

6주차_ 입출력다중화

데이터 네트워크연구실
이현호

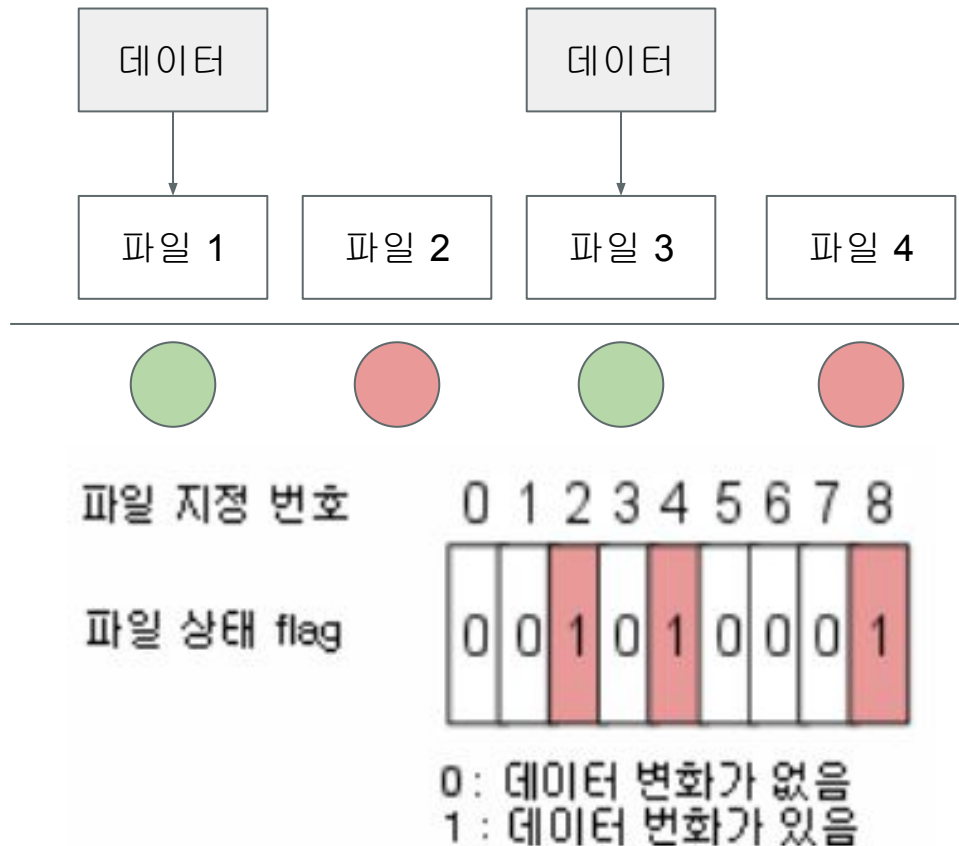
lee075@cs-cnu.org

Goals

- 입출력의 다중화 이해하기
- 타임아웃 구현하기
- **Select** 함수를 이용한 입출력 다중화 구현하기
- 입출력 다중화를 소켓통신에 적용하기

Input / Output

- 작업 교환 단위가 문맵이 아닌 파일 입출력 이라면?
- 파일 1, 3은 읽어야 할 파일
- 읽어야 할 파일을 체크 해 놓고 읽기 함수로 읽는다



