Interprocess Communications

Franco Maria Nardini

Pipe

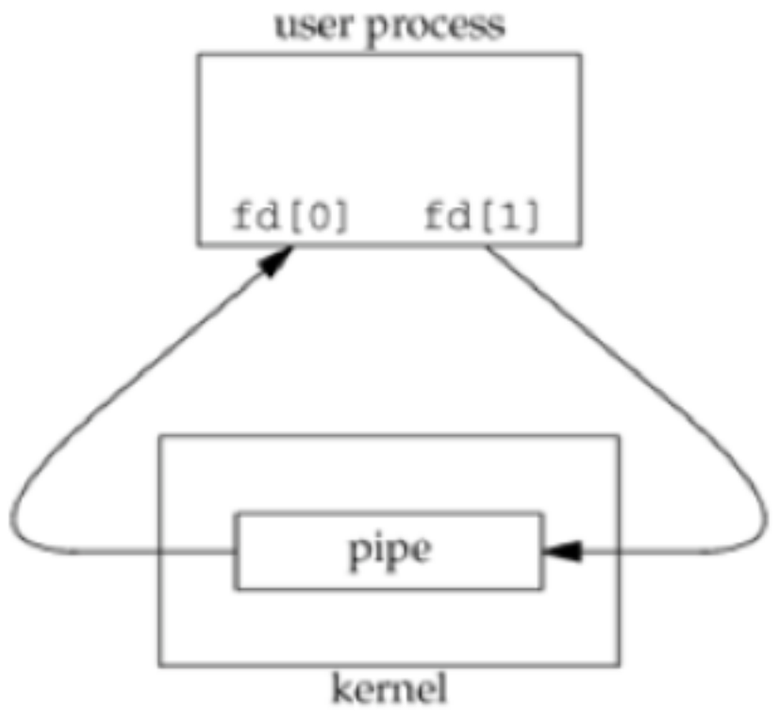
```
cat pippo.txt | less

sort pluto.txt | uniq -c | less
```

- è il metodo più datato di IPC in UNIX
- half-duplex
  - POSIX.1 consente full-duplex
  - half-duplex sempre per massima portabilità
- può esser usata tra processi che hanno ancestor comune

