

Advanced Programming in the UNIX Environment

Week 08, Segment 2: System V IPC

**Department of Computer Science
Stevens Institute of Technology**

Jan Schaumann

`jschauma@stevens.edu`

`https://stevens.netmeister.org/631/`

System V IPC

There are three types of *asynchronous* IPC originating from System V:

- Semaphores
- Shared Memory
- Message Queues

All three use IPC structures, referred to by an *identifier* and a *key*; all three are (necessarily) limited to communication between processes on one and the same host.

Since these structures are not known by name, special system calls (`msgget(2)`, `semop(2)`, `shmat(2)`, etc.) and special userland commands (`ipcrm(1)`, `ipcs(1)`, etc.) are necessary.

System V IPC: Semaphores

A semaphore is a counter used to provide access to a shared data object for multiple processes. To obtain a shared resource a process needs to do the following:

1. Test semaphore that controls the resource.
2. If value of semaphore > 0, decrement semaphore and use resource; increment semaphore when done.
3. If value == 0 sleep until value > 0

Semaphores are obtained using `semget(2)`, properties controlled using `semctl(2)`, operations on a semaphore performed using `semop(2)`.

System V IPC: Semaphores

The screenshot shows two terminal sessions side-by-side. The left session shows the compilation and execution of a C program named semdemo.c, which demonstrates semaphore operations. The right session shows the execution of the resulting binary, a.out, which performs similar operations.

Left Terminal Session:

```
jschauma@apue$ cc -Wall -Werror -Wextra semdemo.c
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
Unlocked.
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock: █
```

Right Terminal Session:

```
Terminal — 129x24
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
```

System V IPC: Semaphores

The screenshot shows a terminal window titled "Terminal — 129x24" with two panes. The left pane shows the compilation and execution of a C program named semdemo.c. The right pane shows the output of the program, which involves locking and unlocking a semaphore.

```
jschauma@apue$ cc -Wall -Werror -Wextra semdemo.c
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
Unlocked.
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
Unlocked.
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
```

Terminal — 129x24

```
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
```

System V IPC: Semaphores

```
Terminal — 129x24

Unlocked.
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
Unlocked.
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
Unlocked.
jschauma@apue$ ipcs -s
IPC status from <running system> as of Mon Oct 28 09:40:49 2024
Semaphores:
T      ID      KEY      MODE      OWNER      GROUP
s    65536  704770294 --rw-rw-rw- jschauma  users

jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
Unlocked.
jschauma@apue$ sudo su - fred
fred@apue$ whoami
fred
fred@apue$ cd ~jschauma/08
fred@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
```