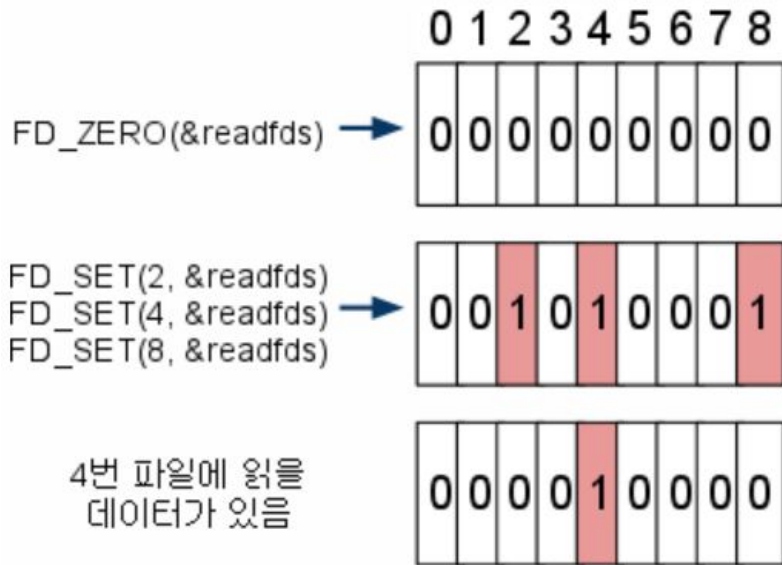
Select Function

```
int select (int nfds, fd_set *readfds, fd_set *writefds,  
            fd_set *exceptfds, struct timeval *timeout);
```

- **nfds**: 관리하는 파일의 개수를 등록한다. 파일의 개수는 최대 파일 지정 번호 + 1로 지정
- **fd_set**: 관리하는 파일의 지정번호가 등록되어 있는 비트 배열 구조체
 - **readfds**: 읽을 데이터가 있는지 검사하기 위한 파일 목록
 - **writefds**: 쓰여진 데이터가 있는지 검사하기 위한 파일 목록
 - **exceptfds**: 파일에 예외 사항들이 있는지 검사하기 위한 파일 목록
- **timeout**: **select** 함수는 **fd_set**에 등록된 파일들에 데이터 변경이 있는지를 **timeout** 동안 기다린다. 만약 **timeout** 시간 동안 변경이 없다면 0을 반환

fd_set 관리

- fd_set은 단일 비트 필드 테이블이라는 것
- 데이터가 변경된 파일의 개수 즉 fd_set에서 비트 값이 1인 필드의 개수를 반환
 - 변경된 파일의 목록을 반환하지 않는다는 것에 주의
- FD_ZERO(fd_set *fds); 0으로 초기화
- FD_SET(int fdnum, *fds): 관리하기 위한 목록추가
- FD_ISSET(fdnum, *fds): 파일에 변화가 생기면 1
- FD_CLR(fd, *fds); : fd를 fds 비트 필드에서 제거




- 일정 시간내에 변화가 있는지 감지하는 프로그램

C++


```
typedef struct timeval {  
    long tv_sec;  
    long tv_usec;  
} timeval;
```

```
1 #include <sys/time.h>  
2 #include <sys/types.h>  
3 #include <unistd.h>  
4 #include <fcntl.h>  
5 #include <stdlib.h>  
6 #include <string.h>  
7 #include <stdio.h>  
8  
9 #define STDIN 0  
10 #define MAX_LINE 80  
11  
12 int main(int argc, char **argv)  
13 {  
14     fd_set readfds;  
15     int fdn;  
16     char rbuf[MAX_LINE];  
17     struct timeval timeout;  
18  
19     FD_ZERO(&readfds);
```

```
20 FD_SET(STDIN, &readfds);
21
22 while(1)
23 {
24     timeout.tv_sec = (long)5;
25     timeout.tv_usec = 0L;
26     printf("> ");
27     fflush(stdout);
28     fdn=select(STDIN+1, &readfds, NULL, NULL, &timeout);
29     if(fdn == 0)
30     {
31         printf("\nError : Time Out\n");
32     }
33     else
34     {
35         memset(rbuf, 0x00, MAX_LINE);
36         read(STDIN, rbuf, MAX_LINE - 1);
37         if(strncmp(rbuf, "quit\n",5) == 0)
38             break;
```