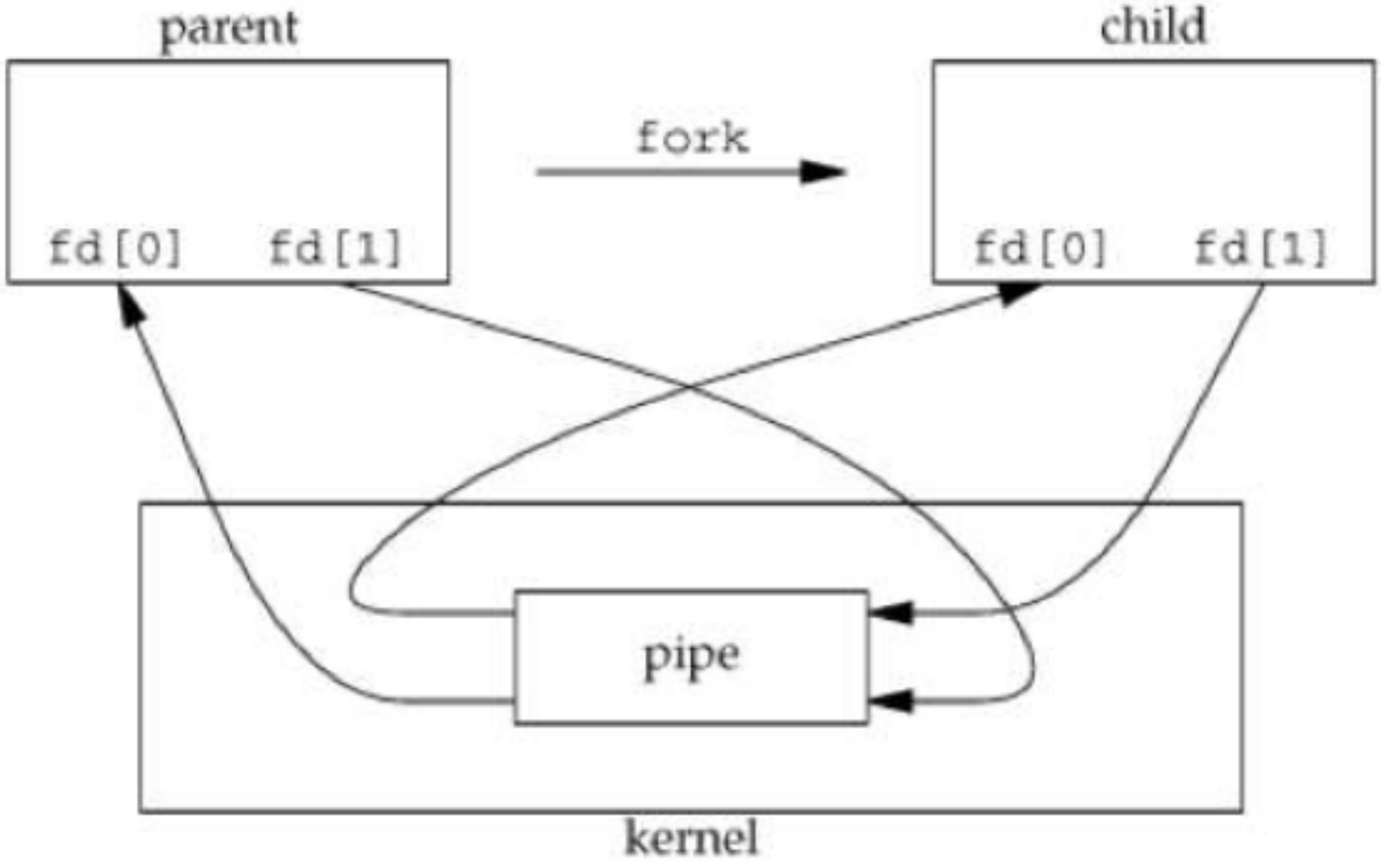
pipe(2)

```
#include <unistd.h>

int pipe(int filedes[2]);
```

