1. *Using Fork() and Exec() or Clone(), create four child processes. Load the Hello” program in each process after creation (same program). Each child program should print its own PID or some other parameters distinguishing its execution.*
2. *Each process should run and return to the parent process where the parent is waiting to terminate. The parent should only terminate after all children returned from execution.*
   1. **child\_create.c**

#include <sys/types.h>

#include <stdio.h>

#include <unistd.h>

int main()

{

printf("Parent PID: %d\n", getpid());

pid\_t pid;

int childNum;

for(childNum = 1; childNum < 5; ++childNum)

{

pid = fork();

if (pid < 0) {

fprintf(stderr, "Fork Failed");

return 1;

}

else if (pid == 0) {

printf("Child %d PID: %d\n", childNum, getpid());

execlp("/home/mjs/Homework3/hello", "hello", NULL);

}

else {

wait(NULL);

printf("Child %d Complete\n", childNum);

}

}

printf("All children complete\n");

return 0;

}

* 1. **output**

Parent PID: 1356

Child 1 PID: 1357

File file.txt opened and reads as follows:

Hello World!

Writing to file: Hello to you too!

Child 1 Complete

Child 2 PID: 1358

File file.txt opened and reads as follows:

Hello World!

Hello to you too!

Writing to file: Hello to you too!

Child 2 Complete

Child 3 PID: 1359

File file.txt opened and reads as follows:

Hello World!

Hello to you too!

Hello to you too!

Writing to file: Hello to you too!

Child 3 Complete

Child 4 PID: 1360

File file.txt opened and reads as follows:

Hello World!

Hello to you too!

Hello to you too!

Hello to you too!

Writing to file: Hello to you too!

Child 4 Complete

All children complete

* 1. **hello.c**

#include <stdio.h>

#include <stdlib.h>

int main ()

{

// Open file "file.txt"

char file\_name[] = "file.txt";

FILE \*fp = fopen(file\_name, "r+a");

// Check for error opening file

if(fp == NULL)

{

perror("Error opening file.\n");

exit(EXIT\_FAILURE);

}

// Print file contents

printf("File %s opened and reads as follows: \n", file\_name);

char \*line = NULL;

size\_t length = 0;

ssize\_t read;

while((read = getline(&line, &length, fp)) != -1)

{

printf("%s", line);

}

// Write to file

char message[] = "Hello to you too!\n";

printf("Writing to file: %s", message);

fprintf(fp, message);

// Close file

fclose(fp);

// Cleanup

if(line != NULL)

free(line);

// Return

return 0;

}

* 1. **file.txt before run**

Hello World!

* 1. **file.txt after run**

Hello World!

Hello to you too!

Hello to you too!

Hello to you too!

Hello to you too!