\_t** pid, sid;

*/\* Fork off the parent process \*/*

pid **=** fork();

**if** (pid **<** 0) {

exit(EXIT\_FAILURE);

}

*/\* If we got a good PID, then*

*we can exit the parent process. \*/*

**if** (pid **>** 0) {

exit(EXIT\_SUCCESS);

}

*/\* Change the file mode mask \*/*

umask(0);

*/\* Open any logs here \*/*

*/\* Create a new SID for the child process \*/*

sid **=** setsid();

**if** (sid **<** 0) {

*/\* Log the failure \*/*

exit(EXIT\_FAILURE);

}

*/\* Change the current working directory \*/*

**if** ((chdir("/")) **<** 0) {

*/\* Log the failure \*/*

exit(EXIT\_FAILURE);

}

*/\* Close out the standard file descriptors \*/*

close(STDIN\_FILENO);

close(STDOUT\_FILENO);

close(STDERR\_FILENO);

*/\* Daemon-specific initialization goes here \*/*

*/\* The Big Loop \*/*

**while**(1**==**1)

{

**time\_t** current\_time;

**char\*** c\_time\_string;

current\_time **=** time(NULL);

c\_time\_string **=** ctime(**&**current\_time);

(**void**) printf("Current time is %s", c\_time\_string);

**FILE** **\***file;

file **=** fopen("/tmp/lablog22.txt","a+");

**if** (file **!=** NULL)

{

fprintf(file,"%s",c\_time\_string);

fclose(file);

}

sleep(10);

}

exit(EXIT\_SUCCESS);

}

Problem 3.1