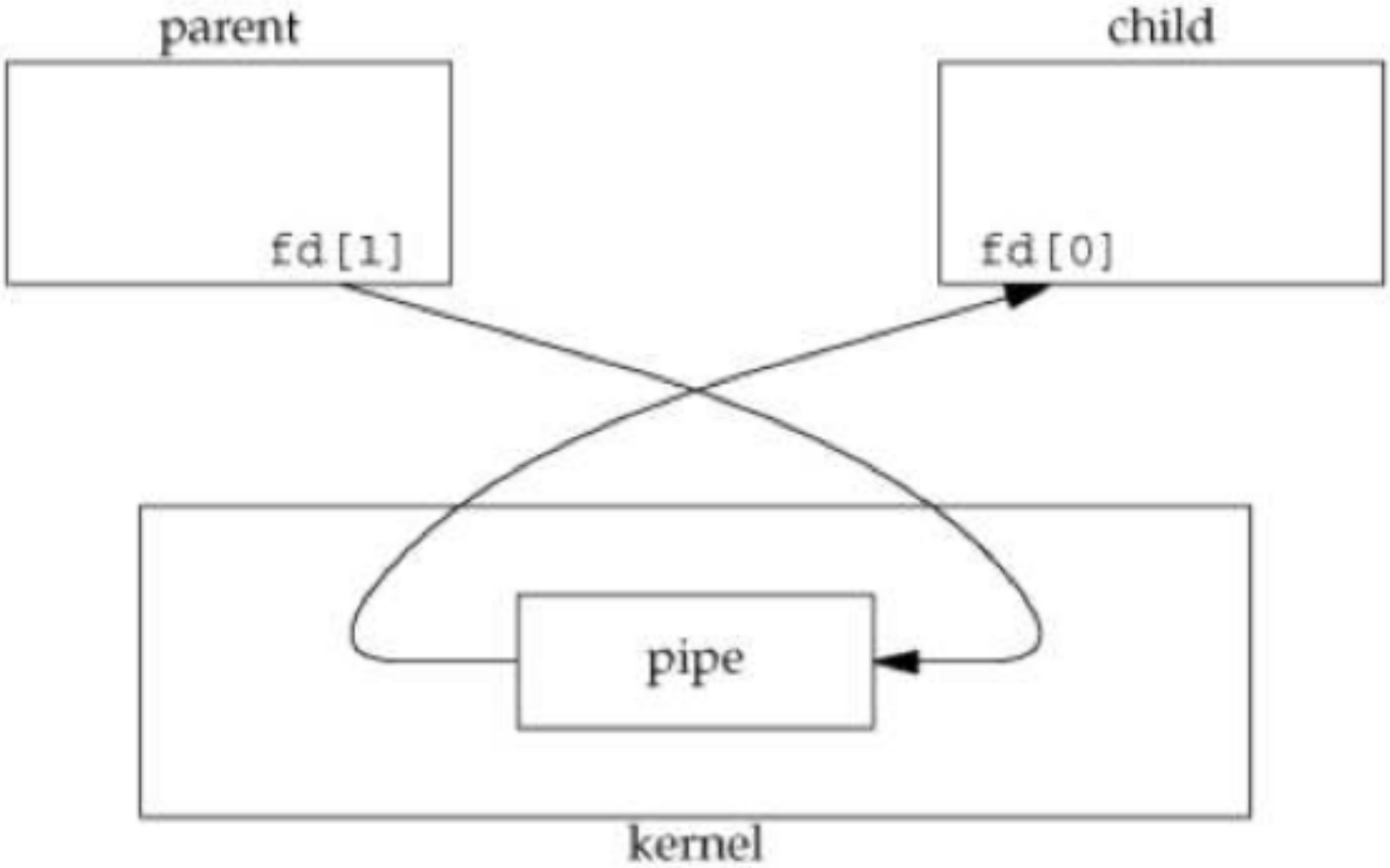
Returns: 0 if OK, -1 otherwise



Pipe



Pipe



Pipe

- Dopo le chiusure:
  - **`read(2)`** da un pipe il cui *fd[1]* è stato chiuso: ritorna 0 dopo aver letto tutti i dati
  - **`write(2)`** su un pipe con *fd[0]* chiuso:
    - **`write(2)`** ritorna -1
- Quando si scrive su pipe, la costante **`PIPE_BUF`** specifica la dimensione del buffer del kernel associato al pipe

Esempio

```
$ wget http://hpc.isti.cnr.it/~nardini/siselab/10/pipe1.c

$ cc -Wall ./pipe1.c

$ ./a.out
```

XSI IPC

- Tre tipi di IPC introdotti da System V:
  - semafori
  - shared memory
  - code di messaggi
- Comunicazioni tra processi su stesso host
  - Tutte consentono comunicazione *asincrona*

XSI IPC

- Ogni struttura IPC è identificata mediante un *identificatore*
  - intero non negativo che il kernel usa per riferire la risorsa
  - diversamente dai *file descriptors* (**?**), non sono piccoli
    - crescono e tornano all'inizio (limite inferiore), quando raggiunto il MAX_INT
- Serie di comandi e system calls per interagire
  - `msgget(2)`, `semop(2)`
  - `ipcs(1)`

XSI IPC: Semafori

- Un semaforo è:
    - un contatore per consentire il corretto accesso ad una risorsa condivisa da parte di processi multipli
- Per ottenere la risorsa, un processo deve:
    - testare il semaforo che controlla la risorsa
    - se valore > 0, decremento il semaforo e uso la risorsa. al termine, incremento di nuovo il valore del semaforo.
    - se valore == 0, aspetto finché valore > 0
- I semafori sono ottenuti con **`semget(2)`**, controllati con **`semctl(2)`**. Le operazioni sono fatte con **`semop(2)`**.

