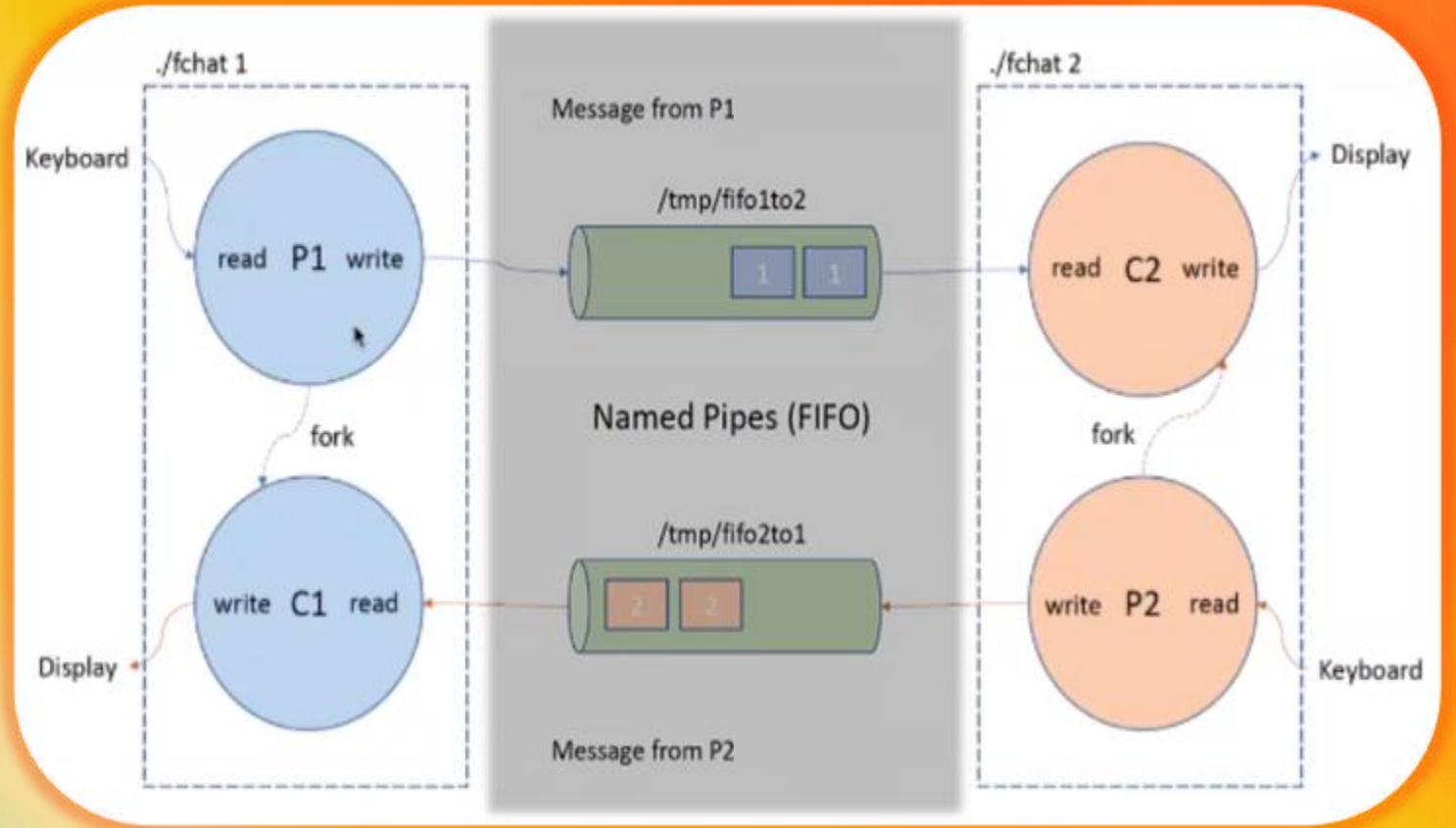


FIFO Chat



```
static void forceKill()
{
    wait();
    exit(EXIT_SUCCESS);
}
```

จะถูกเรียกโดย signal ใช้ในการแก้ zombie

```
FIFO_FD1 = open(FIFO_1,O_RDWR);
FIFO_FD2 = open(FIFO_2,O_RDWR);
argv++;
```

