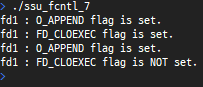
컴퓨터학부 20162448 김병준

1. 결과



1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <fcntl.h>  #include <string.h>  int main(){  char \*filename = "ssu\_test.txt";  int fd1, fd2;  int flag;  if((fd1 = open(filename, O\_RDWR|O\_APPEND, 0644)) < 0){ // open file  fprintf(stderr, "open error for %s\n", filename);  exit(1);  }  if(fcntl(fd1, F\_SETFD, FD\_CLOEXEC) == -1){ // set close-on-exec bit & set file  fprintf(stderr, "fcntl F\_SETFD error\n");  exit(1);  }  if((flag = fcntl(fd1, F\_GETFL, 0)) == -1){ // get file flag  fprintf(stderr, "fcntl F\_GETFL error\n");  exit(1);  }  if(flag & O\_APPEND) // check flag contain O\_APPEND flag  printf("fd1 : O\_APPEND flag is set. \n");  else  printf("fd1 : O\_APPEND flag is NOT set.\n");  if((flag = fcntl(fd1, F\_GETFD, 0)) == -1){ // get file flag  fprintf(stderr, "fcntl F\_GETFL error\n");  exit(1);  }  if(flag & FD\_CLOEXEC) // check flag contain FD\_CLOEXEC flag  printf("fd1 : FD\_CLOEXEC flag is set. \n");  else  printf("fd1 : FD\_CLOEXEC flag is NOT set.\n");  if((fd2 = fcntl(fd1, F\_DUPFD, 0)) == -1){ // duplicate filedescriptor  fprintf(stderr, "fcntl F\_DUPFD error\n");  exit(1);  }  if((flag = fcntl(fd2, F\_GETFL, 0)) == -1){ // get file flag  fprintf(stderr, "fcntl F\_GETFL error\n");  exit(1);  }  if(flag & O\_APPEND) // check flag contain O\_APPEND flag  printf("fd1 : O\_APPEND flag is set. \n");  else  printf("fd1 : O\_APPEND flag is NOT set.\n");  if((flag = fcntl(fd2, F\_GETFD, 0)) == -1){ // get file flag  fprintf(stderr, "fcntl F\_GETFL error\n");  exit(1);  }  if(flag & FD\_CLOEXEC) // check flag contain FD\_CLOEXEC flag  printf("fd1 : FD\_CLOEXEC flag is set. \n");  else  printf("fd1 : FD\_CLOEXEC flag is NOT set.\n");  exit(0);  } |