```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/time.h>
#include <sys/wait.h>

#define MSGSIZE 6
char *msg1 = "hello";
char *msg2 = "bye!";

void parent(int [][2]);
int child(int []);

void main()
{
        int pip[3][2];
        int i;

        for (i = 0; i < 3; i++) {
                if (pipe(pip[i]) == -1) {
                        perror("pipe call");
                        exit(1);
                }

                switch (fork()) {
                        case -1 :
                                perror("fork call");
                                exit(2);
                        case 0 :
                                child(pip[i]);
                }
        }
        parent(pip);

        exit(0);
}
```

```
void parent(int p[3][2])
{
        char buf[MSGSIZE], ch;
        fd_set set, master;
        int i;

        for (i = 0; i < 3; i++)
                close(p[i][1]);

        FD_ZERO(&master);
        FD_SET(0, &master);
        for (i = 0; i < 3; i++)
                FD_SET(p[i][0], &master);

        while (set = master, select(p[2][0] + 1, &set, NULL, NULL, NULL) > 0) {
                if (FD_ISSET(0, &set)) {
                        printf("From standard input ... ");
                        read(0, &ch, 1);
                        printf("%c\n", ch);
                }

                for (i = 0; i < 3; i++) {
                        if (FD_ISSET(p[i][0], &set)) {
                                if (read(p[i][0], buf, MSGSIZE) > 0) {
                                        printf("Message from pipe #%d\n", i);
                                        printf("MSG=%s\n", buf);
                                }
                        }
                }

                if (waitpid(-1, NULL, WNOHANG) == -1)
                        return;
        }
}
```

```
int child(int p[2])
{
        int count;

        close(p[0]);

        for (count = 0; count < 2; count++) {
                write(p[1], msg1, MSGSIZE);
                sleep(getpid() % 5);
        }

        write(p[1], msg2, MSGSIZE);
        exit(0);
}
```