System V IPC: Semaphores

```
Terminal — 129x24

Locked.
Press return to unlock:
Unlocked.
jschauma@apue$ ipcs -s
IPC status from <running system> as of Mon Oct 28 09:40:49 2024
Semaphores:
T      ID      KEY      MODE      OWNER      GROUP
s    65536  704770294 --rw-rw-rw- jschauma    users

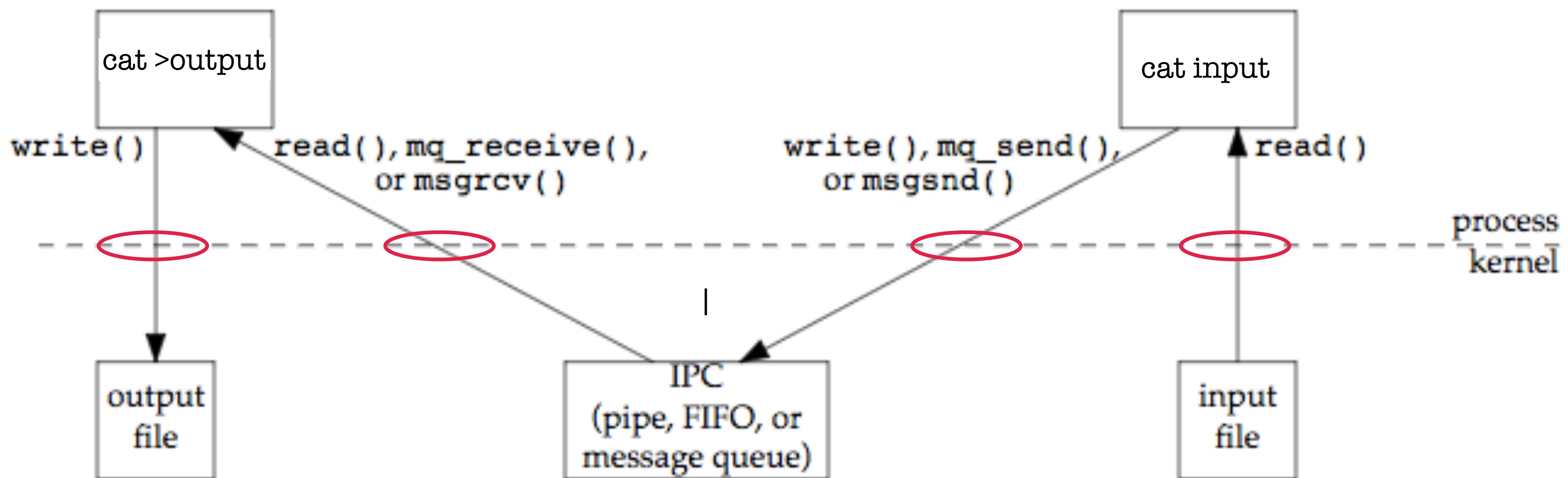
jschauma@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
Unlocked.
jschauma@apue$ ipcrm -s 65536
jschauma@apue$ ipcs -s
IPC status from <running system> as of Mon Oct 28 10:35:35 2024
Semaphores:
T      ID      KEY      MODE      OWNER      GROUP
s    65536  704770294 --rw-rw-rw- jschauma    users

fred@apue$ ./a.out
Trying to lock...
Locked.
Press return to unlock:
Unlocked.
jschauma@apue$ sudo su - fred
fred@apue$ whoami
fred
fred@apue$ cd ~jschauma/08
fred@apue$ ./a.out
Press return to lock:
Trying to lock...
Locked.
Press return to unlock:
Unlocked.
fred@apue$ ipcs -s
IPC status from <running system> as of Mon Oct 28 09:42:09 2024
Semaphores:
T      ID      KEY      MODE      OWNER      GROUP
s    65536  704770294 --rw-rw-rw- jschauma    users

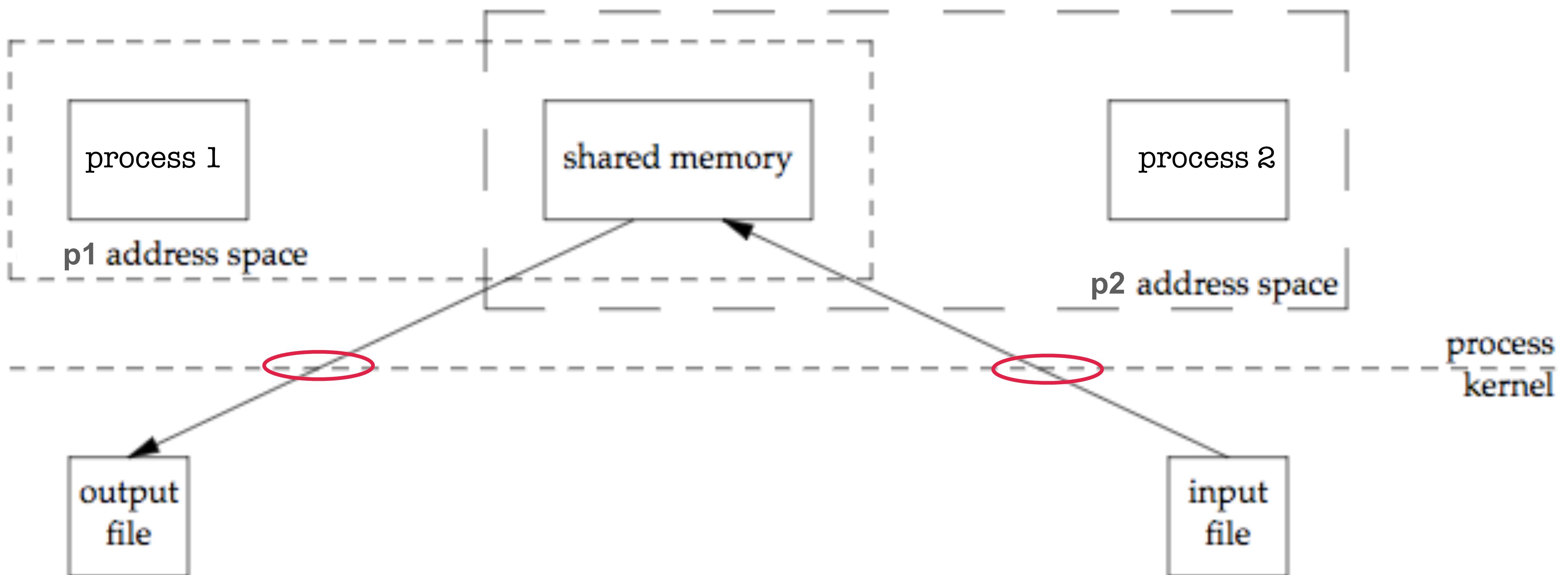
fred@apue$ ipcrm -s 65536
ipcrm: semid(65536): : Operation not permitted
fred@apue$
```

