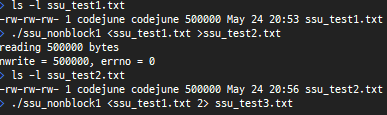
컴퓨터학부 20162448 김병준

1. 결과



1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <errno.h>  #include <fcntl.h>  void set\_flags(int fd, int flags);  void clr\_flags(int fd, int flags);  char buf[500000];  int main(void)  {  int ntowrite, nwrite;  char \*ptr;  // 표준입력으로부터 읽어들임  ntowrite = read(STDIN\_FILENO, buf, sizeof(buf));  // 바이트 수 출력  fprintf(stderr, "reading %d bytes\n", ntowrite);  // 표준 출력 nonblocking 설정  set\_flags(STDOUT\_FILENO, O\_NONBLOCK);  ptr=buf;  // 모든 버퍼에 쓰기  while(ntowrite>0)  {  errno=0;  // 표준출력에 쓰기  nwrite = write(STDOUT\_FILENO, ptr, ntowrite);  fprintf(stderr, "nwrite = %d, errno = %d\n", nwrite, errno);  if (nwrite>0)  {  ptr+=nwrite;  ntowrite-=nwrite;  }  }  // 디스크럽터로부터 nonblock 플래그 제거  clr\_flags(STDOUT\_FILENO, O\_NONBLOCK);  exit(0);  }  void set\_flags(int fd, int flags)  {  int val;  // 파일 플래그 가져오기  if ((val = fcntl(fd, F\_GETFL, 0))<0)  {  fprintf(stderr, "fcntl F\_GETFL failed");  exit(1);  }  // 현재 플래그에서 플래그 추가  val|=flags;  // 파일에 플래그 지정  if (fcntl(fd, F\_SETFL, val)<0)  {  fprintf(stderr, "fcntl F\_SETFL failed");  exit(1);  }  }  void clr\_flags(int fd, int flags)  {  int val;  // 파일 플래그 가져오기  if ((val = fcntl(fd, F\_GETFL, 0))<0)  {  fprintf(stderr, "fcntl F\_GETFL failed");  exit(1);  }  // 현재 플래그에서 플래그 제외  val&=~flags;  // 파일에 플래그 지정  if(fcntl(fd, F\_SETFL, val)<0)  {  fprintf(stderr, "fcntl F\_SETFL failed");  exit(1);  }  } |