



Java

TDDC77

Objektorienterad Programmering

Föreläsning 0

Sahand Sadjadee
IDA, Linköpings Universitet

Outline

- ❑ **Kursinfo**
- ❑ **Introduktion till Programmering(1.1, 1.2)**
- ❑ **Linux**
- ❑ **Java Development Kit(JDK)**
- ❑ **Git**

Kursinfo● 1

Hemsida

- Hemsidan: <http://www.ida.liu.se/~TDDC77>
- Objektorienterad programmering.
- Föreläsningar, labbar, projekt, datorbaserad, tenta(TDDC77), basgruppsarbete(TDDC77).
- Vi använder Java under hela kursen.

Organisation

- 12 föreläsningar
- 6 hackathons
- **30 lärarledda laborationstillfällen**
- **18 lärarlösa laborationstillfällen**

Examination

- Kursmoment:
 - 6 laborationsuppgifter (HT1 och HT2). 3hp. LAB (U, G)
 - 1 programmeringsprojekt (HT2). 3hp. UPG(U,3,4,5)
 - 1 datorbaserad tenta. 2hp. DAT (U,3,4,5)(bara TDDC77)
 - Basgruppsarbete kring 8 vinjetter. BAS (U,G)(bara TDDC77)
- Godkänt på alla moment krävs för att få slutbetyget.
- Slutbetyget för TDDC77 bestäms av UPG och DAT.
- Slutbetyget för 729G90 bestäms av UPG.
- Examinationen sker alltid individuellt.
- **Arbetet på LAB och UPG sker i grupper av två studenter.**

Examination-regler

- Man får **INTE** lämna in kopierade, även modifierade, lösningar.
- Varje student måste kunna visa att hen förstått **ALLA** detaljer i sin lösning. Varje student kommer att få frågor individuellt under redovisningen.
- Varje student ska redovisa till läraren som är ansvarig för gruppen som studenten har registrerat sig i.
- Om någon form av fusk misstänks, är läraren skyldig att göra en anmälan till universitetets disciplinnämnd.
- Konsekvenserna kan bli varning eller tillfällig avstängning från studierna.

Föreläsningar

- Föreläsningarna sker 100% på plats.
- **Det är starkt rekommenderat att vara påläst i förväg på ämnena som tas upp på varje föreläsning.**

Labbtillfällen

- 5 labbuppgifter under första perioden, 1 labbuppgift under andra perioden.
- Labbarna ska göras i ordning.
- **Det är krav att labba i par.**

Webreg

- Webreg rapportering av LAB, UPG:
 - LAB: i grupper av två personer innan 2024-09-02,
<https://www.ida.liu.se/webreg3/TDDC77-2024-1/LAB>
<https://www.ida.liu.se/webreg3/729G90-2024-1/LAB1>
 - UPG: i grupper av två personer innan 2024-09-02,
<https://www.ida.liu.se/webreg3/TDDC77-2024-1/UPG>
<https://www.ida.liu.se/webreg3/729G90-2024-1/UPG4>

Frågor

- Frågor som rör vad ni lär er i kursen kan ställas till handledarna eller examinatorn.
- Övriga frågor, problem, administrativa ärenden osv kan ni diskutera med examinatorn.

Frågor?