IPC Data Flow

```
$ cat input | cat >output
```



System V IPC: Shared Memory



System V IPC: Shared Memory

- fastest form of IPC
- access to shared region of memory often controlled using semaphores
- obtain a shared memory identifier using `shmget(2)`
- attach shared memory segment to a processes address space by calling `shmat(2)`
- detach it using `shmdt(2)`
- catch-all for shared memory operations: `shmctl(2)`

System V IPC: Shared Memory

```
Terminal — 129x24 [2/61]

8310: writing to segment: "The cow says 'moo'." jschauma@apue$ ipcs -ma
IPC status from <running system> as of Mon Oct 28 10:38:51 2024

Shared Memory:
T      ID      KEY      MODE      OWNER      GROUP      CREATOR      CGROUP      NATTCH      SEGSZ      CPID      LPID      ATIME      DTIME      CTIME
m    65536  704770300 --rw-rw-rw- jschauma    users      jschauma    users          0      1024      8310      8310 10:38:43 10:38:43 10:38:43

jschauma@apue$ ./a.out
7337: segment contains: "The cow says 'moo'." jschauma@apue$ ./a.out
5682: segment contains: "The cow says 'moo'."

8438: segment contains: "The cow says 'moo'." fred@apue$ ./a.out "The pig says 'oink'." 8228: writing to segment: "The pig says 'oink'." fred@apue$ ipcs -ma
IPC status from <running system> as of Mon Oct 28 10:41:25 2024

Shared Memory:
T      ID      KEY      MODE      OWNER      GROUP      CREATOR      CGROUP      NATTCH      SEGSZ      CPID      LPID      ATIME      DTIME      CTIME
m    65536  704770300 --rw-rw-rw- jschauma    users      jschauma    users          0      1024      8310      8228 10:40:41 10:40:41 10:38:43

fred@apue$
```