เปิดแบบ RDWR เพื่อให้เขียนและอ่านได้
โดยไม่ต้องเปิดใหม่

```

int main(int argc, char*argv[])
{
    int child;
    int nbytes;
    char rbuf[MAX_RBUF] = "";
    if(argc!=2)
    {
        fprintf(stderr, "Usage: %s <[ 1 2 ]>\n", *argv);
        exit(EXIT_FAILURE);
    }
    if(access(FIFO_1, F_OK) == -1)
    {
        FIFO_FD1 = mkfifo(FIFO_1, 0777);
        if(FIFO_FD1)
        {
            fprintf(stderr, "Could not create fifo %s\n", FIFO_1);
            exit(EXIT_FAILURE);
        }
    }

    if(access(FIFO_2, F_OK) == -1)
    {
        FIFO_FD2 = mkfifo(FIFO_2, 0777);
        if(FIFO_FD2)
        {
            fprintf(stderr, "Could not create fifo %s\n", FIFO_2);
            exit(EXIT_FAILURE);
        }
    }
}

```

ถ้าใส่ argument ไม่ถูก
จะจบโปรแกรม

เช็คว่ามี FIFO หรือไม่
ถ้าไม่มีให้สร้างใหม่
ถ้าสร้างไม่ได้ให้จบโปรแกรม

```

if(strcmp(*argv,"1")==0)
{
    child= fork();
    switch(child)
    {
        case -1: perror("Forking Failed");
                exit(EXIT_FAILURE);
        case 0 : while(strncmp(rbuf,"end chat",8))
                {
                    memset(rbuf,0,sizeof(rbuf));
                    nbytes = read(FIFO_FD2,rbuf,MAX_RBUF);
                    if (nbytes>1)
                    {
                        write(1,"User 2:", 7);
                        write(1, rbuf, nbytes);
                    }
                }
                kill(getppid(),SIGALRM);
                break;
    }
}

```

เช็คว่าเป็นแพท 1 หรือไม่

สร้าง process ใหม่

ทำจนกว่าจะเจอ"end chat"

ล้าง rbuf แล้วรับจาก FIFO_FD2

ถ้ามีค่าให้แสดงทางหน้าจอ

ส่งสัญญาณให้ parent เพื่อออกloop


```
default : while(strncmp(rbuf,"end chat",8))
{
    memset(rbuf,0,sizeof(rbuf));
    signal(SIGALRM,forceKill);
    nbytes = read(0,rbuf,MAX_RBUF);
    if (nbytes>1)
    {
        write(FIFO_FD1,rbuf, nbytes);
    }
}
kill(child,SIGKILL);
wait();
}
}
```

ถ้ามี SIGALRM จะให้ทำforceKill

ถ้ามีค่าให้เขียนลง FIFO_FD1

เมื่อออกกลุ่มให้ส่ง SIGKILL ฆ่าลูก
แล้วรอจนลูกตายค่อยไปต่อ

```

else if(strcmp(*argv,"2")==0)
{
    child= fork();
    switch(child)
    {
        case -1: perror("Forking Failed");
                exit(EXIT_FAILURE);
        case 0 : while(strncmp(rbuf,"end chat",8))
                {
                    memset(rbuf,0,sizeof(rbuf));
                    nbytes = read(FIFO_FD1,rbuf,MAX_RBUF);
                    if (nbytes>1)
                    {
                        write(1,"User 1:", 7);
                        write(1,rbuf, nbytes);
                    }
                }
                kill(getppid(),SIGALRM);
                break;
        default : while(strncmp(rbuf,"end chat",8))
                {
                    memset(rbuf,0,sizeof(rbuf));
                    signal(SIGALRM,forceKill);
                    nbytes = read(0,rbuf,MAX_RBUF);
                    if (nbytes>1)
                    {
                        write(FIFO_FD2,rbuf, nbytes);
                    }
                }
                kill(child,SIGKILL);
                wait();
    }
}

```

ทำงานเหมือนกับแชท 1 แต่
ให้ลูกอ่านจาก FIFO_FD1

ให้พ่อแม่เขียนลง
FIFO_FD2

```

if(FIFO_FD1 != -1 )
    close(FIFO_FD1);
if(FIFO_FD2 != -1 )
    close(FIFO_FD2);

unlink(FIFO_1);
unlink(FIFO_2);

```

ปิดทุกอย่าง