Programming { }

Datorer överallt



Hårdvara VS Mjukvara

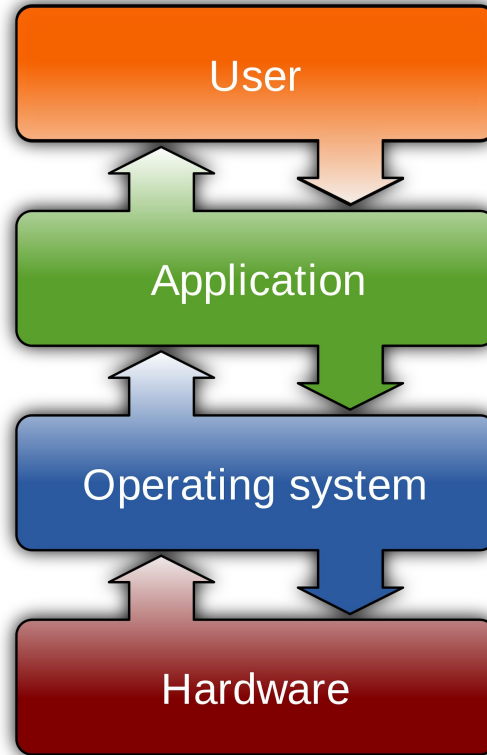
- **Hårdvara** är fysiska delarna som en dator består av. Till exempel, tangentbord, hårddisk, processor, skärm och så vidare.
- Hårdvara existerar fysiskt.
- **Mjukvara** är **instruktioner/kommandon** som **exekveras** av processorn, CPU. Med andra ord styrs datorer av mjukvara. Till exempel, Adobe Photoshop, Linux, Chrome och så vidare.
- Mjukvara existerar **INTE** fysiskt.

Operativsystem

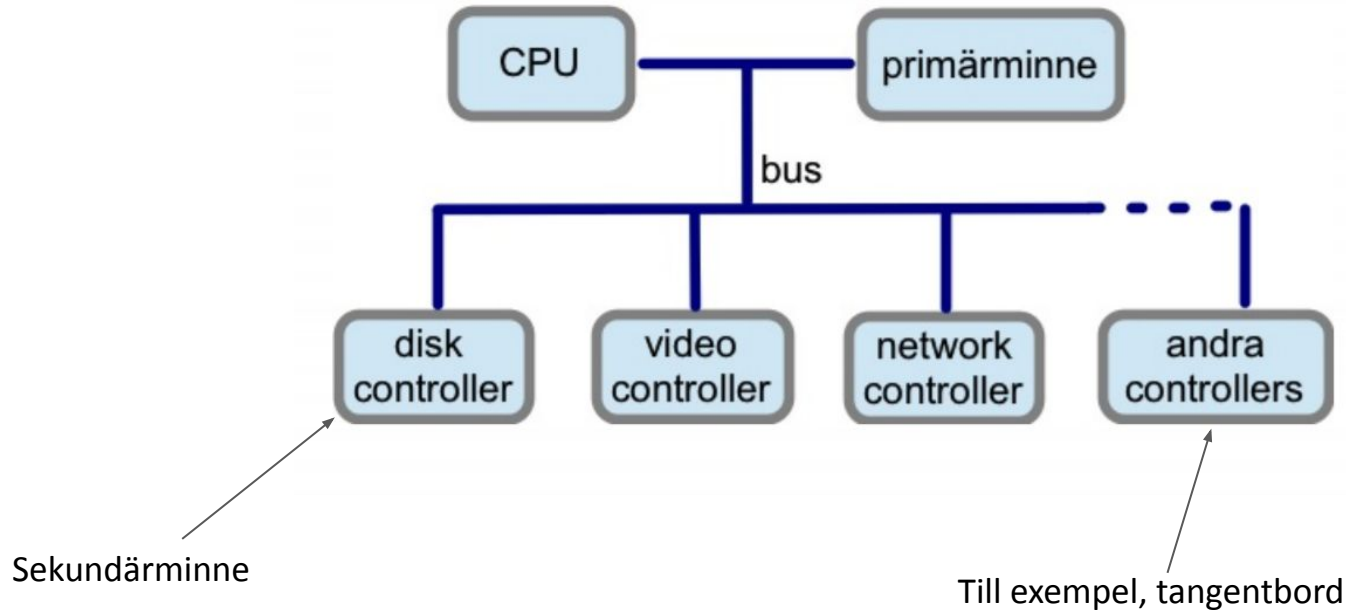
- är en mjukvara.
- ligger mellan Hårdvaran och andra mjukvaror, applikationer.
- är ansvarig för att dela resurser, till exempel processorn, mellan olika applikationer som körs samtidigt.
- är ansvarig för att **köra** applikationer.



