

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <pthread.h>
6
7 void *writefile(void *arg)
8 {
9     FILE *in;
10    char str[BUFSIZ];
11
12    memset(str, '\0', sizeof(str));
13
14    if( (in=fopen((char *)arg,"w+")) == NULL ) {
15        perror("fopen<in>");
16        exit(EXIT_FAILURE);
17    }
18
19    printf("Input something into file %s\n", (char *)arg);
20
21    if( fgets(str,BUFSIZ,stdin)==NULL ){
22        perror("fgets");
23        exit(EXIT_FAILURE);
24    }
25
26    printf("Your input is: %s",str);
27
28    if( !fputs(str, in) ){
29        perror("fputs");
30        exit(EXIT_FAILURE);
31    }
32
33    fclose(in);
34
35    printf("Done writing into %s.\n\n", (char *)arg);
36
37    pthread_exit(NULL);
38 }
39
40 int main(void)
41 {
42     int i=0, status;
43     FILE *in, *out;
44     char *f[4]={"f0","f1","f2","f3"}; /* cat f[012] >> f3 */
45 }
```

```
45 pthread_t t[3];
46 char str[BUFSIZ];
47
48 memset(str, '\\0', sizeof(str));
49
50 for (; i<3; i++){
51     printf("Main: creating thread %d... ", i);
52
53     if( (status = pthread_create(&t[i], NULL, writefile, (void *)f[i]
54         perror("pthread_create");
55         exit(-1);
56     }
57     puts("Done.");
58
59     /* Joining current thread before starting another so that the ST
60        raced by multiple threads. */
61     printf("Joining thread %d... ", i);
62
63     if( pthread_join(t[i], NULL) ){
64         perror("pthread_join");
65         abort() ;
66     }
67     puts("Done.");
68 }
69
70 /* parent */
71 if( (out=fopen(f[3], "a+")) == NULL ) {
72     perror("fopen<out>");
73     exit(EXIT_FAILURE);
74 }
75
76 for(i=0;i<3;i++){
77     if( (in=fopen(f[i], "r")) == NULL ) {
78         perror("fopen<out>");
79         exit(EXIT_FAILURE);
80     }
81
82     if( fgets(str, BUFSIZ, in) == NULL ){
83         perror("fgets");
84         exit(EXIT_FAILURE);
85     }
86
87     if( fputs(str, out) < 0 ){
88         perror("fputs");
89         exit(EXIT_FAILURE);
90     }
91 }
```

```
91         fclose(in);
92     }
93
94     if( fseek(out,0L,SEEK_END) !=0 ){
95         perror("fseek");
96         exit(EXIT_FAILURE);
97     }
98
99     printf("The size of %s is %ld.\n", f[3], ftell(out));
100
101     fclose(out);
102     return 0;
103 }
104
105
106 /* Local Variables: */
107 /* compile-command: "gcc -Wall -Wextra file-sharing-thread.c -o /tmp/a.out -pthr
108 /* End: */
```