1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #define BUFFER\_SIZE 1024  void ssu\_setbuf(FILE \*fp, char \*buf);  int main(void) {  char buf[BUFFER\_SIZE];  //현재 표준 입력에 접속된 터미널 장치 파일명  char \*fname = "/dev/pts/17";  FILE \*fp;  //fname 파일을 읽기 모드로 open  if((fp = fopen(fname, "w")) == NULL) {  fprintf(stderr, "fopen error for %s\n", fname);  exit(1);  }  //fp에 BUFFER\_SIZE 만큼의 버퍼 크기를 설정  ssu\_setbuf(fp, buf);  fprintf(fp, "Hello, ");  sleep(1);  fprintf(fp, "UNIX!!");  sleep(1);  //"Hello UNIX!!"를 버퍼에 저장해놨다가 new line을 만나면 출력  fprintf(fp, "\n");  sleep(1);  //Non buffering  ssu\_setbuf(fp, NULL);  sleep(1);  //즉시 출력  fprintf(fp, "HOW");  sleep(1);  fprintf(fp, " ARE");  sleep(1);  fprintf(fp, " YOU?");  sleep(1);  fprintf(fp, "\n");  sleep(1);  exit(0);  }  void ssu\_setbuf(FILE \*fp, char \*buf) {  size\_t size;  int fd;  int mode;  //파일 디스크립터 추출  fd = fileno(fp);  if(isatty(fd))//fd가 terminal와 연결된 파일 디스크립터인지 테스트  mode = \_IOLBF;//line buffering  else//fully buffering  mode = \_IOFBF;    if(buf == NULL) {  mode = \_IONBF;//Non buffering  size = 0;  }  else  size = BUFFER\_SIZE;    //설정된 모드, 사이즈를 이용하여 setvbuf 호출  setvbuf(fp, buf, mode, size);  } |

2. 실행결과