Operativsystem



En vanlig dator

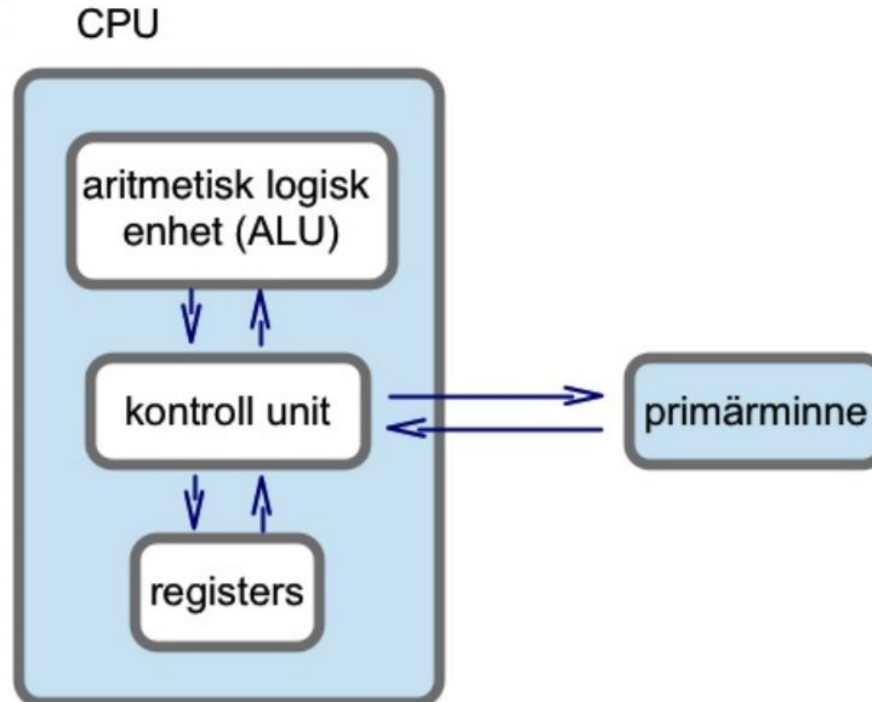


Minnet... 8 bits ger 1 byte

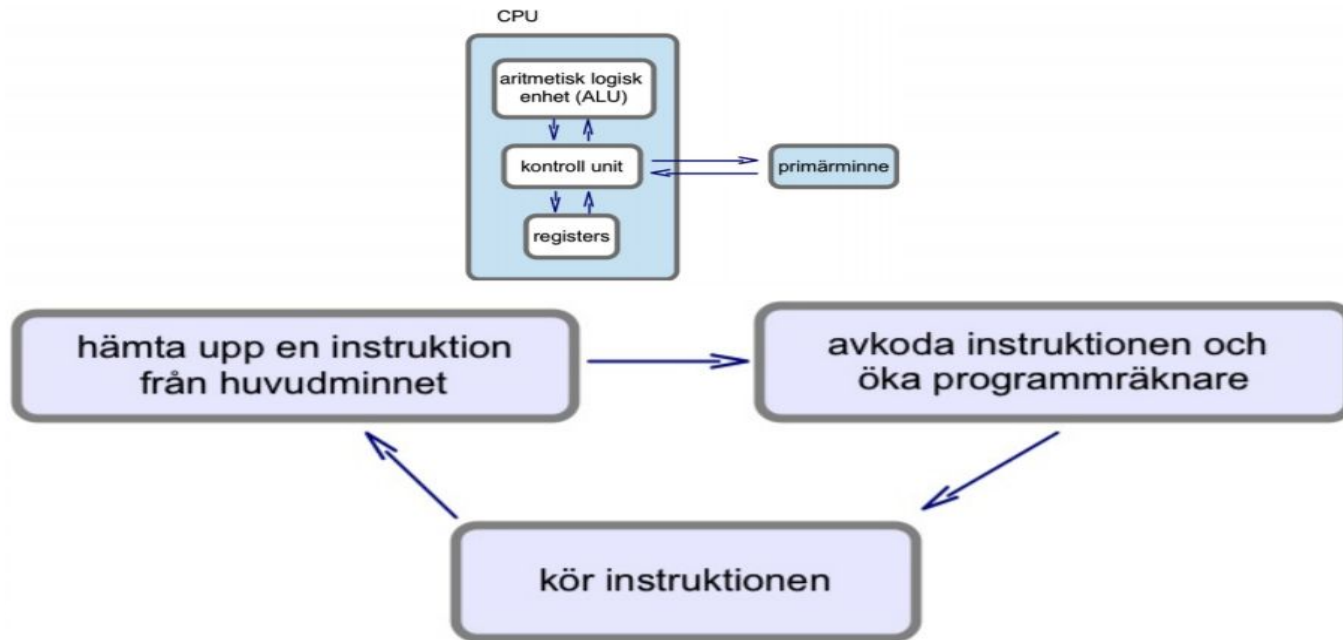


byte	B	$2^0 = 1$	10^0	B	byte
Kibibyte	KiB	$2^{10} = 1024$	10^3	KB	Kilobyte
Mebibyte	MiB	$2^{20} = 1,048,576$	10^6	MB	Megabyte
Gibibyte	GiB	$2^{30} = 1,073,741,824$	10^9	GB	Gigabyte
Tebibyte	TiB	$2^{40} = 1,099,511,627,776$	10^{12}	TB	Terabyte
Pebibyte	PiB	$2^{50} = 1,125,899,906,842,624$	10^{15}	PB	Petabyte

CPU...



Hämta-avkoda-kör cykeln



Maskinkod

```
80483b4: 55          push    %ebp
80483b5: 89 e5       mov     %esp,%ebp
80483b7: 83 e4 f0    and     $0xffffffff0,%esp
80483ba: 83 ec 20    sub     $0x20,%esp
80483bd: c7 44 24 1c 00 00 00 movl    $0x0,0x1c(%esp)
80483c4: 00
80483c5: eb 11       jmp     80483d8 <main+0x24>
80483c7: c7 04 24 b0 84 04 08 movl    $0x80484b0,(%esp)
80483ce: e8 1d ff ff ff call    80482f0 <puts@plt>
80483d3: 83 44 24 1c 01 addl    $0x1,0x1c(%esp)
80483d8: 83 7c 24 1c 09 cmpl    $0x9,0x1c(%esp)
80483dd: 7e e8       jle     80483c7 <main+0x13>
80483df: b8 00 00 00 00 mov     $0x0,%eax
80483e4: c9         leave
80483e5: c3         ret
80483e6: 90         nop
80483e7: 90         nop
80483e8: 90         nop
80483e9: 90         nop
80483ea: 90         nop
```

- Förstås bara av processorn.
- Det enda språket som processorn förstår.
- Maskinkod är ett lågnivåspråk.

Maskinkod

Problem:

- Man måste kunna procesorns arkitektur.
- Skrivna programmet baserat på en viss CPU kan inte köras av andra typer av CPU.

Högnivåspråk

- Java, C++, PHP, Javascript, ...
- Programmet körs inte direkt av processor:n
- Programmet omvandlas, **Kompileras/tolkas**, till **maskinkod**. Det är en liten skillnad när det gäller Java? Vad är skillnaden?
- Det behövs inte kunna processorns detaljer när högnivåspråk används.

