**Linux commandline – Bash scripting**

*Daan Wijns*

*Datum : 18/05/2020*

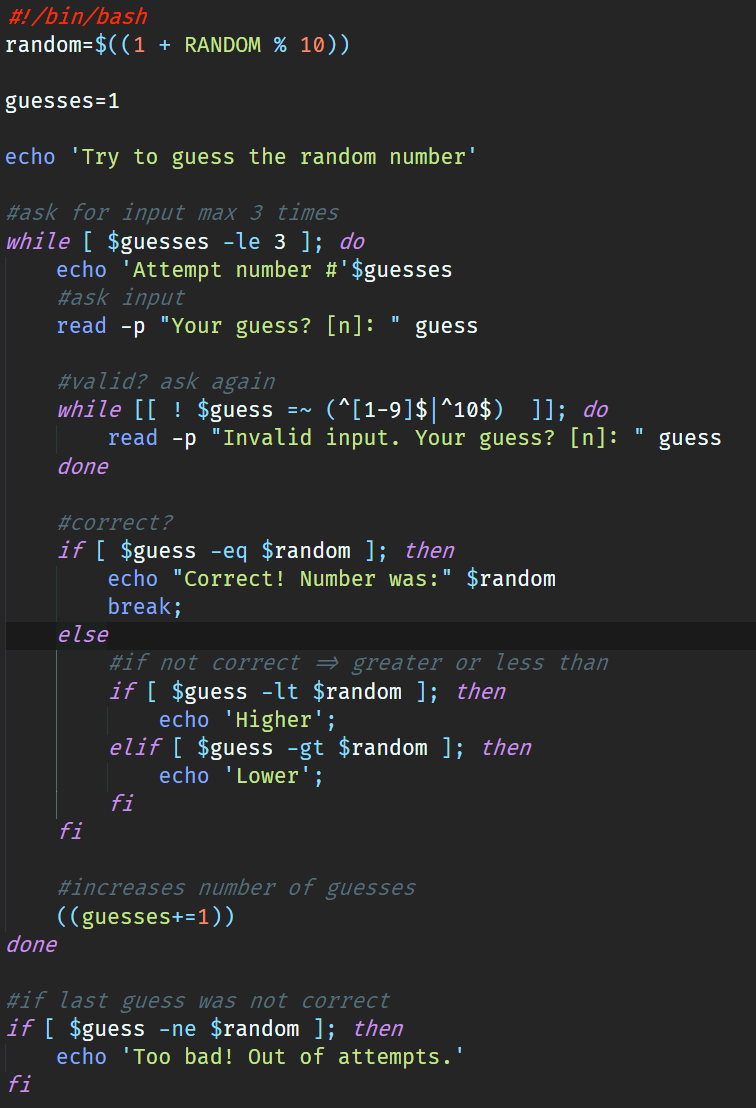
***Probleemstelling***

In dit labo is het de bedoeling om te leren werken met bash als scripting taal.

***LinkedIn Learning – Learning bash scripting***

******

***Bouw een raadspel***

******

Het script begint met de bash shebang, dit is het pad naar de bash interpreter.

Vervolgens gaan we een random nummer gaan genereren tussen 1 en 10 en we initialiseren het aantal guesses met 1. Het echte programma begint bij de while-lus. Deze gaat lopen zolang we geen 3 keer gegokt hebben. Vervolgens wordt het pogingnummer uitgeprint en wordt de input voor een gok gevraagd.

Daarna gaan we de input gaan valideren, dit is misschien het meest complexe deel van het script. Hiervoor maken we gebruik van een while-lus en een regular expression, zolang de input geen cijfer is tussen 1 en 10 gaat er een nieuwe input gevraagd worden. Dus letters of getallen niet binnen die grenzen liggen worden afgekeurd.

