//版本: 发布版 v1.0

1. 概述

1.1 架构分析

Flood 采用一核心 Server, 若干 MySQL Server, 若干 Judge, 若干 PHP Server, 若干 Client 的架构模式, 由 Server 对其他四类模块进行通信, 其他四类模块之间相互之间无任何通信关系. 由于架构设计, 且本系统内部除 MySQL 外无任何与其他系统或商业系统的通用接口, 故 MySQL 采用标准端口和接口协议, 其他模块接口会采用自定义端口和接口协议.

本文所讨论的均为 Judge, PHP Server, Client 与 Server 的通信接口关系.

1.2 通用接口分析

// 此段落失效, 请看下面具体的分析以及源代码的配置文件

自定义的端口为 Judge 3600 - 3615, PHP Server 为 3616 - 3632, Client 统一使用 3633 端口. 通信的所有信息均采用二进制编码, 并且数据按网络高位模式传输. 通信协议头部统一为 10 个字节, 头两个字节为区别模块及请求类型, 其中第一和第二位用来区别模块, 00 为 Judge, 01 为 Client, 10/11 均为 PHP Server, 后八个字节为各模块自定义内容. 如某次通信需要传输其他数据, 则头部后的数据长度需要在头部中后四字节说明长度, 最长为 2G, 数据结束不设置特殊字符标识.

2. Judge 端接口设计

请求 Judge

s->j[1]: char[9], 1 字节表示源代码类型, 2-5(4) 字节表示题号, 6-9(4) 字节表示题目版本号 (version)

j->s[2]: char, READY(状态正常, 可以后续操作) 或 SYSTEM_ERROR/UNSUPPORTED_FILE_TYPE

发代码文件

s->j[3]: char[4], 代码文件长度

j->s[4]: char, READY(状态正常, 可以后续操作) 或 SYSTEM_ERROR/INVALID_FILE_SIZEa

s->j[5]: 步骤 3 里的文件长度那么长的 char*, 代码文件内容

j->s[6]: char, READY(需要数据文件, 跳至步骤 7) 或 DATA_EXSIST(数据文件已存在, 跳至步骤 10) 或 SYSTEM_ERROR,

发数据文件

s->j[7]: char[4], 打包后数据文件的长度

j->s[8]: char, READY(状态正常, 可以后续操作) 或 SYSTEM_ERROR/INVALID_FILE_SIZE

s->j[9]: 步骤 7 里的文件长度那么长的 char*, 代码文件内容

j->s[10]: char, READY(状态正常, 可以后续操作) 或 SYSTEM_ERROR

发题目限制

s->j[11]: char[13], 1 字节表示 test_case 个数, 2-5(4) 字节表示 time_limit(ms), 6-9(4) 字节表示 case_time_limit(ms), 10-13(4) 字节表示 memory_limit(KiB)

j->s[12]: char, READY(状态正常, 请服务器接收后续返回) 或 SYSTEM_ERROR

编译阶段

j->s[13]: char, COMPILING(开始编译)

j->s[14]: char, READY(编译结束, 状态正常, 请服务器接收后续返回, 跳至步骤 17) 或 SYSTEM_ERROR, 或 COMPILE_ERROR(跳至步骤 15)

j->s[15]: char[2], CE 信息长度(uint16_t)

j->s[16]: 步骤 15 里的信息长度那么长的 char*, CE 错误信息

运行阶段

j->s[17]: char, RUNNING(开始运行)
j->s[18]: char, READY(运行结束, 状态正常, 请服务器接收后续返回, 跳至步骤 21) 或 RUNTIME_ERROR_JAVA/RUNTIME_ERROR_PASCAL(跳至步骤 19) 或其他终止信息(跳至步骤 23)
j->s[19]: char[2], RE 信息长度(uint16_t)
j->s[20]: 步骤 19 里的信息长度那么长的 char*, RE 错误信息
判断阶段
j->s[21]: char, JUDGING(开始判断)
j->s[22]: char, ACCEPTED(如果还有 test_case, 跳至步骤 17, 否则跳至步骤 23), WRONG_ANSWER/PRESENTATION_ERROR 或 SYSTEM_ERROR(跳至步骤 23)
最终结果阶段
j->s[23]: char[9], 1 字节表示结果, 2-5(4) 字节表示运行时间(ms), 6-9(4) 字节表示运行耗内存