Interpretera/Kompilera

Källkod

```
import java.util.Scanner;  
  
public class JamforTva{  
  
    public static void main (String[] args){  
        Scanner in = new Scanner (System.in);  
        int x, y;  
        System.out.println ("Mata in ett heltal: ");  
        ...  
    }  
}
```

Interpretator (interpreter)

Ett program som exekvera eller utför vad källkoden, eller en intermediär kod, beskriver



Källkod

```
import java.util.Scanner;  
  
public class JamforTva{  
  
    public static void main (String[] args){  
        Scanner in = new Scanner (System.in);  
        int x, y;  
        System.out.println ("Mata in ett heltal: ");  
        ...  
    }  
}
```

Kompilator

Översätta källkoden till maskinkod

Maskinkod

07AF00973278



Olika sorters program

- Webbapplikationer
- Mobilapplikationer
- **Desktopapplikationer(Terminal-baserade och GUI-baserade program)**
- Bibliotek(program som används av andra program)
- inbyggda program(bromssystem i bilar)
- ...

- Analysera din uppgift och försök hitta en lösning i ett språk som du kan, t.ex. Svenska!
- Översätt din lösning till programkod, t.ex. Java.
- Försök inte att lösa allt på en gång.

Att bryta ner en uppgift i små steg

Låt användaren skriva in två heltal x och y . Vi vill räkna upp skillnadens absoluta värde (dvs, $|x - y|$) och skriva ut resultatet.

1. Fråga användaren efter ett heltal
2. Läs in heltalet x
3. Fråga användaren efter ett heltal
4. Läs in heltalet y
5. Om x är större än y , räkna upp $(x - y)$
6. Om y är större än x , räkna upp $(y - x)$
7. Skriv ut resultatet

Flowchart Diagram

Används för att skissa/grafiskt presentera algoritmer.



Terminator

Indicates the beginning or end of a program flow in your diagram.



Process

Indicates any processing function.



Decision

Indicates a decision point between two or more paths in a flowchart.



Delay

Indicates a delay in the process.



Data

Can represent any type of data in a flowchart.



Document

Indicates data that can be read by people, such as printed output.



Multiple documents

Indicates multiple documents.



Subroutine

Indicates a predefined (named) process, such as a subroutine or a module.



Preparation

Indicates a modification to a process, such as setting a switch or initializing a routine.



Display

Indicates data that is displayed for people to read, such as data on a monitor or projector screen.



Manual input

Indicates any operation that is performed manually (by a person).



Manual loop

Indicates a sequence of commands that will continue to repeat until stopped manually.



Loop limit

Indicates the start of a loop. Flip the shape vertically to indicate the end of a loop.



Stored data

Indicates any type of stored data.



Connector

Indicates an inspection point.



Off-page connector

Use this shape to create a cross-reference and hyperlink from a process on one page to a process on another page.



Off-page connector



Off-page connector



Off-page connector



Or
Logical OR



Summing junction

Logical AND



Collate

Indicates a step that organizes data into a standard format.



Sort

Indicates a step that organizes items list sequentially.



Merge

Indicates a step that combines multiple sets into one.



Database

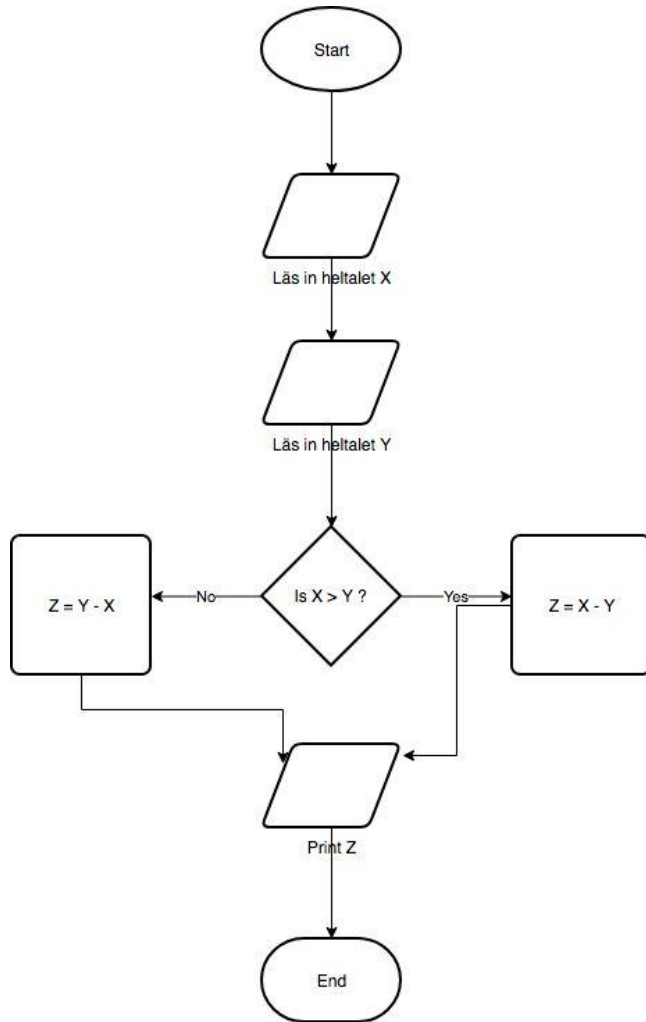
Indicates a list of information with a standard structure that allows for searching and sorting.



