



12

소켓 프로그래밍 활용

IT CookBook, 유닉스 시스템 프로그래밍

학습목표

- 소켓 인터페이스를 활용한 다양한 프로그램을 작성할 수 있다.



목차

- TCP 기반 프로그래밍
- 반복서버
- 동시동작서버
- 동시동작서버-exec함수 사용하기
- 동시동작서버-명령행인자로 소켓 기술자 전달하기
- UDP 프로그래밍



TCP 기반 프로그래밍

□ 반복서버

- 데몬 프로세스가 직접 모든 클라이언트의 요청을 차례로 처리 **한 프로세스에 과부하**

□ 동시동작서버

- 데몬 프로세스가 직접 서비스를 제공하지 않고, 서비스와 관련있는 다른 프로세스를 fork 함수로 생성해 클라이언트와 연결시켜준다. **요청을 처리할 프로세스를 만들어서
떠넘긴다**



[예제 12-1] (1) 반복서버(서버)

ex12_1s.c

```
<sys/types.h><sys/socket.h><unistd.h><netdb.h><netinet.h><arpa/inet.h>
<stdlib.h><stdio.h><string.h>
11 #define PORTNUM 9001
12 int main(void) {
13     char buf[256];
14     struct sockaddr_in sin, cli;
15     int sd, ns, clientlen = sizeof(cli);
16
17     memset((char *)&sin, '\0', sizeof(sin));
18     sin.sin_family = AF_INET;
19     sin.sin_port = htons(PORTNUM);
20     sin.sin_addr.s_addr = inet_addr("192.168.162.133"); 127.0.0.1
21
22     if ((sd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
23         perror("socket");
24         exit(1);
25     }
26
27     if (bind(sd, (struct sockaddr *)&sin, sizeof(sin))) {
28         perror("bind");
29         exit(1);
30     }
31
32     if (listen(sd, 5)) {
33         perror("listen");
34         exit(1);
35     }
```

소켓 주소구조체 생성

소켓 생성

클라 최대 5인수용

클라이언트 접속 대기

```
37  while (1) {
38      if ((ns = accept(sd, (struct sockaddr *)&cli, &clientlen)) == -1) {
39          perror("accept");
40          exit(1);
41      }
42      sprintf(buf, "%s", inet_ntoa(cli.sin_addr));
43      printf("*** Send a Message to Client(%s)\n", buf);
44
45      strcpy(buf, "Welcome to Network Server!!!");
46      if (send(ns, buf, strlen(buf) + 1, 0) == -1) {
47          perror("send");
48          exit(1);
49      }
50
51      if (recv(ns, buf, strlen(buf), 0) == -1) {
52          perror("recv");
53          exit(1);
54      }
55      printf("** From Client : %s\n", buf);
56      close(ns);
57  }
58  close(sd);
59
60  return 0;
61 }
```

클라이언트 접속

클라이언트에 정보전송

클라이언트의 데이터 수신

```
<sys/types.h><sys/socket.h><netinet.h><arpa/inet.h>
<stdlib.h><stdio.h><string.h>
11  #define PORTNUM 9001
12
13  int main(void) {
14      int sd;
15      char buf[256];
16      struct sockaddr_in sin;
17
18      memset((char *)&sin, '\0', sizeof(sin));
19      sin.sin_family = AF_INET;
20      sin.sin_port = htons(PORTNUM);
21      sin.sin_addr.s_addr = inet_addr("192.168.162.133"); 127.0.0.1
22
23      if ((sd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
24          perror("socket");
25          exit(1);
26      }
27
28      if (connect(sd, (struct sockaddr *)&sin, sizeof(sin))) {
29          perror("connect");
30          exit(1);
31      }
```

소켓 주소구조체 생성

소켓 생성

서버에 연결 요청

```

33     if (recv(sd, buf, sizeof(buf), 0) == -1) {
34         perror("recv");
35         exit(1);
36     }
37
38     printf("** From Server : %s\n", buf);
39
40     strcpy(buf, "I want a HTTP Service.");
41     if (send(sd, buf, sizeof(buf) + 1, 0) == -1) {
42         perror("send");
43         exit(1);
44     }
45
46     close(sd);
47
48     return 0;
49 }