3. Client 端接口设计

4. PHP Server 端接口设计

协议:
header: header长度为**10**字节, 包括两个部分, 前两个字节标示协议类型, 后**8**个字节标示待接收数据长度

(1), **st (status** 提交状态**)**
p->s: header
p->s: pagenum, problem_id(0), user_id(?), Result(?), language(?), contest_id(?), share_code(Y/N[N:all]) ,type(N[N:common, P:problem_status, R:root_status, S:standard_limit, A:admin, F:file]), cur_user_id(?)
s->p: header
s->p: solution_id,user_id,problem_id,result,memory,time,language,code_length,in_date, code_id, errir_id, permission(Y/N)

type用于区别是status还是problem status

(2), ss (solution status 题目的提交状态, **ac**总数, **wa**总数等**)**
p->s: header
p->s: problem_id
s->p: header
s->p: Accepted, TotalSubmits, UserSubmited, UserSolved, Pe, Tle, e, Ole, Wa, Re, Ce

(4), sc(源代码**)**
p->s: header
p->s: code_id
s->p: header
s->p: source

(5), ml(邮件列表**)**
p->s: header
p->s: user_id, pagenum
s->p: header

s->p: mail_id, to_user, from_user,title, in_date, read

(6), mc(邮件内容**)**
p->s: header
p->s: mail_id
s->p: header
s->p: to_user, from_user, title, in_date, concont

(7), dm(删除邮件**)**
p->s: header
p->s: mail_id, user_id
s->p: 'Y'/'N'

(8), am(添加邮件**)**
p->s: header
p->s: topicid(-1), title, content, touser, fromuser
s->p: 'Y'/'N'

(9), hp(HomePage)
p->s: header
s->p: header
s->p: UpcomingContest(contest_id, title, start_time), MostDiligentProgrammer(num:个数, 每项是user_id) , news(title, time)

(9+), nl(news list)
p->s: header
s->p: header
s->p: news(title, time)

(10), rg(register 注册**)**
p->s: header
p->s: user_id, password, nick, share_code, school, email, share_email, language
s->p: 's'

(11), ei(user_ID是否存在**)**
p->s: header
p->s: user_id
s->p: 'Y'/'N'

(12), pl(题目列表**)**
p->s: header
p->s: pagenum, problem_id(0), title(?), source(?) , contest_id(0), user_id(?)[pagenum=?表示请求所有页面]
s->p: header
s->p: pagenum, (problem_id, title, source, ac, total, result(0/1ac/2wa))

(13), li(登录**)**
p->s: header
p->s: user_id, password, ip
s->p: 'Y'/'N'
s->p: new_mail_count(10 bytes)

(14), pb (problem)

p->s: header
p->s: problem_id
s->p: header （当problem不存在是返回0,跳出)
s->p: title, description, input, output, sample_input, sample_output, hint, source
time_limit, memory_limit, submit, accept, version

(15), cl(contest list 比赛列表**)**
p->s: header
p->s: pagenum, type(C,V) [c:普通比赛 v:虚拟比赛]
s->p: header
s->p: contest_id, title, start_time, end_time, type, sequence_no
[注]:N 为正常公开比赛, P为正常非公开比赛, V为虚拟比赛:
sequence_no 使得正常比赛编号连续

(16), cc(contest 比赛**)**
p->s: header
p->s: contest_id
s->p: header
s->p: title, description, start_time, end_time, sequence_no, version

(17), cp(contest problem 比赛题目列表**)**
p->s: header
p->s: contest_id, user_id
s->p: header
s->p: permission(Y/N), (problem_id, in_contest_id, problem_title, problem_ac, problem_total)

(18), rk(user ranklist 用户排名**)**
p->s: header
p->s: pagenum, of1 [of1:排序关键字] 0正常, 1是submit, 2是radio
s->p: header
s->p: user_id, nick, solved, submit

(19), ui(userinfo 用户信息**)**
p->s:header
p->s:user_id
s->p:header
s->p:rank, solved, submit, school, email, solvedproblem

(19+), ub(userbase 用户基本信息**)**
p->s:header
p->s:user_id
s->p:header
s->p:user_id, nickname, share_code, school, email, showemail

