Teoricamente, una volta fatto l’update dall’svn dovresti solo compilare il server e il client e tutto dovrebbe andare.

Noterai che ora nella cartella del server c’è un file configurazioneServer.cfg in cui c’è appunto la configurazione del server. Anche questi file stanno già sistemati sull’svn ma, dentro ci andrebbe:

|  |
| --- |
| ID:1 |
| Porta:5000 |
| cartella file condivisi:fileCondivisi/ |
| IPServerDNS:127.0.0.1 |
| PortaDNS:7000  L’ID del server deve essere UNIVOCO per ogni server. Il server non controlla se esiste un altro server con lo stesso ID quindi, nella guida, mettici che se l’amministratore di sistema si sbaglia siamo nella cacca. Magari non scrivere proprio nella cacca :D  La porta è quella di ascolto del server, le altre specifiche le capisci da sola. ☺  Nella cartella server ora ci sono due sottocartelle, server2 e server3. Questo per evitare che quando avii più server, essi scrivano nella stessa cartella fileCondivisi e leggano lo stesso file di configurazione (quindi stesso ID, stessa porta, casini su casini). Questa regola vale solo per il testing in locale, scrivi che, qualora si dovesse testare l’accrocchio in un ambito di rete reale, basta solo un server e non le due sottocartelle.  Ora non lo è, ma nel momenti in cui consegneremo il progetto, esso sarà dentro un file tar.gz. Quindi di che sto coso andrà scompattato prima di essere compilato.  Metti anche che va fatto il make del server e del client. Il make del server attualmente è ottimizzato solo per il testing in locale. Ovvero, prevede l’esistenza delle cartelle server2, server3 e ci inserisce all’interno la copia del file ServerReplica (eseguibile) già compilato.  Nella guida dovresti anche inserire i vari tipi di operazione che il client può sottomettere al server che sono:   * Lista File: Chiede al server la lista file e la stampa a video * Leggi File: Prende un file dal server e lo salva il locale * Scrivi File: Permette al client di scrivere una serie di cose in un file del server. Durante l’operazione di scrittura file è prevista l’operazione di “abort” per annullare tutto o di “commit” per scrivere le modifiche nel file system * Uscita: Permette al client di chiudere la connessione con il server   Credo che ti abbiamo detto tutto.  Ah, mettici pure che il server crea un file di log, server Replica.log dove c’è lo stato del server. |

|  |
| --- |
|  |