```

서버의 데이터 수신

서버에 데이터 송신

ex12_1s.out

서버

ex12_1c.out

클라이언트

** From Server : Welcome to Network Server!!!

ex12_1s.out

서버

*** Send a Message to Client(192.168.162.133)

** From Client : I want a HTTP Service.

ex12_1s.out

클라이언트

*** Send a Message to Client(192.168.162.131)

** From Client : I want a FTP Service.


```
<sys/types.h><sys/socket.h><unistd.h><netinet.h><arpa/inet.h>
<stdlib.h><stdio.h><string.h>
11  #define PORTNUM 9002
12  int main(void) {
13      char buf[256];
14      struct sockaddr_in sin, cli;
15      int sd, ns, clientlen = sizeof(cli);
16
17      if ((sd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
18          perror("socket");
19          exit(1);
20      }
21
22      memset((char *)&sin, '\0', sizeof(sin));
23      sin.sin_family = AF_INET;
24      sin.sin_port = htons(PORTNUM);
25      sin.sin_addr.s_addr = inet_addr("192.168.162.133");
26
27      if (bind(sd, (struct sockaddr *)&sin, sizeof(sin))) {
28          perror("bind");
29          exit(1);
30      }
31
32      if (listen(sd, 5)) {
33          perror("listen");
34          exit(1);
35      }
```

```
37     while (1) {
38         if ((ns = accept(sd, (struct sockaddr *)&cli, &clientlen)) == -1) {
39             perror("accept");
40             exit(1);
41         }
42         switch (fork()) {
43             case 0: 자식
44                 close(sd);
45                 strcpy(buf, "Welcome to Server");
46                 if (send(ns, buf, strlen(buf) + 1, 0) == -1) {
47                     perror("send");
48                     exit(1);
49                 }
50
51                 if (recv(ns, buf, strlen(buf), 0) == -1) {
52                     perror("recv");
53                     exit(1);
54                 }
55                 printf("** From Client: %s\n", buf);
56                 sleep(5);
57                 exit(0);
58             }
59         close(ns);
60     }
61
62     return 0;
63 }
```

fork로 자식 프로세스 생성

자식 프로세스가
클라이언트로
메시지 보내고
데이터 수신

[예제 12-2] 실행결과

```
# ex12_2s.out
*** Send a Message to Client(192.168.162.133)
** From Client : I want a HTTP Service.
```

- 클라이언트는 ex12_1c.c를 포트번호만 바꾸고 그대로 사용
- 클라이언트가 접속했을 때 서버의 실행상태

```
# ps -ef | grep pts/2
root 1571 1568 0 2월 08 pts/2 0:03 _ksh
root 7175 7172 0 09:55:32 pts/2 0:00 ex12_2s.out
root 7172 1571 0 09:55:18 pts/2 0:00 ex12_2s.out
```

- 서버 프로세스가 2개 임을 알 수 있다.
- 7172는 부모 프로세스, 7175는 자식 프로세스

자식이 5초간은 생존



[예제 12-3] (1) 동시동작서버 – exec 함수 사용하기(서버)

ex12_3s.c

```
<sys/types.h><sys/socket.h><unistd.h><netinet.h><arpa/inet.h>
<stdlib.h><stdio.h><string.h>
40     while (1) {
41         if ((ns = accept(sd, (struct sockaddr *)&cli,
                        &clientlen)) == -1) {
42             perror("accept");
43             exit(1);
44         }
45         printf("** Acceptspt Client\n");
46
47         switch (fork()) {
48             case 0:
49                 printf("** Fork Client\n");
50                 close(sd);
51                 ns를 표준입력 dup2(ns, STDIN_FILENO);
52                 ns를 표준출력 dup2(ns, STDOUT_FILENO);
53                 close(ns);
54                 → execl("./han", "han", (char *)0);
55             }
56         close(ns);
57     }
58
59     return 0;
60 }
```

클라이언트의 요청 처리를 위한
별도의 프로그램(han) 실행

```
01  #include <unistd.h>
02  #include <stdio.h>
03
04  int main(void) {
05      printf("Welcome to Server, from Han!");
06      sleep(5);
07
08      return 0;
09  }
```

간단한 환영메시지 출력



[예제 12-3] (3) 동시동작서버 - exec 함수 사용(클라이언트)

ex12_3c.c

```
<sys/types.h><sys/socket.h><unistd.h><netdb.h><netinet/in.h><arpa/inet
.h><stdlib.h><stdio.h><string.h>
28     printf("==> Create Socket\n");
29     if (connect(sd, (struct sockaddr *)&sin, sizeof(sin))) { 권한받기
30         perror("connect");
31         exit(1);
32     }
33
34     printf("==> Connect Server\n");
35     if ((len = recv(sd, buf, sizeof(buf), 0)) == -1) { 받기
36         perror("recv");
37         exit(1);
38     }
39     buf[len] = '\0';
40
41     printf("==> From Server : %s\n", buf);
42
43     close(sd);
44
45     return 0;
46 }
```

연결요청

메시지 수신

```
# ex12_3c.out
==> Create Socket
==> Connect Server
==> From Server : Welcome to Server, from Han!
```

클라이언트

```
# ps
PID TTY          TIME CMD
676 pts/2        0:00 ksh
760 pts/2        0:00 ex12_3s.out
763 pts/2        0:00 han
```

han 실행

5초간은 자식 생존 →

```
<sys/types.h><sys/socket.h><unistd.h><netdb.h><netinet/in.h><arpa/inet  
.h><stdlib.h><stdio.h><string.h>  
40     while (1) {  
41         if ((ns = accept(sd, (struct sockaddr *)&cli,  
                           &clientlen)) == -1) {  
42             perror("accept");  
43             exit(1);  
44         }  
45         printf("** Accept Client\n");  
46  
47         switch (fork()) {  
48             case 0:  
49                 printf("** Fork Client\n");  
50                 close(sd);  
51                 sprintf(buf, "%d", ns);  
52                 execlp("./bit", "bit", buf, (char *)0);  
53                 close(ns);  
54             }  
55             close(ns);  
56         }  
57  
58     return 0;  
59 }
```