Example

```
$ wget http://hpc.isti.cnr.it/~nardini/siselab/11/
semdemo.c

$ cc -Wall ./semdemo.c

$ ./a.out (prima shell)

$ ./a.out (seconda shell)

$ ipcs -s

$ ipcrm -s semID (rimuove il semaforo)
```

XSI IPC: Shared Memory

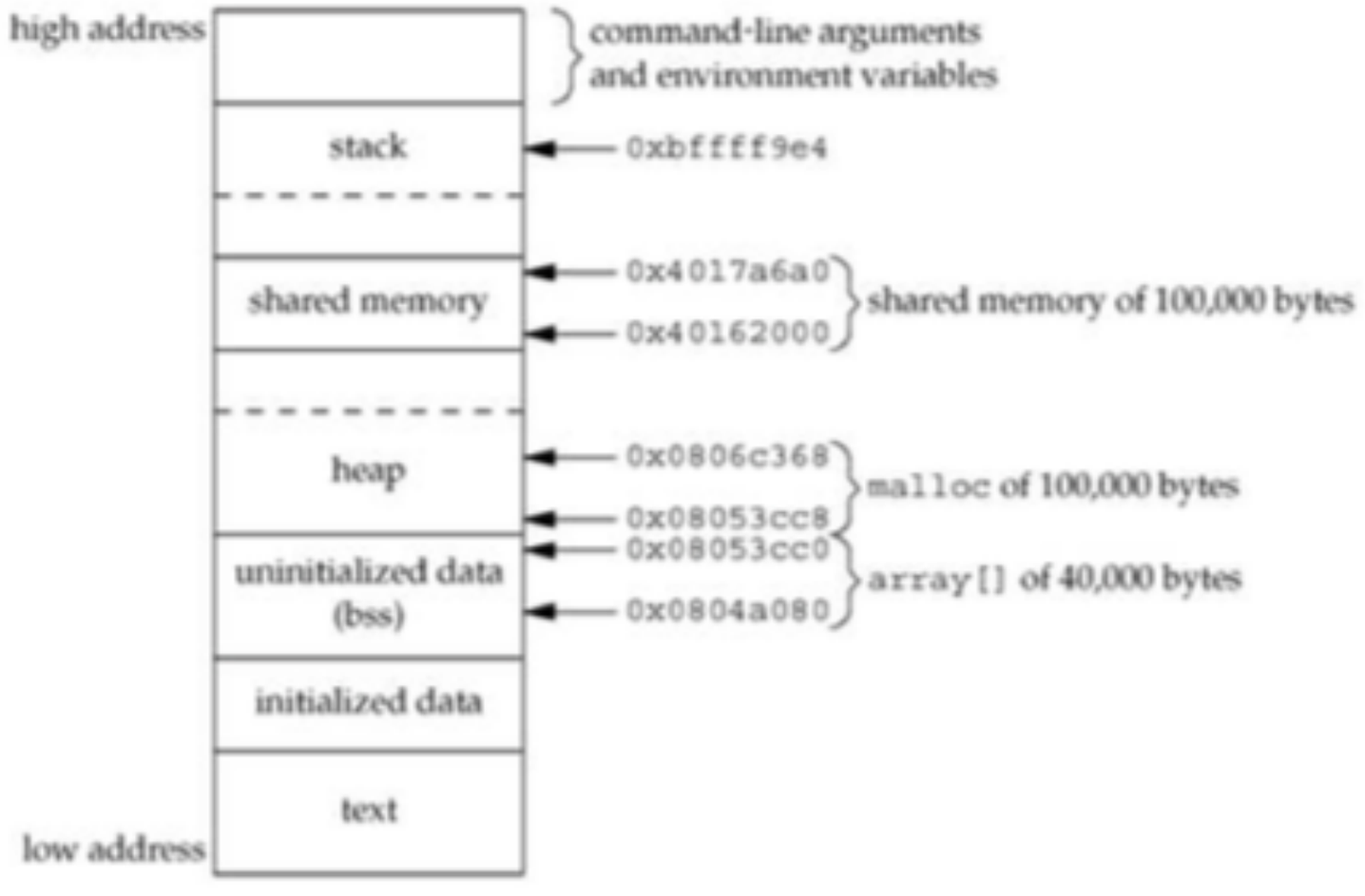
- Metodo più veloce di IPC
- Consiste nell'accedere ad una memoria condivisa a volte controllata con semafori
  - ottenimento di un identificatore di memoria condivisa con **`shmget(2)`**
  - collegamento del segmento condiviso allo spazio degli indirizzi di un processo con **`shmat(2)`**
  - scollegamento con **`shmdt(2)`**