Terminal — 81x44

```
[jschauma@apue ./a.out
High address (args and env):
-----
envp[16] at : 0x7FFF2279D8
environ[16] at : 0x7FFF2279D8
envp[0] at : 0x7FFF227958
environ[0] at : 0x7FFF227958
last arg at : 0x7FFF227950
first arg at : 0x7FFF227948

Stack:
-----
First variable inside main at : 0x7FFF2278DC
func_array[] ends at : 0x7FFF2278D0
func_array[] (like 'array[]', but on stack) begins at : 0x7FFF2278C0
argc at : 0x7FFF2278BC
argv at : 0x7FFF2278B0
envp at : 0x7FFF2278A8
func2 (from main): frame at : 0x7FFF22788C
func frame at : 0x7FFF227884
static int n within func at : 0x 601EEC
func (called 1 times): frame at : 0x7FFF227884
func2 (from func): frame at : 0x7FFF22785C

Shared Memory:
-----
shared memory area ends at : 0x7FF7EFE6A0
shared memory area begins at : 0x7FF7EE6000

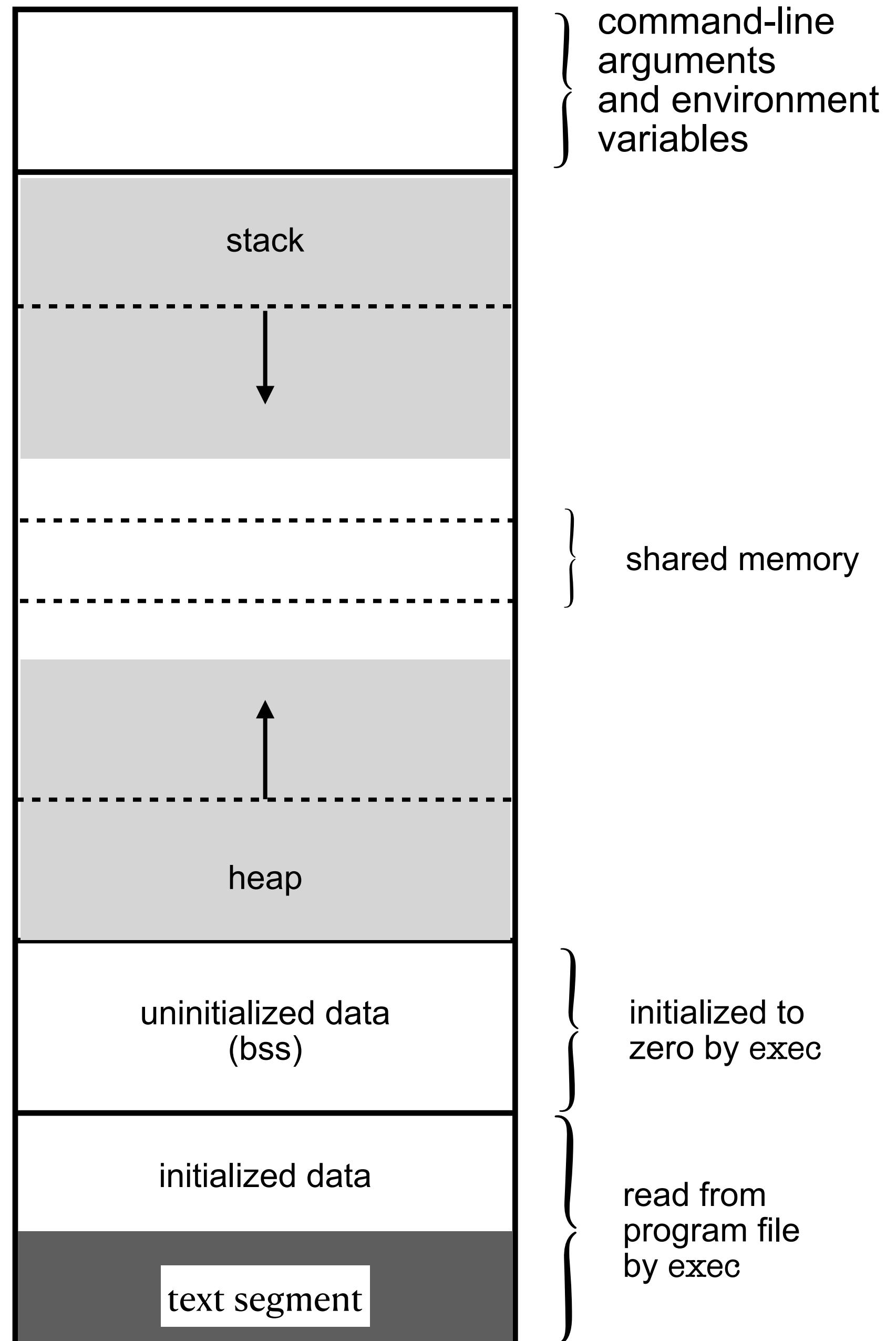
Heap:
-----
malloced area ends at : 0x7926986E5020
malloced area begins at : 0x7926986E5000

Uninitialized Data (BSS):
-----
array[] ends at : 0x 601F70
array[] (uninitialized, fixed-size char * on BSS) from : 0x 601F60
num2 (uninitialized global int) at : 0x 601F58
string2 (uninitialized global char *) at : 0x 601F50
extern **environ at : 0x 601F40

Initialized Data:
-----
```