(20), cr(contest ranklist 比赛排名**)**
p->s:header
p->s:contest_id, page_id
s->p:header
s->p:problem_num, (user_id, nickname, accepted, penalty, (problem_penalty, submit))
[注]: submit 已经改成显示的状态,

(21), cs(contest statistics 比赛统计**)**
p->s:header
p->s:contest_id
s->p:header
s->p:problem_id, AC, PE, CE, WA, TLE, RE, E, OLE, Total, C/C++, Java, Pascal

(21), dl(discuss list 讨论列表**)**
p->s:header
p->s:title(?), problem_id(0), contest_id(0), user_id(?), page_id(0)
s->p:header
s->p:level, discuss_id, title, date, user_id, problem_id, contest_id
[1]: level=1,2,3...

(22), dc(discuss content 讨论内容)
p->s:header
p->s:discuss_id
s->p:header
s->p:topic_id, title, problem_id, contest_id, user_id, date, content

(23), ad(add discuss 提交讨论**)**
p->s:header
p->s:reply_id(0), topic_id(-1), user_id, problem_id(0), contest_id(0), title, content
s->p:Y/N

(24), dd(disable discuss 删除讨论**)**
p->s:header
p->s:discuss_id
s->p:Y/N

(25), uu(update user 更新用户**)**
p->s:header
p->s:user_id, old_password, new_password, email, show_email(Y/N), nickname, school, share_code(Y/N), language
s->p:Y/N

(26), du(disable user 删除用户**)**
p->s:header
p->s:user_id
s->:Y/N

(27), an(add news 添加新闻**)**
p->s:header
p->s:title, content
s->p:Y/N

(28), ap(add problem 添加题目**)**
p->s:header
p->s:title, descriptionm, input, output, sample_input, sample_output, hint, source, time_limit, case_time_limit, memory_limit, spj
s->p:problem_id(10位)

(29), io(add input and output file 添加**input**和**output)**
p->s:header (此时的length 为 input-output 对的个数)

p->s:header(problem_id, 10 bytes)
p->s:header (inputlength)
p->s:inputfile
p->s:header(outputlength)
p->s:outputfile
s->p:Y/N

(31), fp(add file to problem 提交与题目有关的文件)
p->s:header
p->s:filename
p->s:problem_id (10位),
p->s:header(filelength(10位))
p->s:filedata
s->p:'Y'/'N'

(32), up(update problem 更新题目信息**)**
p->s:header
p->s:problem_id, title, description, input, output, sample_input, sample_output, hint, source, time_limit, case_time_limit, memory_limit, spj, update_file_flag(Y/N) (需要更新 in 和 out 或者 spj 时用 Y)
s->p:Y/N

(33), mp(most problem 题目所有信息**)**
p->s:header
p->s:problem_id
s->p:header
s->p:title, description, input, output, sample_input, sample_output, hint, sources, addin_time, time_limit, case_time_limit, memory_limit, standard_time_limit, standard_memory_limit, version, spj

(34), pv(problem version 题目的版本**)**
p->s:header
p->s:problem_id
s->p:header
s->p:version

(35), bp(enable/disable problem 屏蔽**/**取消屏蔽题目**)**
p->s:header
p->s:problem_id, able(Y(cancel bp)/N(bp))
s->p:Y/N

(36), gp(get problem file 获取题目有关文件**)**
p->s:header
p->s:problem_id,
s->p:header(文件个数)
s->p:filename_length, filename, size, data

(37), rp(admin problem list 管理员控制台题目列表**)**
p->s:header
p->s:type, page_id,

s->p:header(大小)
s->p:problem_id, title, accepted, submit, available('Y'/'N')

(38), ac(add contest 添加比赛**)**
p->s:header
p->s:title, description, start_time, end_time, type(public:N, private:P Virtual: V)
s->p:contest_id(10bytes)

(39), rc(admin contest list 管理员控制台比赛列表**)**
p->s:header
p->s:page_id
s->p:header
s->p:contest_id, title, start_time, end_time, type, public_id, available

(40), bc(able contest 屏蔽**/**取屏蔽比赛**)**
p->s:header
p->s:contest_id, available(Y(cancel bp)/N(bp))
s->p:Y/N

(41), cv(contest version 获取比赛版本)
p**->**s:header
p->s:contest_id
s->p:header
s->p:version

