```c
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

int main(void)
{
        int i=0;
        FILE *in, *out;
        char *f[4]={"f0","f1","f2","f3"}; /* cat f[012] >> f3 */
        int status=0;
        char str[BUFSIZ];
        pid_t pid;

        memset(str, '\0', sizeof(str));

    for(;i<3;i++){
                if( (pid=fork()) == 0 ){                    /* child */
                        printf("PID: %d\n",getpid());
                        if( (in=fopen(f[i],"w+")) == NULL ) {
                                perror("fopen<in>");
                                exit(EXIT_FAILURE);
                        }

                        printf("Input something into file %s\n", f[i]);
                        if( fgets(str,BUFSIZ,stdin)==NULL ){
                                        perror("fgets");
                                        exit(EXIT_FAILURE);
                                }
                        printf("Your input is: %s",str);

                        if( !fputs(str, in) ){
                                        perror("fputs");
                                        exit(EXIT_FAILURE);
                        }
                        /* if( fflush(in) != 0 ){ */
                        /*                 perror("fflush"); */
                        /*                 exit(EXIT_FAILURE); */
                        /* } */
                        fclose(in);
                        printf("Done writing into %s.\n\n", f[i]);
                        exit(EXIT_SUCCESS);
                }
```

```
45                  else
46                          if( waitpid(pid,&status,WUNTRACED) == -1){
47                                  perror("wait");
48                                  exit(EXIT_FAILURE);
49                          }
50          }
51
52          /* parent */
53          if( (out=fopen(f[3], "a+")) == NULL ) {
54                  perror("fopen<out>");
55                  exit(EXIT_FAILURE);
56          }
57
58          for(i=0;i<3;i++){
59                  if( (in=fopen(f[i],"r")) == NULL ) {
60                          perror("fopen<out>");
61                          exit(EXIT_FAILURE);
62                  }
63
64                  if( fgets(str,BUFSIZ,in) == NULL ){
65                          perror("fgets");
66                          exit(EXIT_FAILURE);
67                  }
68
69                  if( fputs(str, out) < 0 ){
70                          perror("fputs");
71                          exit(EXIT_FAILURE);
72                  }
73
74                  fclose(in);
75          }
76
77          if( fseek(out,0L,SEEK_END) !=0 ){
78                          perror("fseek");
79                          exit(EXIT_FAILURE);
80          }
81          printf("The size of %s is %ld.\n", f[3], ftell(out));
82
83          fclose(out);
84          return 0;
85  }
86
87  /* Local Variables: */
88  /* compile-command: "gcc -Wall -Wextra fork-file-sharing.c -o /tmp/a.out" */
89  /* End: */
```