process (base class)

- pid_t **pid** (process ID)

listening process
(parent; inherits from process)

- pid_t **distributorPID**
  (ID of distributor process)
- Socket **listeningSocket**

distributor process
(coordinates client-oriented processes;
inherits from process;
spawned by the listening process)

- string **usersFileName**
  (file for storing user information)
- managed_shared_memory* **pUserSegment**
  (pointer to the object managing shared memory for exchanging user data)
- managed_shared_memory* **pMessageSegment**
  (pointer to the object managing shared memory for exchanging messages between processes)
- boost::interprocess::allocator* **pUserAllocator**
- boost::interprocess::allocator* **pMessageAllocator**
- boost::interprocess::vector* **pUserVector**
- boost::interprocess::allocator* **pMessageVector**

client-dedicated process
(communicates with other client-oriented processes through the distributor process;
inherits from process;
spawned by the listening process)

- string **usersFileName**
  (file for storing user information)
- Socket& **rCommunicationSocket**
- managed_shared_memory* **pUserSegment**
- managed_shared_memory* **pMessageSegment**
- boost::interprocess::vector* **pUserVector**
- boost::interprocess::allocator* **pMessageVector**

socket

- int **sfd** (file descriptor)
- int **bufSize**
- char* **buffer**
- sockaddr **networkAddress**

user

- int **userPID**
- string **userName**
- string **name**
- string **encryptedPassword**
- enum **status**
- vector<int> **contactIDs**

message

- pid_t **from**
- pid_t **to**

process ID of message sender and recipient

- enum **mType**
- int **length**
- char* **contents**
- offset_ptr<char> **offsetPtr** (?)

client (inherits from NCursesApplication)

Socket **socket**

login dialog (inherits from dialog; started by client; not spawned as a new process)

Socket* **pSocket**

(I can’t remember why I decided to use a reference to Socket on the server side, and a pointer here. It’s possible that I was just experimenting to see what would work. Is there a design preference?)

login item (inherits from NCursesMenuItem; started by login dialog; placed as one of the menu items in the login dialog)

LoginDialog* **pParentPanel**

(Needs connection with the parent window in order to access data from its form fields.)

signup item (inherits from NCursesMenuItem; started by login dialog; placed as one of the menu items in the login dialog)

Socket* **pSocket**

quit item (inherits from NCursesMenuItem; started by login dialog; placed as one of the menu items in the login dialog)

Socket* **pSocket**

dialog (inherits from NcursesForm and NcursesMenu)
(Is multiple inheritance considered a bad design decision?)

int **dalogHeight**

int **dalogWidth**

int **dalogY**

int **dalogX**

bool **inForm**

bool **inMenu**

NCursesFormField** **paFormFields**

NCursesMenuItem** **paMenuItems**

(pointers to arrays of structures that describe form fields and menu items in the form)

Alphanumeric_Field* **pANFType**

Regular_Expression_Field* **pREFType**

(used for describing field types)

signup dialog (inherits from dialog; started by login dialog (by the signup item in it); not spawned as a new process)

(No new members. Constructor populates different fields and menus of the parent Dialog class.)

signup OK item (inherits from NCursesMenuItem; started by signup dialog; placed as one of the menu items in the signup dialog)

SignupDialog* **pParentPanel**

Socket* **pSocket**

signup cancel item (inherits from NCursesMenuItem; started by signup dialog; placed as one of the menu items in the signup dialog)

(No members. Just used to cancel the sign-up process.)