

Linux vs Windows

En djupgående analys av prestandaskillnader mellan
Microsoft Windows och Manjaro Linux

Eddie Englund

VT 2020



NTI GYMNASIET GÄRDET

EXAMENSARBETE, APRIL 2020

**Student: Eddie Englund, eddie.englund@elev.ga.ntig.se även
eddie.englund@protonmail.com**

Handledare NTIG: Haris Kasumović, Haris.Kasumovic@ntig.se

Examinator NTIG: Haris Kasumović, Haris.Kasumovic@ntig.se

Sammanfattning

Abstract

Innehåll

| | |
|--------------------------------|---|
| Sammanfattning | 1 |
| Inledning | 1 |
| Bakgrund/Teori | 2 |
| Syfte/Frågeställning | 2 |
| Hypotes | 2 |
| Linux | 2 |
| Windows | 3 |
| Metod | 4 |
| Språk och program | 4 |
| Vad tror utvecklare? | 5 |
| Resultat | 6 |
| Formulär | 6 |
| Analys/diskussion | 7 |
| Slutsats | 7 |

Inledning

I och med att världen blir mer och mer digitaliserad för varje dag som går, så är det inte konstigt att det finns en massiv marknad med olika alternativ för nästintill allting i den digitaliserade världen. Allt från telefoner till hemdatorer eller kanske till och med robotdammsugare eller också en robotgräsklippare. Alla dessa digitaliserade mirakel har en sak gemensamt. Dem alla har ett operativsystem.

Världens mest kända operativsystem[3] är Microsoft Windows. Microsoft har en lång rad med olika versioner av sitt operativsystem[11]: Windows XP,

Windows Vista, Windows 7[...] och deras senaste (och förmodligen sista) operativsystem Windows 10.

Apple har flera olika operativsystem[6] som dom använder sig utav på sina olika produkter. I deras datorer så har man macOS, i deras telefoner har dom IOS och i deras nya Ipads så har dom iPadOS. Det finns faktisk ett tredje "operativsystem"; Linux är en så kallad Kernel [8] som är det man kan bygga ett operativsystem på. Linux har många olika operativsystem som är kallade för distributioner eller förkortat "*distros*". Linux är välkänt inom utvecklar världen eftersom att det är mycket vänligt för utvecklingsmiljöer men också eftersom att Linux har (oftast) en mycket god prestanda på maskiner med mindre processorkraft[9]. På grund av detta så är bl.a: iPadOS, IOS, macOS, Android och nästintill allting annat som inte har så mycket processorkraft som t.ex en hiss någon form a Linux som operativsystem.

Bakgrund/Teori

Syfte/Frågeställning

Undersökningens syfte är att ta reda på om Linux eller Windows har olika prestanda och varför. Är det så att dom har olika prestanda i olika typer av applikationer eller är det konstant? Undersökningen kommer att använda en rad olika tester för att ta reda på om hypotesen är san.

Hypotes

Utifrån mina egen upplevelse så är Linux (när man väl har konfigurerat det ordentligt) en mycket mer stabil och snabbare upplevelse en Microsoft Windows 10.

Linux

Linux är inte operativsystem. Däremot, så är Linux det som kallas för en "kernel" [8]. Det är hjärtat av operativsystemet eller kanske lite bättre jämfört med hjärnan av operativsystemet. Kerneln är en typ av mellanhand, mellan mjukvaran och hårdvaran. Den hanterar minnet och processer och även en del andra processer.

På grund utav att Linux är en kernel så finns det många så kallade distributioner/versioner (operativsystem). Skillnaderna på dem olika distributionerna kan variera högt, oftast handlar dom största skillnaderna om vilka

så kallad “repositories” man använder för att distribuera mjukvara. Dessutom så kan det handla om vilka filsystem man använder men också hur mycket eller hur lite förinstallerade program som finns när man installerar distributionen.

En annan fördel med Linux är att man har absolut kontroll. Man kan byta så kallade *window managers* eller förkortat *wm*[10]. Man kan även göra om Linux kerneln ifall man vill göra något specifikt eller kanske inte behöver bitar av den för att optimasera sin maskin. Detta är mycket vanligt inom låg processor kraftiga maskiner som hissar eller smart klockor.

Den största skillnaden mellan Linux och Windows är att Linux inte är proprietär och har öppen källkod vilket betyder att vem som helst kan bidra med kod för att; göra kerneln eller distributionen bättre och lägga till fler användbara funktioner, men även också att fixa buggar. Öppen källkod har även en annan fördel och det är att koden oftast inte är så kallad “*bloated*”, alltså att det finns kod eller funktioner som inte behövs eller att kod kvaliteten inte är bra. Detta leder till att prestandan på Linux är mycket hög. Den

distributionen som jag har valt att använda är Manjaro[7]. Manjaro är en så kallad *Arch based distro*. Den är baserad på en annan distro som heter Arch Linux som ofta blir kallat för den bästa distron. Däremot så gör Manjaro det lättare att komma igång och så har Manjaro dem flesta fördelarna som Arch Linux har.

