E1 (Exercise1)

Write out only the Hit/Sunk log lines generated by Opponent for each available log file where Opponent was the starting player and they managed to sink the Player ship with a length of 2 first.

Reference implementation: Program.cs  
Documentation site: https://learn.microsoft.com/en-us/dotnet/api/?view=net-6.0

Rahmi: Bash script

Omar: PowerShell script

using System.Linq;

using System.Text.RegularExpressions;

string[] fileNames = Directory.GetFiles(".", "\*.log");

fileNames

.Where(file => {

string[] fileLines = File.ReadAllLines(file);

string firstPlayer = Regex.Replace(fileLines[0], @"\d+\t(\w+).+", "$1");

string firstSunkPlayerShip = fileLines.Where(line => Regex.IsMatch(line, @"Sunk\_\d")).First();

return firstPlayer == "Opponent" && firstSunkPlayerShip.IndexOf("Sunk\_2") >= 0;

})

.ToList()

.ForEach(file => {

string[] outputLines = File.ReadAllLines(file)

.Where(line => Regex.IsMatch(line, @"^.+Opponent.+(Hit|Sunk).+$"))

.ToArray();

File.WriteAllLines(Regex.Replace(file, @"\.log$", "\_exercise1.txt"), outputLines);

})

;

E2 (Exercise2)

Calculate the minimal number of tries (log entries) for Opponent between the first and last Hit(/Sunk) log line and copy the corresponding files (one or more).

Reference implementation: exercise2.sh  
Documentation site: https://linuxcommand.org/lc3\_man\_page\_index.php

Noble: C#

Rahmi: PowerShell script

#!/bin/bash

min=1000

for f in \*.log

do

fst=$(grep -P -n 'Hit\_\d' "${f}" | head -n1 | sed 's/:.\*//')

lst=$(grep -P -n '(Hit|Sunk)\_\d' "${f}" | tail -n1 | sed 's/:.\*//')

diff=$((lst - fst))

(( min > diff )) && min=$diff

done

for f in \*.log

do

fst=$(grep -P -n 'Hit\_\d' "${f}" | head -n1 | sed 's/:.\*//')

lst=$(grep -P -n '(Hit|Sunk)\_\d' "${f}" | tail -n1 | sed 's/:.\*//')

diff=$((lst - fst))

(( min == diff )) && cp "${f}" "${f%.log}\_exercise2.txt"

done

E3 (Exercise 3)

Transform all the timestamp and coordinate values to the discussed format and split each log file into 2 player-specific files based on which player generated a given line.

Reference implementation: exercise3.ps1  
Documentation site: https://learn.microsoft.com/en-us/powershell/module/

Omar: C#

Noble: Bash script

$files = Get-ChildItem -Path .\ -Filter \*.log

$files | ForEach-Object {

$\_props = @{}

$\_props.BaseName = $\_.BaseName

Get-Content $\_ |

ForEach-Object {

$\_ -match '^(\d+)(\t\w+\t)(\d)(:)(\d)(\t.+)$'

"$((Get-Date '1970-01-01 00:00:00.000Z').AddMilliseconds([long]$Matches[1]).ToLocalTime().ToString())" +

($Matches[2]) + ([char](65+[int]$Matches[3])) + ($Matches[4]) + ([int]$Matches[5]+1) + ($Matches[6])

} |

Where-Object { $\_ -match '^.+\t(\w+)\t\w:\d\t.+$' } |

Group-Object { $Matches[1] } |

ForEach-Object { $\_.Group | Set-Content "$($\_props.BaseName + '\_' + $\_.Name + '.txt')" }

}

# Notes

Recommended approach:

* Understanding the reference implementations at a granular level
* Using web search engines effectively to find relevant programming forums, tutorials, …
* Providing command/method arguments properly

Example scenarios

1. c# how to get the files with a given extension in a folder
2. <https://stackoverflow.com/questions/3152157/find-a-file-with-a-certain-extension-in-folder>
3. System.IO.Directory.GetFiles(path, "\*.txt");
4. **string[] fileNames = Directory.GetFiles(".", "\*.log")**
5. bash how to get the files with a given extension in a folder
6. <https://stackoverflow.com/questions/14505047/loop-through-all-the-files-with-a-specific-extension>
7. for i in \*.java
8. **for f in \*.log**
9. powershell how to get the files with a given extension in a folder
10. <https://stackoverflow.com/questions/45147547/get-only-file-with-given-extension-in-directory>
11. Get-ChildItem -Path $dir -Filter \*.xyz
12. **$files = Get-ChildItem -Path .\ -Filter \*.log**
13. bash how to store file contents in a variable
14. <https://stackoverflow.com/questions/14116748/need-to-assign-the-contents-of-a-text-file-to-a-variable-in-a-bash-script>
15. $(cat answer.txt)