

Preparazione di 'hicgi' & C. in ambienti di tipo unix partendo dallo 'zip' dei sorgenti presenti in RightWay

Prima della lista di operazioni per per compilare i sorgenti e ottenere 'hicgi' e C., una lista di osservazioni in ordine sparso:

1. i sorgenti sono strutturati per essere contenuti in un direttorio "script" che però tipicamente non è compreso negli 'zip' dei sorgenti presenti in RightWay; si dovrà quindi creare il direttorio "script" e ivi decomprimere i sorgenti;
2. i sorgenti sono quasi certamente stati preparati in una piattaforma di tipo Win32 con fine riga di tipo "\x0d\x0a"; i file dello 'zip' dei sorgenti che interessano, nelle piattaforme di tipo unix, sono solo di tipo testo; anche se potrebbe non essere necessario, si suggerisce quindi di estrarre i sorgenti indicando che tutti i file sono di tipo testo (opzione "-aa" di "unzip");
3. oltre ad un banalissimo "makefile" principale, direttamente dentro "script", i sorgenti di 'hicgi' prevedono una coppia di "makefile" effettivi dentro a "script/umak" ("script/umak/makefile", per gli eseguibili, e "script/umak/makeso", per le (in realtà, una sola) librerie dinamiche) e una serie di file di supporto ai due makefile principali
4. la selezione della piattaforma di compilazione, nel caso del makefile relativo agli eseguibili, avviene in "script/umak/makeutil"; nella parte iniziale di tale file si trova una serie di righe, ognuna delle quali preposta ad assegnare un alla variabile "WHAT" - che indica la specifica piattaforma di compilazione - un opportuno valore; ovviamente una - e solo una - delle righe di assegnazione per "WHAT" non deve essere commentata
5. si noti che il meccanismo di cui al punto precedente è del tutto analogo a quello usato per 'xw' & C. tranne che non è presente un meccanismo di riconoscimento automatico per la piattaforma;
6. ancora per quanto riguarda il makefile relativo agli eseguibili, la selezione della modalità di compilazione - "debug" o "release" - è determinata dalla variabile "MODE", come definita all'inizio di "script/umak/makefile"; quindi si commenti/de-commenti opportunamente le due righe di definizione;
7. nel caso, invece, del makefile relativo alle librerie dinamiche, sia la piattaforma che la modalità si impostano commentando o de-commentando interi blocchi di variabili nella parte iniziale di "script/umak/makeso"; questa modalità corrisponde a quella della compilazione degli eseguibili prima della semplificazione che ha portato a introdurre le variabili "WHAT" e "MODE" [punti (4) e (6)]
8. si noti però che, per la compilazione delle librerie dinamiche, la configurazione all'inizio di "script/umak/makeso" è inadeguata, almeno per quanto riguarda la piattaforma linux; si tratta probabilmente di un copia/incolla della configurazione per la produzione di eseguibili che dovrebbe essere adattata - piattaforma per piattaforma - alla produzione di librerie dinamiche;
9. in considerazione del punto precedente - a meno che non si desideri espressamente la compilazione dell'unica libreria dinamica - si suggerisce di disattivare tale compilazione intervenendo su "script/makefile"

Riassumendo, per compilare - in ambienti di tipo unix - 'hicgi' & C. partendo da un file 'zip' di sorgenti di 'hicgi' registrati in RightWay, si proceda in questo modo:

- si crei un direttorio "script" (in minuscolo);
- si copi il file 'zip' di sorgenti dentro il direttorio "script" [per maggiori dettagli, vedasi anche punto (1) della lista precedente];
- si estraiga i sorgenti dal file 'zip' usando "unzip -aa <file.zip>" [punto (2)];
- si verifichi che i file e i direttori estratti ("3d", "base64", "bin", "c++", ecc.) siano direttamente contenuti nel direttorio "script";
- si selezioni la piattaforma di compilazione per gli eseguibili editando "script/umak/makeutil" e attivando l'opportuno valore di "WHAT" [punto (4)];
- si selezioni la modalità di compilazione "MODE" - "debug" o "release" - editando "script/umak/makefile" [punto (6)];
- si editi "script/makefile" commendando le due righe relative a 'makeso' [punto 9]
- dal direttorio "script" si dia il comando "make";
- in assenza di errori, si troveranno gli eseguibili prodotti in "script/bin/debug" o "script/bin/release", a seconda della modalità adottata.