(42), uc(update contest 更新比赛**)**
p->s:header
p->s:contest_id, title, description, start_time, end_time, type
s->p:Y/N

(43), fc(add file to contest 添加比赛有关文件**)**
p->s:header
p->s:filename, contest_id(10位), file_length, file_data
s->p:header(同fp)
s->p: path

(44), gc(get contest file 获取与比赛有关文件**)**
p->s:header
p->s:contest_id,
s->p:header(文件数目)
s->p:filename_length, filename, data_size, data

(45), pc(add problem to contest 向比赛添加题目**)**
p->s:header
p->s:contest_id, (problem_id)
s->p:Y/N

(46), cu(add user to contest 想比赛添加允许用户)
p->s:header
p->s:contest_id, (user_id)
s->p:Y/N

(47), sm(submit 提交代码)

p->s:header （长度不包括 source 的长度)
p->s:user_id, password, problem_id, contest_id(0), language, code_length, share_code(Y/N), ip, type(提交类型 N: 正常提交, R: root提交)
p->s:source;
s->p:Y/N

(48), rj(rejudge Rejudge)
p->s:header
p->s:status_id
s->p:Y/N

(49), ic(standard source test 测试标程**)**
p->s:header
p->s:problem_id, user_id, password
s->p:Y/N
s->p:status_id;

(50), ec(error content 错误信息**)**
p->s:header
p->s:error_id
s->p:header
s->p:error_content

(51), ul(user list 用户列表**)**
p->s:header
p->s:user_id(?) , nickname(?), page_id
s->p:header
s->p:user_id, nickname, last_login_ip, last_login_time, reg_time

(52), rt(check Permission 测试权限**)**
p->s:header
p->s:right_type(V: 查看代码, A: 管理员, C: 参加比赛权限), user_id, code_id(V)/无(A)/contest_id(C)
s->p:Y/N

(52), un(update news 更形新闻**)**
p->s:header
p->s:news-id, title, content
s->p:Y/N

(53), sn(set Notice 设置 **Notice)**
p->s:header
p->s:notice
s->p:Y/N

(54), np(notice 获取**notice)**
p->s:header
s->p:header
s->p:notice

(55), su(add series user 批量添加用户**)**
p->s:header
p->s:number , (user_id, passowrd)

s->p:header
s->p:(Y/N)

(55), si(set user info 修改用户信息**(**管理员修改**))**
p->s:header
p->s:user_id, email, nickname, school
s->p:Y/N

(56), as(add student 添加学生信息**)**
p->s:header
p->s:user_id, student_id, realname, college, grade, class_no
s->p:Y/N

(57), us(update student 更新学生信息**)**
p->s:header
p->s:user_id, student, realname, college, grade, class_no
s->p:Y/N

(58), ds(disable student 屏蔽(或取消屏蔽)学生**)**
p->s:header
p->s:user_id, available
s->p:Y/N

(59), sp(student 获取学生信息**)**
p->s:header
p->s:user_id
s->p:header
s->p:user_id, student_id, realname, college, grade, class, available

(60), at(add teacher 添加教师信息**)**
p->s:header
p->s:user_id, realname
s->p:Y/N

(61), ut(update teacher 修改教师信息)
p->s:header
p->s:user_id, realname
s->p:Y/N

(62), dt(disable teacher 屏蔽(或取消屏蔽)教师)
p->s:header
p->s:user_id, available
s->p:Y/N

(63), aa(add control class 添加课程**)**
p->s:header
p->s:teacher_id, description
s->p:course_id[failed:0]

(64), da(delete control class 删除课程**)**
p->s:header
p->s:teacher_id, class_id
s->p:Y/N

(65), aj(add job 添加作业)
p->s:header
p->s:description, course_id, year, term,
s->p:job_id

(66), pj(add problem to job 向作业加题)
p->s:header
p->s:type(M: 必做题, S: 选做题), job_id, should_do_number(必做题为0), (problem_id ...)
s->p:Y/N

(67), dj(disable job 屏蔽或取消屏蔽作业)
p->s:header
p->s:job_id, available(Y/N)
s->p:Y/N

(68), jc(add job to class 布置作业)
p->s:header
p->s:job_id, course_id
s->p:Y/N

(69), ue(update set 更新选做题组)
p->s:header
p->s:set_id, should_do_number, (problem_id ...)
s->p:Y/N

(70), sj(delete set for job 作业删除选做题组)
p->s:header
p->s:job_id, set_id
s->p:Y/N

(71), ij(is job done 查看学生作业信息)
p->s:header
p->s:user_id, job_id
s->p:header
s->p:job_id, must_do_number(必做题数), (problem_id, isdone(Y/N) ...),
done_must_number(必做题完成数目),
should_do_set_number(选做题目组数), (should_do_problem(需要完成多少题),
should_do_set_size(题目组题目数), (problem_id, isdone(Y/N) ...),
done_should_problem(选做题完成题目数) ...)