Esempio

```
$ wget http://hpc.isti.cnr.it/~nardini/siselab/11/
shmdemo.c

$ cc -Wall ./shmdemo.c

$ ./a.out Ciao!

$ ipcs
```

Esempio

```
$ wget http://hpc.isti.cnr.it/~nardini/siselab/11/memory-layout.c

$ cc -Wall ./memory-layout.c

$ ./a.out
```



XSI IPC: Message Queues

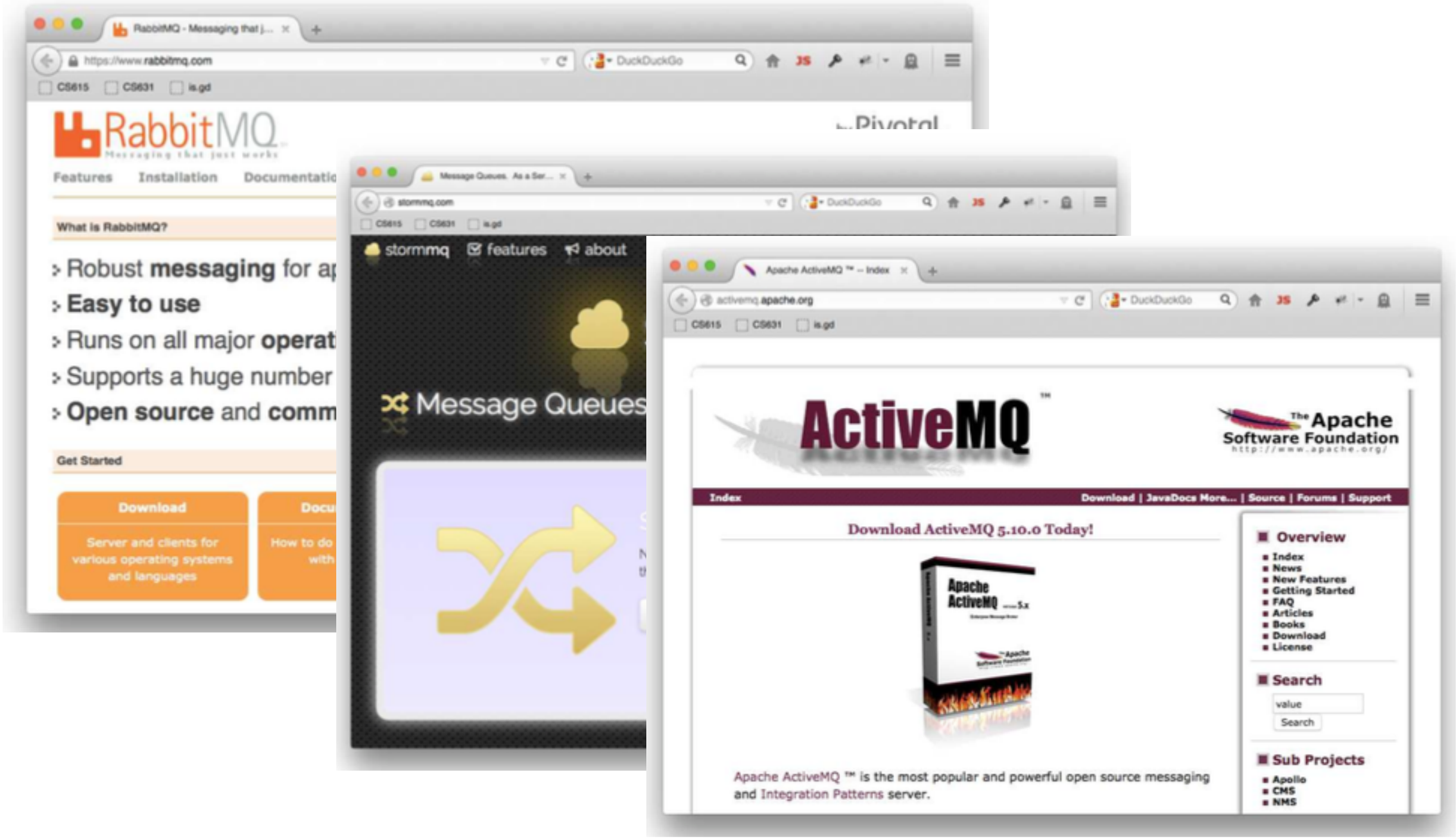
- Liste linkate (**?**) di messaggi memorizzate dal kernel
- Creare e aprire una coda con **`msgget(2)`**
- Aggiungere un messaggio a fine coda con **`msgsnd(2)`**
- Controllo della coda con **`msgctl(2)`**
- Ricezione messaggi dalla coda **`msgrcv(2)`**

