

WESTERN UNIVERSITY
Computer Science 3357a – Fall 2015
Assignment 2 Marking Scheme

Student Username: jquirin2
Marked By: Chesakov, Egor
Submitted On: November 17, 2015

Part 0 (Bonus up to 10%)

Criterion	Mark	Out Of	Comments
Compilation and Execution			
If the code does not compile, skip this section and do not award bonus marks.			
Library			
Passes unit tests	37	41 (1 mark per test)	90%: Checks: 41, Failures: 4, Errors: 0
Efficiency			
Frees dynamically-allocated memory (run <code>valgrind</code> and check for memory leaks)	0	4	<pre> ==4178== LEAK SUMMARY: ==4178== definitely lost: 74 bytes in 5 blocks ==4178== indirectly lost: 0 bytes in 0 blocks ==4178== possibly lost: 0 bytes in 0 blocks ==4178== still reachable: 0 bytes in 0 blocks ==4178== suppressed: 0 bytes in 0 blocks </pre>
libhdb Total	37	45	

Client (35%)

Criterion	Mark	Out Of	Comments
<i>Compilation and Execution</i>			
If code does not compile or run properly, read the code and mark as much as you can. Subtract 5 marks only if their program doesn't work with our test server, but otherwise works fine with their server.	0	Compiles/runs successfully: 0 Wholly incorrect output/crashes often: -13 Compiles; crashes immediately: -20 Does not compile: -24	
<i>Makefile</i>			
Makefile provided	1	1	
Builds code simply by typing make	1	1	
Builds executable named <code>client</code>	1	1	
Makefile uses the <code>--Wall</code> and <code>--Werror</code> flags when compiling code (if they have defined <code>CFLAGS</code> , don't just look at its definition: make sure they're <i>using</i> it in their compilation commands)	2	2	
<i>Command-Line Arguments</i>			
Uses <code>getopt_long</code> for parsing	2	2	
Accepts both short and long options (try <code>-p</code> and <code>--port</code> , for example)	2	2	
Accepts server	1	1	
Accepts port	1	1	
Accepts directory	1	1	
Accepts verbose flag	1	1	
Prints error upon missing arguments	1	1	
Prints error if invalid directory given	1	1	
Accepts directory given without trailing slash	1	1	
Accepts directory given with trailing slash	1	1	

Criterion	Mark	Out Of	Comments
<i>Output</i>			
Uses syslog to display program output	2	2	
--verbose results in more verbose output	1	1	
Sends AUTH command containing username and password in headers	2	2	
AUTH request is well-formed (e.g. AUTH\Headers\n\n)	3	3	
AUTH request contains no body	1	1	
Prints error and exits on authentication failure	1	1	
Sends LIST command containing server-issued token in headers	2	2	
LIST request is well-formed (e.g. LIST\Headers\n\nBody)	3	3	
LIST request contains body	2	2	
LIST request has files on odd-numbered lines (1, 3, 5, ...)	2	2	
LIST request has checksums on even-numbered lines (2, 4, 6, ...)	2	2	
LIST request contains checksums in uppercase hexadecimal (e.g AABBBCCDD)	1	1	
Prints list of files requested by server	1	1	
If no files requested, prints appropriate message	1	1	
<i>Efficiency</i>			
Frees dynamically-allocated memory (run valgrind and check for memory leaks)	4	4	
<i>Client Total</i>	45	45	

Server (45%)

Criterion	Mark	Out Of	Comments
<i>Compilation and Execution</i>			
If code does not compile or run properly, read the code and mark as much as you can. Subtract 5 marks only if their program doesn't work with our test server, but otherwise works fine with their server.	0	Compiles/runs successfully: 0 Wholly incorrect output/crashes often: -13 Compiles; crashes immediately: -20 Does not compile: -24	
<i>Makefile</i>			
Makefile provided	1	1	
Builds code simply by typing make	1	1	
Builds executable named hmds	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 <i>using</i> it in their compilation commands)	2	2	
<i>Command-Line Arguments</i>			
Uses getopt_long for parsing	2	2	
Accepts both short and long options (try -p and --port, for example)	2	2	
Accepts port (-p / --port)	1	1	
Accepts Redis server hostname (-r / --redis)	1	1	
Accepts verbose flag (-v / --verbose)	1	1	
Prints error if invalid port given	1	1	

Criterion	Mark	Out Of	Comments
<i>Output</i>			
Uses syslog to display program output	2	2	
--verbose results in more verbose output	1	1	
First AUTH request returns 200 Authentication successful	3	3	
AUTH response is well-formed (e.g. 200 Authentication successful\nHeaders\n\n)	3	3	
AUTH Response contains no body	2	2	
Subsequent successful AUTH request returns new token	3	3	
Subsequent failed AUTH request returns 401 Unauthorized	2	2	
LIST request with new files returns 302 Files requested	3	3	
LIST response is well-formed (e.g. 302 Files requested\nHeaders\n\nBody)	3	3	
LIST response contains correct list of files requested in body (One per line; no checksums should be in response)	3	3	
Server prints list of files sent by client in verbose mode	1	1	
Server prints list of files requested in verbose mode	1	1	
LIST request with invalid token returns 401 Unauthorized	2	2	
<i>Efficiency</i>			
Frees dynamically-allocated memory (run valgrind and check for memory leaks)	3	3	
<i>Client Total</i>	45	45	

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)	3	3	
Style and Comments Total	45	45	

Submission Errors (up to 5% off)

Criterion	Deduction	Out Of	Comments
Submission not tagged as n2	0	-3	
Directories client and hmds not found in <u>root</u> of repository	0	-2	
Submission contains disallowed files	0	-1 per file (max. -3)	
Submission missing required files	0	-1 per file (max. -3)	
Total Deductions	0	Max. -5	

Evaluation Summary

Section	Mark	Out Of	Weight	Weighted Mark (round up to nearest integer)
Bonus	37	45	+10%	9
Client	45	45	35%	35
hmds	45	45	45%	45
Style and Comments	45	45	20%	20
Subtotal				109
- Submission Errors (max. -5)				0
- Late Penalty no penalty if submitted by 23:59:59 on November 13 4% penalty by 23:59:59 on November 15 8% penalty by 23:59:59 on November 16 16% penalty by 23:59:59 on November 17 do no mark after 23:59:59 on November 17				-16
Total (out of 100)				93

Comments