high address

low address



System V IPC: Message Queues

- linked list of messages stored in kernel space
- create or open existing queue using `msgget(2)`
- add message at end of queue using `msgsnd(2)`
- receive messages from queue using `msgrcv(2)`
- control queue properties using `msgctl(2)`

The message itself is contained in a user-defined structure such as

```
struct mymsg {  
    long mtype;          /* message type */  
    char mtext[512];    /* body of message */  
};
```

System V IPC: Message Queues

```
Terminal — 129x24 [4/415] E

jschauma@apue$ cc -Wall -Werror -Wextra -o send msgsend.c
jschauma@apue$ cc -Wall -Werror -Wextra -o recv msgrecv.c
jschauma@apue$ ./send 1 "Hello!"
jschauma@apue$ ipcs -qa
IPC status from <running system> as of Mon Oct 28 10:52:48 2024

Message Queues:
T      ID      KEY      MODE      OWNER      GROUP    CREATOR      CGROUP   CBYTES   QNUM   QBYTES   LSPID   LRVID   STIME   RTIME   CTIME
q    65536          1 --rw-r--r-- jschauma  users    jschauma  users        7         1     2048    6613      0 10:52:44 no-entry 10:52:44

jschauma@apue$ ./send 1 "How are things?"
jschauma@apue$ ipcs -qa
IPC status from <running system> as of Mon Oct 28 10:53:29 2024

Message Queues:
T      ID      KEY      MODE      OWNER      GROUP    CREATOR      CGROUP   CBYTES   QNUM   QBYTES   LSPID   LRVID   STIME   RTIME   CTIME
q    65536          1 --rw-r--r-- jschauma  users    jschauma  users       23         2     2048    4048      0 10:53:20 no-entry 10:52:44

jschauma@apue$ ./recv 1
Hello!
jschauma@apue$ ipcs -qa
IPC status from <running system> as of Mon Oct 28 10:53:48 2024

Message Queues:
```

System V IPC: Message Queues

```
Terminal — 129x24
T      ID      KEY      MODE      OWNER      GROUP      CREATOR      CGROUP      CBYTES      QNUM      QBYTES      LSPID      LRVID      STIME      RTIME      CTIME
q      65536          1 --rw-r--r-- jschauma    users      jschauma    users           16          1        2048      4048      5681 10:53:20 10:53:46 10:52:44

jschauma@apue$ ipcs -qa
IPC status from <running system> as of Mon Oct 28 10:54:48 2024

Message Queues:
T      ID      KEY      MODE      OWNER      GROUP      CREATOR      CGROUP      CBYTES      QNUM      QBYTES      LSPID      LRVID      STIME      RTIME      CTIME
q      65536          1 --rw-r--r-- jschauma    users      jschauma    users           0          0        2048      4048      8618 10:53:20 10:54:29 10:52:44

jschauma@apue$ ./recv 1

jschauma@apue$ sudo su - fred
fred@apue$ cd ~jschauma/08
fred@apue$ ./recv 1
How are things?
fred@apue$ ./send 1 "Fred saves you!"
msgget: Permission denied
fred@apue$ ./recv 1
```