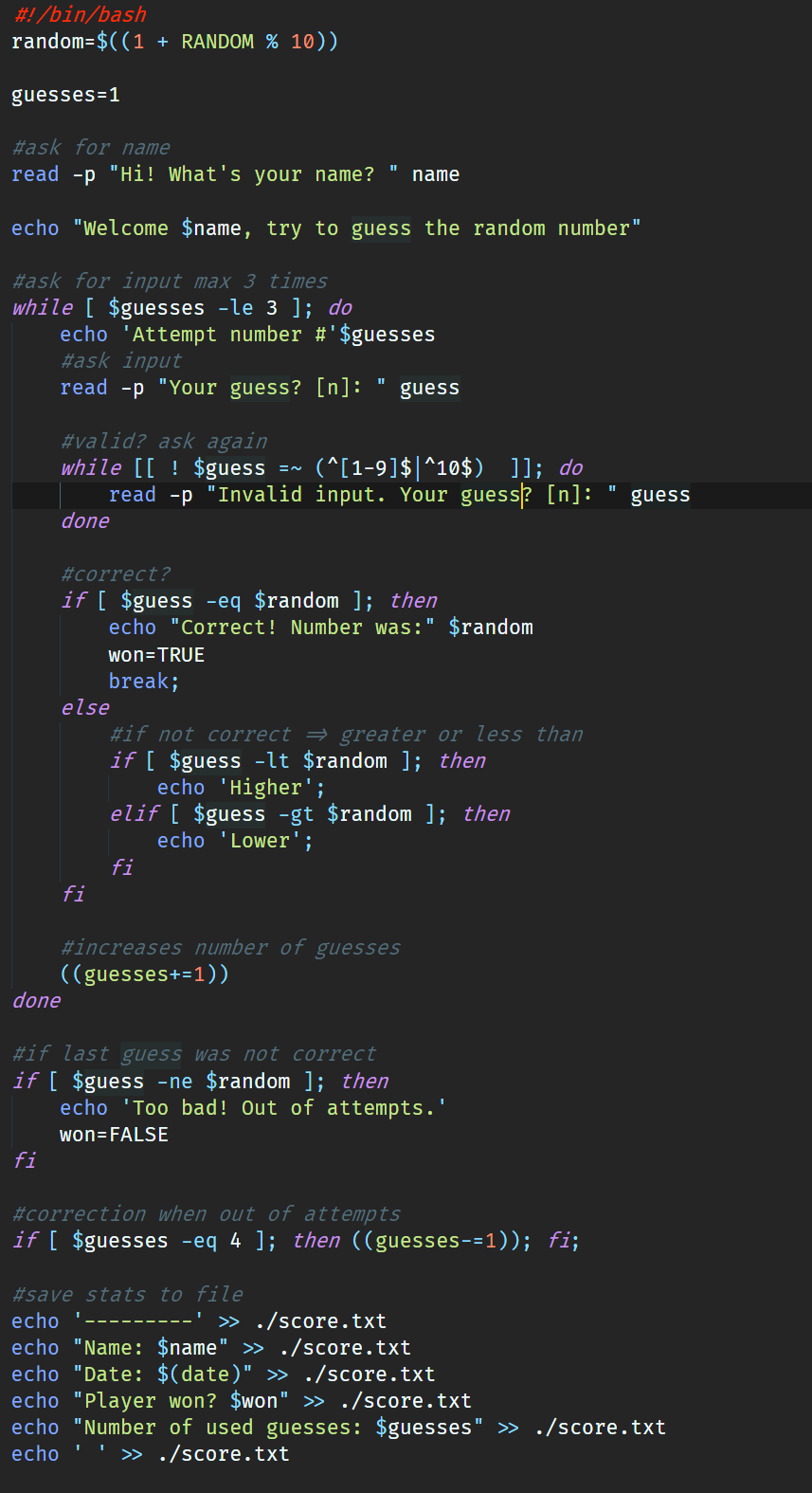
Indien de input wel geldig is gaan we checken of de input wel of niet het juiste getal is. Wanneer dit wel het geval is, gaan we dit uitprinten en met de break-instructie uit de while-lus stappen.

Wanneer de input niet correct was gaan we checken of de input hoger of lager moet zijn, en dit ook gaan uitprinten.

Tot slot gaan we het aantal pogingen verhogen met 1.

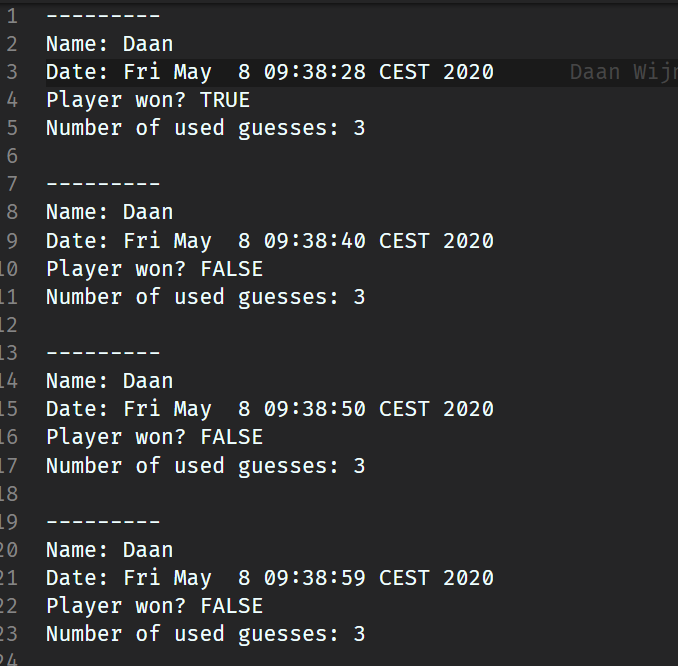
Wanneer er 3 pogingen gebeurt zijn, stappen we sowieso uit de while-lus en dan zeggen we dat het aantal pogingen op zijn. Maar vooraleer we dit doen, checken we nog eens of de poging niet het juiste antwoord was, omdat we bij een correct antwoord, we vroegtijdig de while-lus verlaten en dus ook dat bericht zouden tonen.

***Uitbreiding***



Voor de uitbreiding verandert er niet zo veel. We vragen in het begin de naam van de speler en tijdens het spel houden we de status van ‘won’ bij. Op het einde maken we nog een kleine correctie, men breekt uit de while-lus waneer het aantal pogingen groter is dan 3, dus moeten we dit nog corrigeren voor het logbestand. Tot slot loggen we alle informatie naar een tekstbestand.

Een mogelijk resultaat is in onderstaande figuur te zien.



***Conclusie***

De opdracht zelf is vrij vlot gelukt, ik had deze toevallig reeds voor een ander vak gemaakt. Ik herinner mij dat de grootste moeilijkheid zat in de regular expression om alle edge-cases op te vangen. In het algemeen had ik wel al wat ervaring met het schrijven van bash-scripts dus dit kwam ook wel van pas.

De repository is terug te vinden op Github: <https://github.com/WDaan/emb-bash-scripting>