WESTERN UNIVERSITY
Computer Science 3357a – Fall 2015
Assignment 1 Marking Scheme

Student Username: tetchel
Marked By: Gurjit Randhawa
Submitted On: Oct 13

Client (35%)

Criterion	Mark	Out Of	Comments
Compilation and Execution			
If code does not compile or run properly, read the code and mark as much as you can.	**0**	Compiles/runs successfully: **0** Wholly incorrect output/crashes often: **-13** Compiles; crashes immediately: **-20** Does not compile: **-24**	
Makefile			
`Makefile` provided	**1**	**1**	
Builds code simply by typing `make`	**1**	**1**	
Builds executable named `client`	**1**	**1**	
`Makefile` uses the `--Wall` and `--Werror` flags when compiling code (if they have defined `CFLAGS`, don't just look at its definition: make sure they're ***using*** it in their compilation commands)	**2**	**2**	
Command-Line Arguments			
Accepts directory	**1**	**1**	
Accepts output filename	**1**	**1**	
Prints error upon missing arguments	**1**	**1**	
Prints error if invalid directory given	**1**	**1**	
Prints error if invalid filename given (e.g. a directory name)	**1**	**1**	
Accepts directory given without trailing slash	**2**	**2**	
Accepts directory given with trailing slash	**2**	**2**	

Criterion	**Mark**	**Out Of**	**Comments**
Output			
Writes to output file specified on command line	**1**	**1**	
Overwrites existing output file (does not append to file)	**1**	**1**	
Output file in correct format (one `filename,checksum` pair per line)	**1**	**2**	Space found along with comma
Output file contains correct number of unique, non-hidden files	**4**	**4**	
Output file includes hidden files	**1**	**1**	
Output file contains correct filenames	**3**	**3**	
Output file does not contain duplicates	**2**	**2**	
Output file contains paths specified relative to the Hooli root	**3**	**3**	
Output file contains correct checksums	**2**	**3**	7 valid, 5 invalid
Output file represents checksums in decimal	**1**	**1**	
Program prints nothing to the screen on successful execution	**1**	**1**	
Program exits with return code of `0` on successful execution (`EXIT_SUCCESS`)	**1**	**1**	
Program exists with return code of `1` on failed execution (`EXIT_FAILURE`)	**1**	**1**	
Program does not crash on single file error (e.g. permissions problem)	**1**	**2**	Segmentation Fault
Efficiency			
Does not read entire file into memory all at once to compute checksum (look at their source)	**2**	**2**	
Does not produce memory errors (e.g. `double free or corruption`; `invalid next size`; etc.)	**2**	**2**	
Keeps output file open until directory scanning complete (does not open-write-close for each file – look at their source)	**2**	**2**	
Frees dynamically-allocated memory (run `valgrind` and check for memory leaks)	**2**	**2**	
Program is capable of handling a large directory tree	**2**	**2**	
Client Total	**47**	**50**	

libhdb (45%)

Criterion	Mark	Out Of	Comments
Compilation and Execution			
If code does not compile or run properly, read the code and mark as much as you can.	**0**	Compiles/runs successfully: **0** Had to be modified to compile/run successfully: **-10 to -20** Wholly incorrect output/crashes often: **-13** Compiles; crashes immediately: **-20** Does not compile: **-24**	
Library			
Passes unit tests	**30**	**30 (1 mark per test)**	
Makes efficient use of Redis commands (e.g. doesn't use 3 commands for an operation that only requires 1	**3**	**3**	
Does not print anything to the screen	**2**	**2**	
Did not modify `hdb.h`	**1**	**2**	Modification found
Efficiency			
Frees dynamically-allocated memory (run `valgrind` and check for memory leaks)	**3**	**3**	
libhdb Total	**39**	**40**	

Style and Comments (20%)

All criteria in this section to be marked using the following scale:
Never to almost never: 0; Rarely: 1; Sometimes: 2; Usually to Always: 3

Criterion	Mark	Out Of	Comments
Comments			
Code contains inline comments allowing reader to follow algorithm	3	3	
Variable declarations commented	3	3	
Header comment in each file with the appropriate information	3	3	
Comments are descriptive and do not simply repeat what the code is saying	3	3	
Attention paid to spelling, grammar, punctuation, capitalization	3	3	
Files devoid of code that is commented out	3	3	
Style			
Uses descriptive variable names	3	3	
Uses descriptive function names	3	3	
Uses constants instead of magic numbers	3	3	
Functions kept to a reasonable length	3	3	
Code is modular and divided into multiple files, where appropriate	3	3	
Lines kept to a reasonable length (80 - 100 chars)	3	3	
Code is clean and well-formatted	3	3	
Appropriate use of whitespace	3	3	
Does not use global variables (give 0 here for ANY use of a global variable)	0	3	rootln in client.c
Style and Comments Total	42	45	

Submission Errors (up to 5% off)

Criterion	Deduction	Out Of	Comments
Submission not tagged `asn1`	**-3**	**-3**	No tag found
Directories `client` and `hdb` not found in root of repository	**-1**	**-2**	No hdb directory found
Submission contains disallowed files	**0**	**-1 per file (max. -3)**	
Submission missing required files	**0**	**-1 per file (max. -3)**	
Total Deductions	**-4**	**Max. -5**	

Evaluation Summary

Section	Mark	Out Of	Weight	Weighted Mark (round up to nearest integer)
Client	**47**	**50**	**35%**	**33**
libhdb	**39**	**40**	**45%**	**44**
Style and Comments	**42**	**45**	**20%**	**19**
			Subtotal	**96**
			- Submission Errors (max. -5)	**-4**
			- Late Penalty no penalty if submitted by 23:59:59 on October 13 10% penalty by 4 PM on October 14 do no mark after 4 PM on October 14	**0**
			Total (out of 100)	**92**

Comments

Overall nice effort.