```c
#include <stdio.h>
#include <unistd.h>
#include <string.h>
void syserr(char str[])
{
        perror(str);
        exit(1);
} /* sys_arr */
int main(int argc, char *argv[])
{
      int pfd[2],space=0;
    int pfd2[2];
        char msg[80];
        int status,i;
        char delims[] = " ";
        char *result = NULL;
    char *result2=NULL;
        if (pipe(pfd) == -1)
        syserr("pipe");
    if(pipe(pfd2)==-1)
            syserr("pipe2");
          gets(msg);
        //printf("asd:%s\n",msg);

                char Pipe[] = "|";
                char temp[80];
            char msg2[80],msg3[80];
                strcpy(temp,msg);
        char *argv3[3];
        char *argv2[3];
                result = strtok(msg, Pipe );
        result = strtok(msg,"|");
                i=0;
                while( result != NULL )
                {
            argv2[i]=result;
            i++;
                        result = strtok( NULL, "|");
                }

     switch(fork())
                        {
                                case -1:
                                        syserr("fork");
                                case 0:
                                        if (dup2(pfd[1], 1) == -1)
                                        syserr("dup2");
                                        if (close(pfd[0]) == -1 || close(pfd[1]) == -1)
                                                syserr("close");

                                        result2 = strtok(argv3[0]," ");
                                        int i;
                                        i=0;
                                        while( result2 != NULL )
                                        {
                                        argv2[i]=result2;
                                         i++;
                                         result2 = strtok(NULL, "" );
                                        }

                                         argv2[i]=NULL;
                                        execvp(argv2[0], argv2);

        } /* switch */

                    switch (fork())
                        {
                          case -1: // error

                          case 0:
               if (dup2(pfd[0], 0) == -1)
                                        syserr("dup2");

                                        if (close(pfd[0]) == -1 || close(pfd[1]) == -1)
                                                syserr("close");
            if (dup2(pfd2[1], 1) == -1)
                                        syserr("dup2");
                                        if (close(pfd2[0]) == -1 || close(pfd2[1]) == -1)
                                                syserr("close");


                        result2 = strtok(argv3[1]," ");
                        i=0;
                        while( result2 != NULL )
                        {
                                argv2[i]=result2;
                                i++;
                                 result2 = strtok(NULL, "" );
                        }
                        argv2[i]=NULL;
                                execvp(argv2[0], argv2);




                        }


                        switch (fork())
                        {
                          case -1: // error
                                perror("fork call");
                                exit(1);
                          case 0:
             if (dup2(pfd2[0], 0) == -1)
                                        syserr("dup2");
                                        if (close(pfd2[0]) == -1 || close(pfd2[1]) == -1)
                                        syserr("close");

                        result2 = strtok(argv3[2]," ");
                        i=0;
                        while( result2 != NULL )
                        {
                                argv2[i]=result2;
                                i++;
                                 result2 = strtok(NULL, "" );
                        }
                        argv2[i]=NULL;
                                execvp(argv2[0], argv2);



                        }


    if (close(pfd[0]) == -1 || close(pfd[1]) == -1)
        syserr("close");
        while (wait(NULL)>0);
}
```