

47.3. A NAMED PIPE (OFTEN REFERRED TO AS A FIFO) IS AN OLD UNIX IPC MECHANISM FOR PROCESSES COMMUNICATING ON THE SAME MACHINE. IT WORKS JUST LIKE A REGULAR, CONNECTED ANONYMOUS PIPES, EXCEPT THAT THE PROCESSES RENDEZVOUS USING A FILENAME AND DON'T HAVE TO BE RELATED.

CHAPTER 47. PERLIPC

```
#!/usr/bin/perl -w
use strict;
use Socket;
my ($remote,$port, $iaddr, $paddr, $proto, $line);

$remote = shift || 'localhost';
$port    = shift || 2345; # random port
if ($port =~ /\D/) { $port = getservbyname($port, 'tcp') }
die "No port" unless $port;
$iaddr   = inet_aton($remote)           || die "no host: $remote";
$paddr   = sockaddr_in($port, $iaddr);

$proto   = getprotobyname('tcp');
socket(SOCK, PF_INET, SOCK_STREAM, $proto) || die "socket: $!";
connect(SOCK, $paddr) || die "connect: $!";
while (defined($line = <SOCK>)) {
    print $line;
}

close (SOCK)           || die "close: $!";
exit;
```

And here's a corresponding server to go along with it. We'll leave the address as INADDR_ANY so that the kernel can choose the appropriate interface on multihomed hosts. If you want sit on a particular interface (like the external side of a gateway or firewall machine), you should fill this in with your real address instead.

```
#!/usr/bin/perl -Tw
use strict;
BEGIN { $ENV{PATH} = '/usr/ucb:/bin' }
use Socket;
use Carp;
my $EOL = "\015\012";

sub logmsg { print "$0 $$: @_ at ", scalar localtime, "\n" }

my $port = shift || 2345;
my $proto = getprotobyname('tcp');

($port) = $port =~ /^(\d+)/ or die "invalid port";

socket(Server, PF_INET, SOCK_STREAM, $proto) || die "socket: $!";
setsockopt(Server, SOL_SOCKET, SO_REUSEADDR,
            pack("l", 1)) || die "setsockopt: $!";
bind(Server, sockaddr_in($port, INADDR_ANY)) || die "bind: $!";
listen(Server, SOMAXCONN) || die "listen: $!";

logmsg "server started on port $port";

my $paddr;

$SIG{CHLD} = \&REAPER;

for ( ; $paddr = accept(Client,Server); close Client) {
    my($port,$iaddr) = sockaddr_in($paddr);
    my $name = gethostbyaddr($iaddr,AF_INET);
```