Windows

Windows 10 är skrivet i programmeringspråket C# som är skapat av Microsoft[2]. Microsoft skapade C# 20 år sedan (2000) när dom ville skriva delar av Windows XP i Java men det fick dem inte eftersom att Oracle sa nej. Så dom utväcklade vad Klaus och Angelika Langer sa i en blog post att “Java and C# are almost identical programming languages. Boring repetition that lacks innovation”. [1] Windows är också ett operativsystem som är skrivet helt “in house” (det är inte öppen källkod).

Eftersom att det inte är öppen källkod så vet vi inte riktigt va Windows faktiskt håller på med i bakgrunden. Det gör så att man inte kan ändra på saker lika mycket som man kan på Linux. Dessutom så gör det att Microsoft kan göra lite vad det vill eftersom att det inte måste informera sina användare om alla ändringar (om inte lagen kräver det). Dessutom så är Windows ganska restriktivt i och med att man inte har absolut kontroll över bl.a när man

ska/vill uppdatera.

Däremot så finns det faktiskt en fördel med att inte ha öppen källkod. Microsoft kan skapa en virtuell monopoli över vissa marknader. Tex så är Windows (inom pc världen) det bästa operativsystemet att spela på. Det här är eftersom att dom flesta spelen är skapade för att köra på Windows. Det är eftersom att DirectX[4] finns på Windows men också eftersom att det inte är öppen källkod så är det inte särskilt lätt att piratkopiera spelen och deras innehåll.

Metod

Språk och program

För att ta reda på om det finns prestandaskillnader mellan Windows och Linux (manjaro) så behövde jag ta reda på vilket programmeringsspråk som vore relevant för att mäta prestandan mellan Windows och Linux(Manjaro) och en del andra saker.

1. Vilket språk bör användas och varför?
2. Vad bör man utföra för uppgift i programmet för att få maskinerna att arbeta?
3. Hur mäter vi prestandan?
4. Hur kan vi verifiera våra mätningar?

Den första frågan ledde till ett ganska djupt hål eftersom att det finns extremt många programmeringsspråk. Bara för att nämna ett par: Java, Javascript, Python, C, C++, C#, Ruby, Rust, mm.

För att kunna välja ett språk så vi kan börja med att utesluta dem långsamma språken. Då så utesluts: Ruby, Javascript och Python. Ruby, Javascript och Python är alla väldigt långsamma eftersom att (bl.a) så har dem inte multitråders kapabilitet (programmen kan inte göra flera saker samtidigt).

Python är väldigt långsamt eftersom att det var gjort för att vara lätt att skriva vilket skapar en hel del extra uppgifter som kompilator/runtiden måste utföra vilket gör den större. Detta gör att det tar längre tid att kompilera till maskinkod och därför tar det längre tid att utföra en uppgift.

Vi vill heller inte ha språk som måste bli *interpreted* i runtime. Då försvinner Java. Detta eftersom att Java har det som kallas för *JVM* vilket

är kort för “Java Virtual Machine”. Detta var skapat för att ha ett kompatibilitets lager så att man ska kunna köra Java program på alla datorer oavsett operativsystem.

Därefter så vore det också bra om man kan köra/skriva språket på Linux utan att lägga till extra kompatibilitets lager som Javas JVM. Då försvinner C# eftersom att för att köra det på Linux så behöver man det som kallas för “.Net Core” vilket fungerar som JVM och av samma anledning så vill vi inte ha det. Nu har vi endast C, C++ och rust kvar och eftersom att jag har erfarenhet med Rust och eftersom att Rust inte är lika svårt med bl.a minneshantering så slutade det med att jag valde Rust som språk för att skriva mitt program i.

Så nu har vi ett språk som vi vet är snabbt, stabilt och kan köras på flera maskiner (efter kompilering till den plattformen). Vad bör programmet utföra för uppgift?

Ett problem som uppstod när jag först försökte fylla en vektor med 1 gigabyte utav integers och sedan loopa programmet för att ha det igång så att jag kan verifiera att den gör det den ska, så la jag snabbt märke till att programmet inte tog i närheten så mycket som jag hade sagt åt språket att göra.

Så jag gjorde en del research och det visar sig att Rust till skillnad från C och C++ har lite så kallad “garbage collection” som är mycket smart. Om datan inte blir använd så är den datan inte registrerad utan data platsen i ram minnet är endast reserverat. Så jag fick ta ett steg tillbaka och tänka på vad jag kan göra istället.