클라이언트 접속 수용

bit프로그램 실행
명령행 인자로 소켓 전달

```
<sys/types.h><sys/socket.h><unistd.h><stdlib.h><stdio.h><string.h>
08 int main(int argc, char *argv[]) {
09     char buf[256];
10     int len, ns;
11
12     ns = atoi(argv[1]);
13     buf
14     strcpy(buf, "Welcome to Server, from Bit");
15     if ((send(ns, buf, strlen(buf) + 1, 0)) == -1) {
16         perror("send");
17         exit(1);
18     }
19
20     if ((len=recv(ns, buf, strlen(buf), 0)) == -1) {
21         perror("recv");
22         exit(1);
23     }
24     printf("@@ [Bit] From Client: %s\n", buf);
25     close(ns);
26
27     return 0;
28 }
```

명령행 인자로 받은
소켓을 숫자로 변환

클라이언트에 메시지 전달

클라이언트의 응답 받기


```
<sys/types.h><sys/socket.h><unistd.h><netdb.h><netinet.h><arpa/inet
.h><stdlib.h><stdio.h><string.h>
34     printf("==> Connect Server\n");
35 if ((len = recv(sd, buf, sizeof(buf), 0)) == -1) {
36     perror("recv");
37     exit(1);
38 }
39 buf[len] = '\0';
40
41 printf("==> From Server : %s\n", buf);
42
43 strcpy(buf, "I want a TELNET Service.");
44 if (send(sd, buf, sizeof(buf) + 1, 0) == -1) {
45     perror("send");
46     exit(1);
47 }
48
49 close(sd);
50
51 return 0;
52 }
```

서버의 메시지 수신

서버에 메시지 전송



[예제 12-4] 실행결과

서버

```
# ex12_4s.out
** Create Socket
** Bind Socket
** Listen Socket
** Accept Client
** Fork Client
@@ [Bit] From Client: I want a TELNET Service.
```

클라이언트

```
# ex12_4c.out
==> Create Socket
==> Connect Server
==> From Server : Welcome to Server, from Bit
```



```
<sys/types.h><sys/socket.h><netinet.h><arpa/inet.h>
<stdlib.h><stdio.h><string.h>
09  #define PORTNUM 9005
10
11  int main(void) {
12      char buf[256];
13      struct sockaddr_in sin, cli;
14      int sd, clientlen = sizeof(cli);
15
16      if ((sd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
17          perror("socket");
18          exit(1);
19      }
20
21      memset((char *)&sin, '\0', sizeof(sin));
22      sin.sin_family = AF_INET;
23      sin.sin_port = htons(PORTNUM);
24      sin.sin_addr.s_addr = inet_addr("192.168.162.133");
25
26      if (bind(sd, (struct sockaddr *)&sin, sizeof(sin))) {
27          perror("bind");
28          exit(1);
29      }
```

포트번호

소켓 생성(데이터그램)

소켓 주소 구조체 생성

소켓기술자와 소켓 주소
구조체 연결

```
31     while (1) {
32         if ((recvfrom(sd, buf, 255, 0,
33             (struct sockaddr *)&cli, &clientlen)) == -1) {
34             perror("recvfrom");
35             exit(1);
36         }
37         printf("** From Client : %s\n", buf);
38         strcpy(buf, "Hello Client");
39         if ((sendto(sd, buf, strlen(buf)+1, 0,
40             (struct sockaddr *)&cli, sizeof(cli))) == -1) {
41             perror("sendto");
42             exit(1);
43         }
44     }
45
46     return 0;
47 }
```

클라이언트의 메시지 수신

클라이언트로 데이터 보내기



```

<sys/types.h><sys/socket.h><netinet/in.h><arpa/inet.h>
<stdlib.h><stdio.h><string.h>
09  #define PORTNUM 9005
10
11  int main(void) {
12      int sd, n;
13      char buf[256];
14      struct sockaddr_in sin;
15
16      if ((sd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
17          perror("socket");
18          exit(1);
19      }
20
21      memset((char *)&sin, '\0', sizeof(sin));
22      sin.sin_family = AF_INET;
23      sin.sin_port = htons(PORTNUM);
24      sin.sin_addr.s_addr = inet_addr("192.168.162.133");
25
26      strcpy(buf, "I am a client.");
27      if (sendto(sd, buf, strlen(buf)+1, 0,
28                (struct sockaddr *)&sin, sizeof(sin)) == -1) {
29          perror("sendto");
30          exit(1);
31      }

```

포트번호

소켓 생성

소켓 주소 구조체 생성

서버에 메시지 전송

```
33     n = recvfrom(sd, buf, 255, 0, NULL, NULL);
34     buf[n] = '\0';
35     printf("** From Server : %s\n", buf);
36
37     return 0;
38 }
```

서버가 보낸 데이터 읽기

```
# ex12_5s.out
** From Client : I am a client.
```

서버

```
# ex12_5c.out
** From Server : Hello Client
```

클라이언트





Thank You !

IT CookBook, 유닉스 시스템 프로그래밍