



## ვებ კომპილატორი ფუნქცია

მოცემული პროექტი არის ვებ აპლიკაცია, რომლის მეშვეობითაშ შესაძლებელია საიტზე კოდის ატვირთვა და მოცემული ამოცანისთვის ტესტირება და რეზულტატის დაბრუნება. მომხმარებელს შეუძლია აირჩიოს კომპილატორი (ჩვენ შემთხვევაში გვაქვს cpp-ის GNU კომპილატორი) [თუმცა მუშაობის პრინციპიდან გამომდინარე საკმაოდ მარტივია სხვა კომპილატორების ინტეგრაციაც] ატვირთოს კოდი, შემდეგ პროგრამა დააკომპილირებს, გაუშვებს ტესტებზე და დაუბრუნებს რეზულტატს [კომპილაციის შეცდომა | runtime ერორი | მესხიერების ლიმიტი | დროის ლიმიტი | არასწორი პასუხი | Accept ]

## ნაწილები

ჩვენი აპლიკაცია შედგება ორი მოდულისგან, პირობითად ერთი Front-end, მოდული რომელიც პასუხისმგებელია მომხმარებლისთვის მორგებული ინტერფეისის გამზადება, კოდის ასატვირთად და პასუხების გამოსაჩენად და server-side ნაწილი რომელიც აკომპილირებს და უშვებს კოდს და აბრუნებს პასუხს. გამოცანის გამოწვევებიდან გამომდინარე ჩვენ აღწერაში ვკონცენტრირდებით server-side ნაწილზე.

## იმპლემენტაცია

პროგრამის ძირითადი ოპერაცია არის სხვა გარე პროგრამის დასტარტვა როგორც პროცესად, პირველ რიგში ხდება gcc-ის დასტარტვა რომელსაც გადაეცემა დასაკომპილირებელი ფაილის სახელი, თუ პროცესის დასრულებისას სტატუსი დაბრუნდება 0 ნიშნავს რომ კომპილაციამ სწორად იმუშავა და პროცესი გრძელდება, ხოლო თუ დაბრუნდა 0-ისაგან განსხვავებული რიცხვი, მოცემულ შემთხვევაში ხდება გაჩერება და კომპილაციის შეცდომის დაბრუნება. შემდეგ ეტაპზე ხდება არსებული დაბრუნებული პროგრამის გაშვება (სიმარტივისთვის gcc ის გადავყავით -o პარამეტრი და დავრქვით ჩვენთვის საურველი სახელი, შედეგად უკვე ვიცით რომ არსებული მიღებული

პროგრამის სახელი და შეგვიძლია გაშვება.

პროგრამის გაშვების პარალელურად ხდება background thread-ის დასტარება, რომელიც ასევე უშვებს პროგრამას რომელიც პროგრამის მუშაობის დასრულებამდე იღებს ინფორმაციას მისი გამოყენებული ემსხიერებისა და დროის(cpu-ს მოხმარების) შესახებ. როგორც რესურსი გადააჭარბებს ლიმიტს მოცემული thread მოკლავს გაშვებულ პროცესს. შესაბამისად გვაქვს კლასი რომელშიც ეთითება მიმდინარე შეცდომა და ხდება შეცდომის მესიჯთან map-ირება და დაბრუნება. გაშვებული პროგრამის დასაწყისში ვიღებთ input stream-ს და ვწერთ შიგნით არსებულ input-ს და პროცესის დაბრუნებისას ვიღებთ output stream-ს და ვკითხულობთ და ვაღარებთ რეზულტატს. როგორც მოთხოვნაში იყო კითხვა ხდება stdin-ით ანუ კონსოლიდან და ანალოგიურად წაკითხვაც. არსებულ მოდულს გადააეცემა თითოეული ტესტების დირექტორია(მისამართი), ტესტის ფაილი (მისამართი) და ემსხიერების და დროის შეზღუდვები, რომელიც არსებულ პროცესს (რომელიც აღწერეთ) სათითაო ტესტისთვის უშვებს, რაც მერამდენე ტესტზე მოხდა შეცდომა, ამის დადგენის საშუალებას იძლევა.

## სპეციფიკაცია

პროექტის server side ნაწილი დაწერილია java-ზე და დაფუძნებულია რამდენიმე გარე პროგრამაზე, მაგ როგორიცაა g++ (უნდა იყოს დაყენებული არსებული კომპილატორი) და ps (პროცესის ინფორმაციის გასაგებად, არსებული პროგრამა ჩაშენებული აქვს linux-ს, სხვა ოპერაციულ სისტემაზე თუ პრობლემა შეიქმნა გაუშვით ლინუქსზე ან ლინუქსის კონტეინერში) Front-ისთვის:

პროგრამა უნდა დაიბილდოს და გაეშვას java8-ზე..

webserver-ად გამოიყენეთ tomcat 9.

## პროექტზე მუშაობდნენ:

ბერიძე ნიკა

ლაბარტყავა გური

სახელაშვილი მამუკა

ქასრაშვილი ანდრო