System V IPC: Message Queues

```
Terminal — 129x24
q      65536          1 --rw-r--r-- jschauma    users jschauma    users      16     1   2048   4048   5681 10:53:20 10:53:46 10:52:44
jschauma@apue$ ipcs -qa
IPC status from <running system> as of Mon Oct 28 10:54:48 2024

Message Queues:
T      ID      KEY      MODE      OWNER      GROUP      CREATOR      CGROUP      CBYTES      QNUM      QBYTES      LSPID      LRVID      STIME      RTIME      CTIME
q      65536          1 --rw-r--r-- jschauma    users jschauma    users      0          0   2048   4048   8618 10:53:20 10:54:29 10:52:44

jschauma@apue$ ./recv 1
Who will get this?
jschauma@apue$
```

```
jschauma@apue$ sudo su - fred
fred@apue$ cd ~jschauma/08
fred@apue$ ./recv 1
How are things?
fred@apue$ ./send 1 "Fred saves you!"
msgget: Permission denied
fred@apue$ ./recv 1
Unblock thyself.
fred@apue$
```

```
jschauma@apue$ ./send 1 "Who will get this?"
jschauma@apue$ ./send 1 "Unblock thyself."
jschauma@apue$
```

System V IPC: Message Queues

```
Terminal — 129x24
q      65536          1 --rw-r--r-- jschauma    users jschauma    users      0      0 2048 4048 8618 10:53:20 10:54:29 10:52:44
jschauma@apue$ ./recv 1
Who will get this?
jschauma@apue$ ipcs -qa
IPC status from <running system> as of Mon Oct 28 10:57:03 2024

Message Queues:
T      ID      KEY      MODE      OWNER      GROUP      CREATOR      CGROUP      CBYTES      QNUM      QBYTES      LSPID      LRVID      STIME      RTIME      CTIME
q      65536          1 --rw-r--r-- jschauma    users jschauma    users      0      0 2048 8756 4436 10:56:37 10:56:37 10:52:44
jschauma@apue$



jschauma@apue$ sudo su - fred
fred@apue$ cd ~jschauma/08
fred@apue$ ./recv 1
How are things?
fred@apue$ ./send 1 "Fred saves you!"
msgget: Permission denied
fred@apue$ ./recv 1
Unblock thyself.
fred@apue$



jschauma@apue$ ./send 1 "Who will get this?"
jschauma@apue$ ./send 1 "Unblock thyself."
jschauma@apue$ ./recv 2
msgget: No such file or directory
jschauma@apue$ ipcrm -q 65536
jschauma@apue$
```

POSIX Message Queues

`mq(3)` provides a real-time IPC interface similar to System V message queues.
Notably:

- message queues are identified by a named identifier (no `ftok(3)` needed)
- message queues may or may not be exposed in the filesystem (e.g., `/dev/mqueue`)
- `mq_send(3)` and `mq_receive(3)` allow both blocking and non-blocking calls
- `mq_send(3)` lets you specify a priority; equal priority messages are queued as a FIFO, but higher priority messages are inserted before those of a lower priority
- `mq(3)` provides an asynchronous notification mechanism: `mq_notify(3)`

POSIX Message Queues

The screenshot shows a terminal window with three distinct sessions. The first session (leftmost) displays the contents of a message queue. The second session (middle) shows the process of sending messages to the queue. The third session (rightmost) shows the queue after multiple sends.

```
Terminal — 130x24

Number of messages in queue: 1
Message of priority 1: TUNA
Message of priority 0: avocado
Message of priority 0: onion
Message of priority 0: tomato

Number of messages in queue: 1
Message of priority 0: avocado
Number of messages in queue: 1
Message of priority 0: onion
Number of messages in queue: 1
Message of priority 0: tomato
Number of messages in queue: 1
Message of priority 1: TUNA

Number of messages in queue: 1
Message of priority 0: avocado
Number of messages in queue: 1
Message of priority 0: onion
Number of messages in queue: 1
Message of priority 0: tomato
Number of messages in queue: 1
Message of priority 1: TUNA

jschauma@apue$ ./mqsend avocado onion tomato
jschauma@apue$ ./mqsend -w avocado onion tomato
jschauma@apue$ ./mqsend avocado onion tomato
jschauma@apue$ ./mqsend avocado onion tomato
jschauma@apue$
```

0 sh 1 sh

System V IPC

- *asynchronous* IPC between processes on the same system
- old, but not obsolete
- semaphores are useful to guard access to shared resources
- shared memory allows for fast IPC
- message queues as a service are a popular way to implement "pub sub" models
 - Amazon Simple Queue Service
 - Apache Kafka
 - Java Message Service
 - RabbitMQ
 - ...