Jag kom senare på en ide om att använda mig av ett bibliotek som heter “criterion” som är användbart för att göra så kallade “benchmarks”, alltså att mäta prestandan. Med det biblioteket så skrev jag ett kort program som kör fibonacci sekvensen och kalkylerar den. Det är en mycket tung process som börjar trycka på gränserna av 64bit. Efter det så kör jag prestanda testet 20 gånger och tar jag genomsnittet av alla test värden för att se till att testerna är precisa och exakta.

Vad tror utvecklare?

För att ta reda på vad andra utvecklare har för hypotes/teori om prestandan mellan Linux och Windows 10, varför prestandan skulle vara annorlunda, vilket operativsystem dom föredrar och varför. Så delades det runt ett formulär runt om kring olika programmerings/utvecklingsforum men även genom kontakter lyckats skicka vidare formuläret i professionella utvecklingsmiljöer.

Notera att formuläret var skrivet på Engelska för att kunna nå ut till så många utvecklare så möjligt.

OBS! Det fanns några frågor som var alternativa, till exempel: att motivera sina svar.

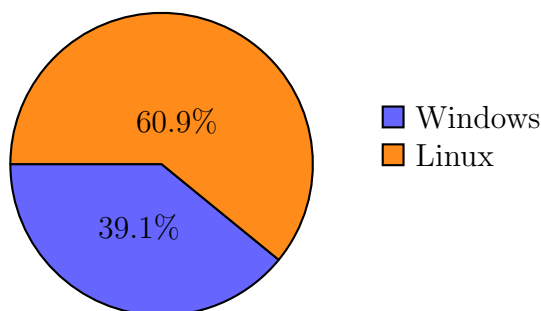
Vill ni läsa dessa motiveringar så finns dom här: <https://tinyurl.com/rccw2af>

Resultat

Formulär

Which operating system would have superior performance in tasks like, compiling code, executing code or even intensive workloads like working in a program like blender or unity?

I denna fråga så frågade vi olika utvecklare vad dom trodde kring Linux. Tror dom att Linux eller Windows har bättre prestanda när det kommer till att kompilera/köra/använda olika applikationer/program

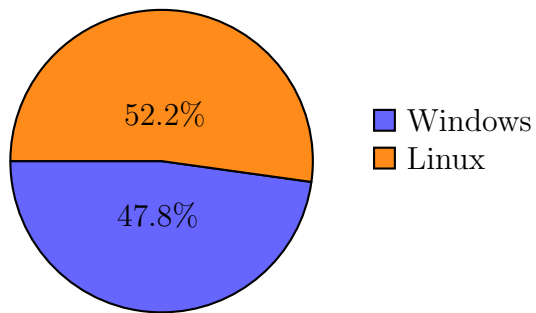


[5]

If you had the option to choose between Windows and Linux which one would you choose? (of course, you can choose any distro for Linux)

I Denna fråga så frågar vi utvecklare vilket operativsystem dom skulle välja om dom hade chansen. Vi fick ett ganska fascinerande resultat. Om

vi hade fått ett jämt antal svar så hade det varit en rak 50 50 på denna fråga men den tjugotredje personen som svarade på frågan valde Linux över Windows.



[5]

Analys/diskussion

Slutsats

Litteratur

- [1] *After Java and C# - what is next?* URL: <https://www.artima.com/weblogs/viewpost.jsp?thread=6543>. (Accessed on 2020-04-2).
- [2] *C sharp programming language*. URL: https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29. (Accessed on 2020-04-2).
- [3] *Desktop Windows Version Market Share Worldwide*. URL: <https://gs.statcounter.com/windows-version-market-share/desktop/worldwide/#monthly-201809-201908-bar>. (Accessed on 2020-04-4).
- [4] *DirectX*. URL: <https://en.wikipedia.org/wiki/DirectX>. (Accessed on 2020-04-4).
- [5] Eddie Englund. *Linux vs Windows*. URL: https://docs.google.com/spreadsheets/d/1-z11yr_POA4UBoPdx8Bt_0tTK3yrz2fDyEVU_jJIvb4/edit?usp=sharing. (Make sure to change this).
- [6] *List of Apple operating systems*. URL: https://en.wikipedia.org/wiki/List_of_Apple_operating_systems. (Accessed on 2020-04-4).
- [7] Manjaro. *Manjaro official website*. URL: <https://manjaro.org/>. (Accessed on 2020-02-25).
- [8] RedHat. *What is the Linux kernel?* URL: <https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel>. (Accessed on 2020-02-25).
- [9] *What is Linux?* URL: <https://www.linux.com/what-is-linux/>. (Accessed on 2020-04-4).
- [10] *Window Manager*. URL: https://en.wikipedia.org/wiki/Window_manager. (Accessed on 2020-04-4).
- [11] *Windows Versions*. URL: https://microsoft.fandom.com/wiki/Windows_Versions. (Accessed on 2020-03-22).