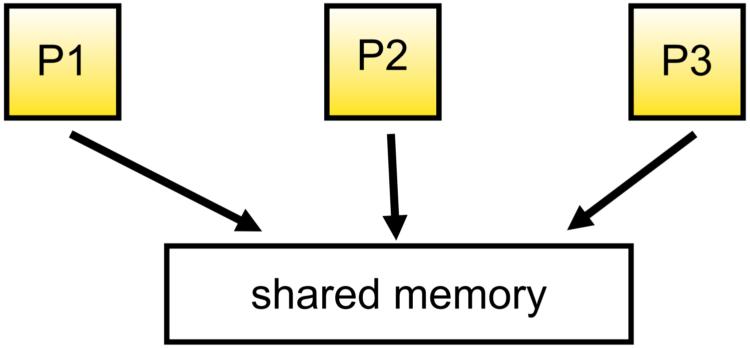
Shared Memory

System V IPC

- Shared memory
  - fastest IPC mechanism
  - map memory segments
    - virtual storage
  - shared by processes
    - not networkable
    - no synchronization



System V IPC

- Shared memory
  - create
  - attach
  - use
  - detach

System V IPC

- `shmget`
  - create a shared memory segment

```
int shmget ( key_t key, int size, int shmflg);
```

  - creates a new memory segment if:
    - key specified as `IPC_CREAT`
    - key does *not* exist and `IPC_CREAT` set as part if `shmflag`
    - key does *not* exist and `IPC_CREAT` and `IPC_EXCL` are set as part if `shmflag`

System V IPC

- arguments:
  - size
    - size of memory block in bytes
    - ignored when segment exists
- limits

Maximum segment size	4MB	SHMMAX
Minimum segment size	1byte	SHMMIN
Systemwide max # segments	4090	SHMMNI
Maximum # of segments per process	X	SHMSEG

System V IPC

- shmid_ds data structure
  - maintained by kernel
    - permissions
    - size of segment
    - time of last attach
    - time of last detach
    - time of last change by shmctl
    - pid of creator

System V IPC

- shmctl

```
int shmctl ( int shmid, int cmd, struct shmid_ds *buf);
```

- IPC_STAT
- IPC_SET
- IPC_RMID
- IPC_LOCK
  - lock in memory
- IPC_UNLOCK

System V IPC

- shmat

```
void *shmat ( int shmid, const void *shmaddr,
                    int shmflg);
```

- adds entry to process segment/page tables
  - inherited by child processes
- shmaddr
  - 0 : system assigns memory location
- shmflag
  - SHM_RDONLY
  - SHM_RND
    - align to page boundary

System V IPC

- shmdt

```
int shmdt ( const void *shmaddr );
```

- detach segment from virtual storage
- if last process
  - segment is removed