

Linux vs Windows

En djupgående analys av prestanda skillnader mellan
Microsoft Windows och Manjaro Linux

Eddie Englund

VT 2020



NTI GYMNASIET GÄRDET

EXAMENSARBETE/MASTER THESIS, 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
Bakrund	2
Vad tror utvecklare?	3
Metod	4
Resultat	5
Analys/diskussion	5
Slutsats	5

Inledning

I och med att världen blir mer och mer digitaliserad varje dag så är det inte konstigt att det finns en massiv marknad med olika alternativ för näst intill allting när det kommer till den digitaliserade världen. Allt från telefoner till hem datorer eller kanske till och med smart dammsugare eller också en smart gräsklippare. Alla dessa digitaliserade mirakel har en sak gemensamt. Dem alla har ett operativ system.

Världens mest kända operativsystem (inom allmänheten) är Microsoft Windows. Microsoft har en lång rad med olika versioner av sitt operativ system[4]: Windows XP, Windows Vista, Windows 7(...) och deras senaste (och förmodligen sista) operativ system Windows 10. I alla Apple datorer så har man macOS, i deras telefoner har dom IOS och i deras nya Ipads så har dom iPadOS. Det finns faktiskt ett tredje “operativsystem”; Linux är en så kallad Kernel [3] 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 världens eftersom att det är mycket vänligt för utvecklingsmiljöer men också eftersom att Linux har (oftast) en mycket god prestanda på gamla/sämmre maskiner. På grund av detta så är bl.a: iPadOS, IOS, macOS, Android och näst intill allting annat som inte har så mycket processor kraft som t.ex en hiss. Men reflekterars Linux kraftiga prestanda på mindre starka maskiner med lite CPU kraft likadant på dom starka maskinerna? Är prestandan bättre på “starka” maskiner som kör Linux som sin kernel?

Bakgrund

Linux är inte operativsystem. Däremot, så är Linux det som kallas för en “kernel” [3]. 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 men även också en del andra saker.

Men eftersom att Linux är en kernel så finns det många så kallade distributioner/versioner (operativsystem). Skillnaderna på distroen kan variera högt, men oftast handlar dom största skillnaderna om vilka så kallade “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.

Den distributionen som jag har valt att använda är Manjaro [2]. 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. Men, Manjaro gör det lättare att komma igång med och har dom flesta fördelarna med Arch.

Men den största skillnaden mellan Gnu/Linux och Windows är att det 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 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.

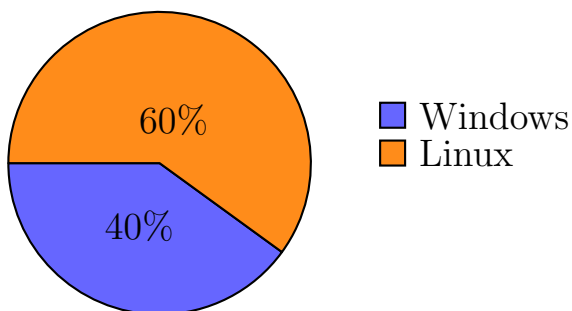
Därför så vill jag ta reda på om prestandan mellan Linux (manjaro) och Microsoft Windows 10 har en uppenbar skillnad och om dom har det, varför? Jag vill dessutom veta vad konsensusen är inom utvecklar miljöer.

Vad tror utvecklare?

För att ta reda på vad andra utvecklare har för hypotes/teori om prestandan mellan Linux i allmänhet 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.

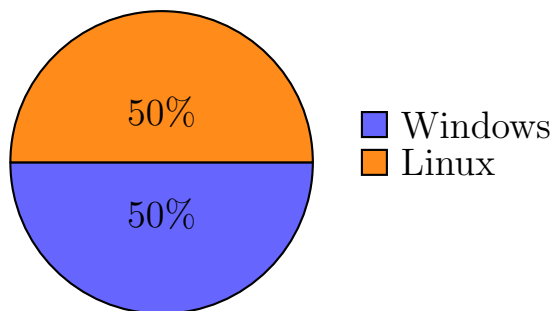
Notera att formuläret var skrivet på Engelska för att kunna nå ut till så många utvecklare så möjligt.

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?



[1]

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?



[1]

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>

Metod

För att ta reda på om det finns prestanda skillnader mellan Windows och Linux (manjaro) så behöver vi först ta reda på en del 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 programmerings språk. Bara för att nämna ett par: Java, Javascript, Python, C, C++, C#, Haskell, Ruby, Rust, mm. Okej, så vi har en hel del olika språk.

Så vi kan börja med utesluta dem långsamma språken. Direkt då så utesluts: Ruby, Javascript och Python. 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. 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 som Java så vill vi inte ha det. Nu har vi endast C, C++ och rust kvar och eftersom att jag har lite erfarenhet med rust och eftersom att rust inte är lika svårt när det kommer till saker som minnes hantering så slutade det med att jag valde Rust som språk.

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 vector med 1 gigabyte utav integers och sedan loopade programmet för att ha det igång så att jag kan verifiera att den gör det den ska. Jag la 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 registerad utan data platsen i ram minnet är endast reserverat. Så jag fick backa tillbaka ett steg 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. Sedan så gjorde jag ett kort program som kör fibonacci sekvensen och kalylerar den vilket är en tung process som börjar trycka på gränserna av 64bit.

Resultat

Analys/diskussion

Slutsats

Litteratur

- [1] 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).
- [2] Manjaro. *Manjaro official website*. URL: <https://manjaro.org/>. (Accessed on 2020-02-25).
- [3] RedHat. *What is the Linux kernel?* URL: <https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel>. (Accessed on 2020-02-25).
- [4] *Windows Versions*. URL: https://microsoft.fandom.com/wiki/Windows_Versions. (Accessed on 2020-03-22).