The figures below show the expected message format, contents and order.
All multi-byte values (for example fd, errno, count etc.) are written in big endian byte order.
Byte streams like strings and read/write data should be written in its original order.

---

open

message

1 byte	2 bytes	n bytes	4 bytes	4 bytes
opcode	length n	path name string	flags	mode

response

4 bytes	4 bytes
return value	errno

---

close

message

1 byte	4 bytes
opcode	fd

response

4 bytes	4 bytes
return value	errno

---

read

message

1 byte	4 bytes	4 bytes
opcode	fd	count

response

4 bytes	4 bytes	n bytes (note that n is given in the return value)
return value	errno	data read

---

write

message

1 byte	4 bytes	4 bytes	count bytes
opcode	fd	count	data to write

//note that count comes before the data so we know how much data there is

response

4 bytes	4 bytes
return value	errno

---

lseek

message

1 byte	4 bytes	4 bytes	4 bytes
opcode	fd	offset	whence

response

4 bytes	4 bytes
return value	errno

---

pipe

message

1 byte
opcode

response

4 bytes	4 bytes	4 bytes	4 bytes
return value	errno	pipefd[0]	pipefd[1]

---

dup2

message

1 byte	4 bytes	4 bytes
opcode	oldfd	newfd

response

4 bytes	4 bytes
return value	errno