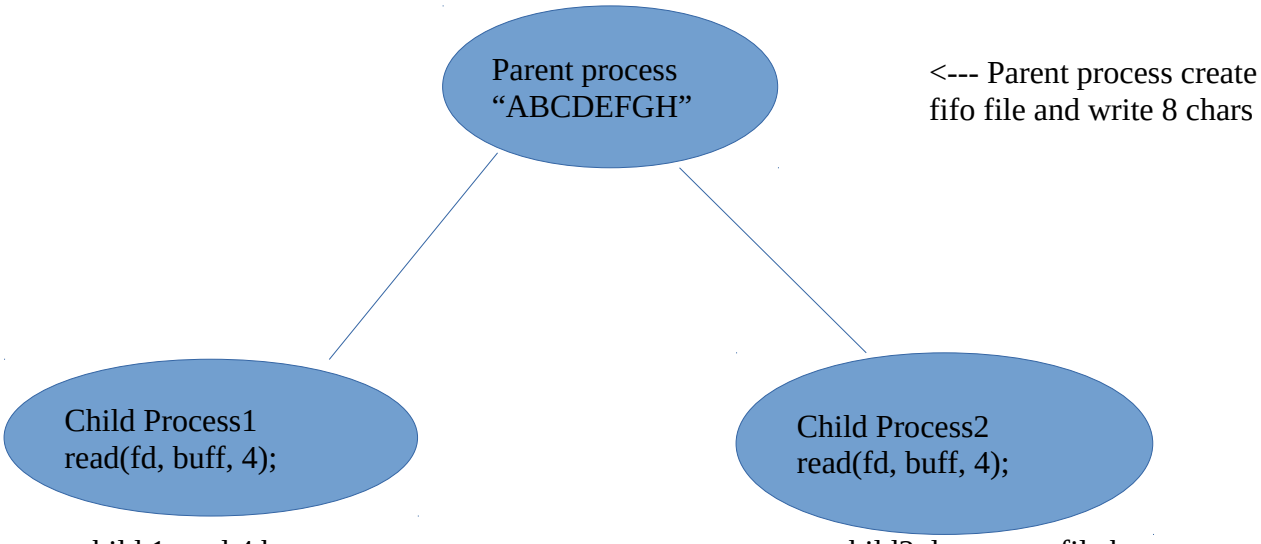

Application Of Parent Process:

```
/*
        named pipe Demo

        in this application we create 3 process  first is parent process which create fifo pipe also known as named pipe
        this pipe exists into file system which get inode number, by using mkfifo() system call we create fifo pipe
        to read and write. pipe creation done in parent process, then parent process write data into fifo file and
        that data is read by two child process half-half
        example:
                parent process :

                mkfifo("/tmp/abc.txt", 0666);

                fifo file gets created at given path

                now parent process open it to write data into fifo file

                written data is:
                                        A               b               C               D
        E               F               G       H
                                        ------------------------        ----------------------
                                        child 1 read this data                  child 2 read this data

                after complete read of fifo file data then fifo file gets empty.


                we can not read parent writte data 8 bytes in from 2 process
                only one process can read data at once
*/
```

```
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<string.h>

int main()
{
	int ret, fd;
	char buff[8] = {'\0'};
	char *path = "/tmp/myfifo";


	ret  = mkfifo(path, 0666);
	if(ret == -1)
	{
		perror("mkfifo");
		exit(-1);
	}



	fd = open(path, O_WRONLY);
	if(fd == -1)
	{
		perror("open");
		exit(-1);
	}


	write(fd, "ABCDEFGH", strlen("ABCDEFGH"));


	printf("data read child successfully\n");

	exit(EXIT_SUCCESS);
}
```

Application of child 1

```
/*

	child 1
	this child read some data from pipe

*/
```

```
#include<stdio.h>
#include<unistd.h>
#include<fcntl.h>
#include<stdlib.h>


int main()
{
        int fd, ret;
        char buff[4] = {'\0'};
        char *path = "/tmp/myfifo";


        fd = open(path, O_RDONLY);
        if(fd == -1)
        {
                perror("read");
                exit(-1);
        }

        read(fd, buff, 4);

        printf("child 1 read data is: %s\n",buff);

        close(fd);

        exit(0);

}
```

Application child 2

```
/*

        child 2
        this child read some data from pipe

*/


#include<stdio.h>
#include<unistd.h>
#include<fcntl.h>
#include<stdlib.h>


int main()
{
        int fd, ret;
        char buff[4] = {'\0'};
        char *path = "/tmp/myfifo";
```

```
        fd = open(path, O_RDONLY);
        if(fd == -1)
        {
                perror("read");
                exit(-1);
        }

        read(fd, buff, 4);

        printf("child 2 read data is: %s\n",buff);

        close(fd);

        exit(0);

}
```

---

only one child can read pipe at a time
if another process try to reread fifo file then it won’t get anything
so if we want to read data from fifo file then read it at once completely