

```

/* Unnamed Pipes */
int fd[2];
if (pipe(fd) == -1) {
...

/* Zum Lesen und Schreiben öffnen, aus Sicht des Parent-Process */
if ((firstStream = fdopen(childToParent[0], "r")) == (FILE *) NULL) {
...
if ((firstStream = fdopen(parentToChild[1], "w")) == (FILE *) NULL) {
...

/* Stdin, Stdout Umleitung auf Pipes (2 Richtungen) */
if (dup2(parentToChild[0], STDIN_FILENO) == -1) {
...
if (dup2(childToParent[1], STDOUT_FILENO) == -1) {
...

/* child process starten */
switch (mPid = fork()) {
    case -1:
        handleError();
        break;
    case 0:
        ChildProcess();
        break;
    default:
        ParentProcess();
        break;
}

/* image laden */
if (execlp(command, argv[0], argv[1], ... ,(const char *) NULL) < 0) {
...

/* Signal handling */
(void) signal(SIGQUIT, exitHandler);
(void) signal(SIGTERM, exitHandler);
(void) signal(SIGINT, exitHandler);

void exitHandler(int nSignal) {
...

```

```

/* Semaphore */
int sem_id;
// Server
if ((sem_id = seminit(SEM_KEY, 0600, 1)) < 0) {
...

// Client
if ((sem_id = semgrab(SEM_KEY)) < 0) {
...

if (semrm(sem_id) < 0) {
...

/* Shared memory anlegen */
int shmid;

// Server
if ((shmid = shmget(SHMEM_KEY, sizeof(Board), IPC_CREAT | PERMISSION)) < 0) {
...

// Client
if ((shmid = shmget(SHMEM_KEY, 0, PERMISSION)) < 0) {
...

Board *sharedBoard;
if ((sharedBoard = (Board *) shmat(shmid, NULL, 0)) == (Board *) -1) {
...

if (shmdt((void *) sharedBoard) < 0) {
...

if (shmctl(shmid, IPC_RMID, NULL) < 0) {
...

```