XSI IPC: Message Queues

- Il messaggio ha una struttura definita dall'utente

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

Altre Code


RabbitMQ - Messaging that j...
https://www.rabbitmq.com
DuckDuckGo
CS615
CS631
is.gd
RabbitMQ
Messaging that just works
Features
Installation
What is RabbitMQ?
Robust messaging for ap
Easy to use
Runs on all major operat
Supports a huge number
Open source and comm
Get Started
Download
Server and clients for various operating systems and languages
Message Queues. As a Ser...
stormmq.com
stormmq
features
about
Message Queues
Apache ActiveMQ ™ – Index
activemq.apache.org
ActiveMQ
The Apache Software Foundation
http://www.apache.org/
Index
Download | JavaDocs More... | Source | Forums | Support
Download ActiveMQ 5.10.0 Today!
Apache ActiveMQ ™ is the most popular and powerful open source messaging and Integration Patterns server.
Overview
Index
News
New Features
Getting Started
FAQ
Articles
Books
Download
License
Search
value
Search
Sub Projects
Apollo
CMS
NMS

Sockets

```
#include <sys/socket.h>

int socket(int domain, int type, int protocol);
```

Domain	Description
PF_LOCAL	local (previously UNIX) domain protocols
PF_INET	ARPA Internet protocols
PF_INET6	ARPA IPv6 (Internet Protocol version 6) protocols
PF_ARP	RFC 826 Ethernet Address Resolution Protocol
...	...

Type	Description
SOCK_STREAM	sequenced, reliable, two-way connection based byte streams
SOCK_DGRAM	connectionless, unreliable messages of a fixed (typically small) maximum length
SOCK_RAW	access to internal network protocols and interfaces
...	...

UNIX/LOCAL domain

- si crea un socket usando **`socket(2)`**
- ci si “attacca” al socket usando **`bind(2)`**
- **`bind`**are un nome in dominio UNIX, corrisponde a creare un file sul file system
- entrambe i processi devono conoscere e concordare il nome del socket
- i file servono per il rendezvous, non per il message delivery quando la connessione è stabilita
- il socket deve esser rimosso con **`unlink(2)`**

Esempio

```
$ wget http://hpc.isti.cnr.it/~nardini/siselab/11/
udgramsend.c
```

```
$ wget http://hpc.isti.cnr.it/~nardini/siselab/11/
udgramread.c
```

```
$ cc -Wall ./udgramsend.c -o send
```

```
$ cc -Wall ./udgramread.c -o read
```

Esempio

```
1$ ./read
```

```
2$ ls -l
```

```
2$ ./send socket
```

```
1$ (?)
```

Internet Domain

- utilizzano Internet! :)
- diversamente dai socket visti prima, i nomi di socket Internet non passano dal file system
  - non c'e' bisogno di unlink dopo la chiusura
- indirizzo può essere un qualsiasi indirizzo di rete valido
- porte riservate a root: 1-1023
- richiesta di una specifica porta con **`bind(2)`**
- determinare la porta usata con **`getsocketname()`**

Esempio

```
$ wget http://hpc.isti.cnr.it/~nardini/siselab/11/
dgramsend.c

$ wget http://hpc.isti.cnr.it/~nardini/siselab/11/
dgramread.c

$ cc -Wall ./dgramsend.c -o send

$ cc -Wall ./dgramread.c -o read
```

Esempio

```
1$ ./read

2$ ./send localhost #porta

1$ (?)
```