(72), jl(job list 作业列表)
p->s:header
p->s:course_id
s->p:header
s->p:(job_id, description, publish_time, course, year, term ...)

(73), jp(job 作业详情)
p->s:header
p->s:job_id
s->p:header
s->p:job_id, description, publish_time, course_id,year, term, must_do_number(必做题数),

(problem_id ...), should_do_set_number(选做题目组数),
(set_id, should_do_problem(需要完成多少题), should_do_set_size(题目组题目数),
(problem_id ...) ...)

(74), lj(list job done 作业完成情况报表)
p->s:header
p->s:job_id
s->p:header
s->p:(userid, realname, student_id, available(Y/N), grade, score(0) ...)

(75), uj(update job 更新作业**)**
p->s:header
p->s:job_id, description, course_id, year, term
s->p:Y/N

(76), es(add series student 批量添加学生**)**
p->s:header
p->s:number, (user_id, password, student_id, realname, grade, class_id ...)
s->p:header
s->p:(Y/N)

(77), ts(teacher or student 判断是不是学生或者教师**)**
p->s:header
p->s:user_id,T/S
s->p:Y/N

(78), sl(student list 学生信息列表**)**
p->s:header
p->s:course_id
s->p:header
s->p:(userid, realname, student_id, available(Y/N), grade, class ...)

(79), tl(teacher list 教师列表**)**
p->s:header
s->p:header
s->p:(user_id, realname, available ...)

(80), ol(course list 课程列表**)**
p->s:header
p->s:teacher
s->p:header
s->p:(course_id, description ...)

(81), op(course process 课程信息**)**
p->s:header
p->s:course_id
s->p:header
s->p:course_id, description, teacher

(82), js(job list for student 学生的作业列表**)**
p->s:header
p->s:student
s->p:header

s->p:(job_id, description, publish_time, course, year, term ...)

(83), so(add student to course 添加学生到课程**)**
p->s:header
p->s:student, course_id
s->p:Y/N

(84), eo(add series student to course 添加学生到课程**)**
p->s:header
p->s:course_id, (student[user_id] ...)
s->p:header
s->p:(Y/N ...)

(85), do(delete student for course 删除学生到课程**)**
p->s:header
p->s:student, course_id
s->p:Y/N

(86), tp(teacher information 教师信息**)**
p->s:header
p->s:user_id
s->p:header
s->p:(user_id, realname, available)

(87), sd(submit doc 提交文件**)**
p->s:header
p->s:user_id, password, problem_id, language, ip, type[F]
s->p:status_id

(87), sr(set result 修改结果**)**
p->s:header
p->s:status_id, result
s->p:Y/N

(88), sa(status list for teacher 作业的**status)**
p->s: header
p->s: job_id, page_id
s->p: header
s->p: solution_id,user_id,problem_id,result,memory,time,language,code_length,in_date, code_id, errir_id

[注:user_id:学号 student_id:身份证号]

Client 网络接口

(1), cp (client 初始化**)**
c->s: header
s->c:size(数据包长度) 以\n结尾
s->c:problem_num, (problem_id, ..), contest_num, (contest_id, contest_problem_num, (contest_problem_id, ..), ..) 以\n结尾

(2), st(client status List)
c->s:header

c->s:user_id
s->c:size(数据包长度) 以\n结束
s->c:status_num, (status_id, problem-id, contest_id, result, memory, time, language, code_length, submit_time, ..) 以\n结束

(3), li(登录)
p->s: header
p->s: user_id, password
s->p: 'Y'/'N'

(4), sm(submit 提交代码)
p->s:header (长度不包括 source 的长度)
p->s:user_id, password, problem_id, contest_id, language, code_length, share_code, source
s->p:Y/N