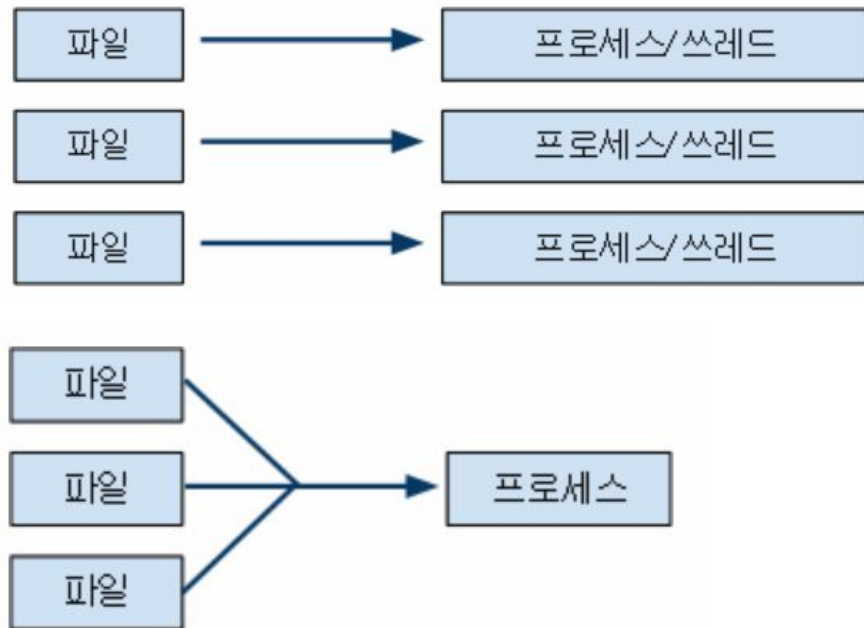
```
39             printf("Your Message : %s", rbuf);
40         }
41         FD_SET(STDIN, &readfds);
42     }
43     return 0;
44 }
45
46 
```


Result

```
hyunholee@DNLAB:~/temp/IO$ ./prompt_timeout
> hi
Your Message : hi
> heoo
Your Message : heoo
> hello
Your Message : hello
>
Error : Time Out
>
Error : Time Out
> 
```

Input / Output

- 관리 할 파일을 등록
- 변화가 생기면 출력
- 입출력의 다중화



```
1 #include <sys/types.h>
2 #include <sys/time.h>
3 #include <unistd.h>
4 #include <stdlib.h>
5 #include <string.h>
6 #include <fcntl.h>
7 #include <sys/stat.h>
8 #include <stdio.h>
9
10 #define MAX_LINE 1024
11
12 void ReadLine(int maxfd, fd_set *readfds);
13
14 int main(int argc, char **argv)
15 {
16     int tfd[2];
17     int i = 0;
18     int maxfd = 0;
19     int fdn = 0;
```

```
20     fd_set readfds;
21
22     if (argc != 3)
23     {
24         fprintf(stderr, "Usage : %s [file 1] [file 2]\n", argv[0]);
25         return 1;
26     }
27
28     FD_ZERO(&readfds);
29     for(i = 0; i < 2; i++)
30     {
31         printf("Open %s\n", argv[i+1]);
32         tfd[i] = open(argv[i+1], O_RDWR);
33         if(tfd[i] == -1)
34         {
35             perror("Error");
36             return 1;
37         }
38         if (tfd[i] > maxfd)
```

```
39         maxfd = tfd[i];
40         FD_SET(tfd[i], &readfds);
41     }
42
43     printf("Max Fd is %d\n", maxfd);
44     while(1)
45     {
46         fdn = select(maxfd+1, &readfds, NULL, NULL, NULL);
47         ReadLine(maxfd+1, &readfds);
48     }
49 }
50
51 void ReadLine(int maxfd, fd_set *readfds)
52 {
53     char buf[MAX_LINE];
54     int i = 0;
55
56     for(i = 0; i < maxfd; i++)
57     {
```

```
58     if(FD_ISSET(i, readfds))
59     {
60         while(1)
61         {
62             memset(buf, 0x00, MAX_LINE);
63             if(read(i, buf, MAX_LINE-1) < MAX_LINE-1) break;
64             printf("(%d) -> %s", i, buf);
65         }
66         printf("(%d) -> %s", i, buf);
67     }
68     FD_SET(i, readfds);
69 }
70 }
71
72
```