Preparazione, in ambienti di tipo unix, di moduli secondari rispetto ad 'hicgi' ('ziw', 'unziw', 'hras', 'h3' e forse altri) partendo dallo 'zip' dei sorgenti presenti in RightWay

Contrariamente ad 'hicgi', per il quale i sorgenti in RightWay sono tipicamente completi (e quindi autosufficienti) e persino sovrabbondanti (poiché permettono la preparazione anche di moduli collaterali ad 'hicgi'), i sorgenti di questi programmi collaterali - come 'ziw', 'unziw', 'h3', 'hras' e forse altri - sono tipicamente incompleti e spesso comprendono solamente gli aggiornamenti specifici per il modulo in oggetto rispetto alla precedente versione di 'hicgi' presente in RightWay.

Quindi, volendo produrre una versione per una piattaforma di tipo unix di un modulo collaterale ad 'hicgi', ci si dovrà pesantemente affidare a RightWay procedendo in questo modo:

- dalla pagina di RightWay (RW in seguito) di una particolare release del modulo di nostro interesse, si estraiga lo 'zip' dei sorgenti; per semplicità, diciamo che questo si chiama "mod.zip";
- dalla stessa pagina di RW, si cerchi di individuare eventuali indicazioni specifiche (annotazioni o altro) relative alla



- preparazione dell'eseguibile; tenere eventualmente conto di queste annotazioni;
- in assenza di annotazioni di cui al punto precedente (o in presenza di annotazioni non contrastanti) si cerchi di capire se il file "mod.zip" contiene tutti i sorgenti necessari alla compilazione o solo quelli specifici del modulo; se i sorgenti sono completi ci si riconduca alle istruzioni per la preparazione di 'hicgi', altrimenti si continui con le presenti indicazioni;
 - si torni a guardare la pagina di RW del modulo per verificare se link-a una particolare versione di 'hicgi'; nel caso positivo, si estragga lo 'zip' dei sorgenti (diciamo "hicgi.zip") di tale versione di 'hicgi' e si passi al punto (vii); nel caso negativo si continui dal punto (v);
 - si individui, confrontando in RW la data della release del modulo con le date delle release di 'hicgi', l'ultima versione di 'hicgi' emessa prima della particolare release del modulo;
 - si estragga da RW lo 'zip' dei sorgenti (diciamo "hicgi.zip") della versione di 'hicgi' individuata al punto precedente;
 - si crei un direttorio "script" (in minuscolo);
 - si estraggano, ad esempio con "unzip -aa hicgi.zip", dentro "script" i sorgenti contenuti in "hicgi.zip";
 - si verifichi che i file e i direttori estratti ("3d", "base64", "bin", "c++", ecc.) siano direttamente contenuti nel direttorio "script";
 - si estragga il contenuto di "mod.zip" ("unzip -aa mod.zip") in un altro punto del file system in maniera tale da non interferire (non sovrascrivere) i file estratti da "hicgi.zip"
 - per ogni file estratto da "mod.zip", si individui il corrispondente file estratto da "hicgi.zip" e lo si sovrascriva (quello da "mod.zip" al posto di quello da "hicgi.zip"); si faccia particolare attenzione in questa fase e si tenga presente che la disposizione in direttori dei file presenti in "mod.zip" può non corrispondere a quella dei file presenti in "hicgi.zip";
 - si selezioni la piattaforma di compilazione per gli eseguibili editando "script/umak/makeutil" e attivando l'opportuno valore di "WHAT";
 - si selezioni la modalità di compilazione "MODE" - "debug" o "release" - editando opportunamente "script/umak/makefile";
 - si editi "script/makefile" commendando le due righe relative a 'makeso';
 - dal direttorio "script" si dia il comando "make";
 - in assenza di errori, si troveranno gli eseguibili prodotti in "script/bin/debug" o "script/bin/release" - a seconda della modalità adottata - tra i quali quello corrispondente al nostro modulo