// C program to demonstrate use of fork() and pipe()

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/types.h>

#include<string.h>

#include<sys/wait.h>

int main()

{

// We use two pipes

// First pipe to send input string from parent

// Second pipe to send concatenated string from child

int fd1[2]; // Used to store two ends of first pipe

int fd2[2]; // Used to store two ends of second pipe

char fixed\_str[] = "forgeeks.org";

char input\_str[100];

pid\_t p;

if (pipe(fd1)==-1)

{

fprintf(stderr, "Pipe Failed" );

return 1;

}

if (pipe(fd2)==-1)

{

fprintf(stderr, "Pipe Failed" );

return 1;

}

scanf("%s", input\_str);

p = fork();

if (p < 0)

{

fprintf(stderr, "fork Failed" );

return 1;

}

// Parent process

else if (p > 0)

{

char concat\_str[100];

close(fd1[0]); // Close reading end of first pipe

// Write input string and close writing end of first

// pipe.

write(fd1[1], input\_str, strlen(input\_str)+1);

close(fd1[1]);

// Wait for child to send a string

wait(NULL);

close(fd2[1]); // Close writing end of second pipe

// Read string from child, print it and close

// reading end.

read(fd2[0], concat\_str, 100);

printf("Concatenated string %s\n", concat\_str);

close(fd2[0]);

}

// child process

else

{

close(fd1[1]); // Close writing end of first pipe

// Read a string using first pipe

char concat\_str[100];

read(fd1[0], concat\_str, 100);

// Concatenate a fixed string with it

int k = strlen(concat\_str);

int i;

for (i=0; i<strlen(fixed\_str); i++)

concat\_str[k++] = fixed\_str[i];

concat\_str[k] = '\0'; // string ends with '\0'

// Close both reading ends

close(fd1[0]);

close(fd2[0]);

// Write concatenated string and close writing end

write(fd2[1], concat\_str, strlen(concat\_str)+1);

close(fd2[1]);

exit(0);

}

}

***Second:-***

#include<stdio.h>

#include<unistd.h>

int main() {

int pipefds[2];

int returnstatus;

char writemessages[2][20]={"Hi", "Hello"};

char readmessage[20];

returnstatus = pipe(pipefds);

if (returnstatus == -1) {

printf("Unable to create pipe\n");

return 1;

}

printf("Writing to pipe - Message 1 is %s\n", writemessages[0]);

write(pipefds[1], writemessages[0], sizeof(writemessages[0]));

read(pipefds[0], readmessage, sizeof(readmessage));

printf("Reading from pipe – Message 1 is %s\n", readmessage);

printf("Writing to pipe - Message 2 is %s\n", writemessages[0]);

write(pipefds[1], writemessages[1], sizeof(writemessages[0]));

read(pipefds[0], readmessage, sizeof(readmessage));

printf("Reading from pipe – Message 2 is %s\n", readmessage);

return 0;

}

***Link:-***

<https://www.tutorialspoint.com/inter_process_communication/inter_process_communication_pipes.htm>

<https://www.geeksforgeeks.org/c-program-demonstrate-fork-and-pipe/>

<https://github.com/SAKET-SK/Semester5-SPPU-Operating-System-Lab/blob/main/Assignment%209)%20Simple%20Pipe%20Demonstration/ass9.c>