Result

```
hyunholee@DNLAB:~/temp/IO$  
Open /tmp/myfifo1  
Open /tmp/myfifo2  
Max Fd is 4  
(3) -> hello  
hi  
my name is  
(4) -> fine  
(4) -> fine  
and you?  
█
```

```
fine  
and you?  
~
```

```
hello  
hi  
my name is█
```

소켓통신과 입출력 다중화

- `socket`: 성공할 경우 정수의 파일 지정 번호


```
1 #include <sys/time.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4
5 #include <sys/socket.h>
6 #include <sys/types.h>
7
8 #include <netinet/in.h>
9 #include <arpa/inet.h>
10
11 #include <stdio.h>
12 #include <string.h>
13
14 #define MAXLINE 1024
15 #define PORTNUM 6292
16 #define SOCK_SETSIZE 1021
17
18 int main(int argc, char **argv)
19 {
```


```
20     int listen_fd, client_fd;
21     socklen_t addrlen;
22     int fd_num;
23     int maxfd = 0;
24     int sockfd;
25     int i= 0;
26     char buf[MAXLINE];
27     fd_set readfds, allfds;
28
29     struct sockaddr_in server_addr, client_addr;
30
31     if((listen_fd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
32     {
33         perror("socket error");
34         return 1;
35     }
36     memset((void *)&server_addr, 0x00, sizeof(server_addr));
37     server_addr.sin_family = AF_INET;
38     server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
```

```
39     server_addr.sin_port = htons(PORTNUM);
40
41     if(bind(listen_fd, (struct sockaddr *)&server_addr, sizeof(server_addr)) == -1)
42     {
43         perror("bind error");
44         return 1;
45     }
46     if(listen(listen_fd, 5) == -1)
47     {
48         perror("listen error");
49         return 1;
50     }
51
52     FD_ZERO(&readfds);
53     FD_SET(listen_fd, &readfds);
54
55     maxfd = listen_fd;
56     while(1)
```

소켓통신

```
57     {
58         allfds = readfds;
59         printf("Select Wait %d\n", maxfd);
60         fd_num = select(maxfd + 1, &allfds, (fd_set *)0,
61                         (fd_set *)0, NULL);
62
63         if (FD_ISSET(listen_fd, &allfds))
64         {
65             addrlen = sizeof(client_addr);
66             client_fd = accept(listen_fd,
67                               (struct sockaddr *)&client_addr,
68                               &addrlen);
69
70             FD_SET(client_fd, &readfds);
71
72             if (client_fd > maxfd)
73                 maxfd = client_fd;
74             printf("Accept OK\n");
75             continue;
```

```
75     }
76
77     for (i = 0; i <= maxfd; i++)
78     {
79         sockfd = i;
80         if (FD_ISSET(sockfd, &allfds))
81         {
82             memset(buf, 0x00, MAXLINE);
83             if (read(sockfd, buf, MAXLINE) <= 0)
84             {
85                 close(sockfd);
86                 FD_CLR(sockfd, &readfds);
87             }
88             else
89             {
90                 if (strncmp(buf, "quit\n", 5) == 0)
91                 {
92                     close(sockfd);
93                     FD_CLR(sockfd, &readfds);
```

```
94                                     break;
95                                 }
96                                 printf("Read : %s", buf);
97                                 write(sockfd, buf, strlen(buf));
98                             }
99                             if (--fd_num <= 0)
100                                 break;
101                             }
102                         }
103                     }
104 }
105
106 
```


Result

```
Read : Hi my name is hyunho
Select Wait 5
Read : My name is Emi
Select Wait 5
Read : My number is 19
Select Wait 5
Read : My number is 20
Select Wait 5
```



```
Hi my name is hyunho
send : Hi my name is hyunho
read : Hi my name is hyunho
My number is 19
send : My number is 19
read : My number is 19
```

```
My name is Emi
send : My name is Emi
read : My name is Emi
My number is 20
send : My number is 20
read : My number is 20
```

