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

#define FIFO1 "/faculty/abukmail/5531/fifo.1"
#define FIFO2 "/faculty/abukmail/5531/fifo.2"

#define FILE_MODE 0666
int main(int argc, char **argv)
{

  int readfd, writefd, n;
  pid_t childpid;
  char buff[100];

  /* create two FIFOs; OK if they already exist */
  if ((mkfifo(FIFO1, FILE_MODE) < 0) && (errno != EEXIST))
    printf("Cannot create %s : %s\n", FIFO1, strerror(errno));

  if ((mkfifo(FIFO2, FILE_MODE) < 0) && (errno != EEXIST)) {
    unlink(FIFO1);
    printf("Cannot create %s : %s\n", FIFO2, strerror(errno));

  }
  if ( (childpid = fork()) == 0) {/* child */
    readfd = open(FIFO1, O_RDONLY, 0);
    writefd = open(FIFO2, O_WRONLY, 0);
    write (writefd, "Hello from child\n", 17);
    n = read(readfd, buff, 100);
    buff[n] = '\0';
    printf("PID=%d  child read: %s\n", childpid, buff);
    exit(0);
  }
  /* parent */
  writefd = open(FIFO1, O_WRONLY, 0);
  readfd = open(FIFO2, O_RDONLY, 0);
  write (writefd, "Hello from parent\n", 18);
  n = read(readfd, buff, 100);
  buff[n] = '\0';
  printf("PID=%d  parent read: %s\n", childpid, buff);

  waitpid(childpid, NULL, 0);/* wait for child to terminate */
  close(readfd);
  close(writefd);
  unlink(FIFO1);
  unlink(FIFO2);
  exit(0);
}
```