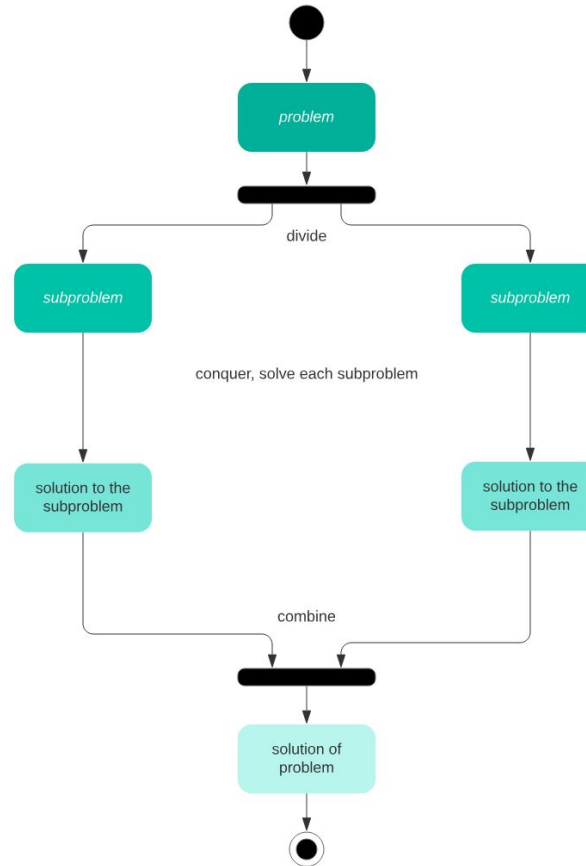
Internal storage

Indicates an internal storage device.

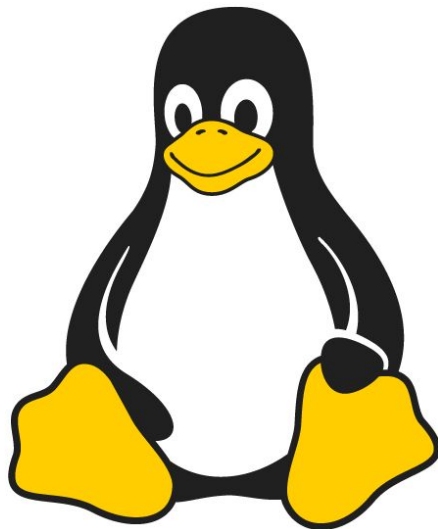
Flowchart



Divide and Conquer



Linux



Linux

- Operativsystem är en mjukvara som ligger mellan hårdvaran och användaren.
- Varje dator behöver ett operativsystem så att den kan användas.
- Linux är ett operativsystem gjort av Linus Torvalds.
- Det är olika företag som släpper olika distributioner/versioner av Linux så som Ubuntu, Redhat, Suse, Arch osv.



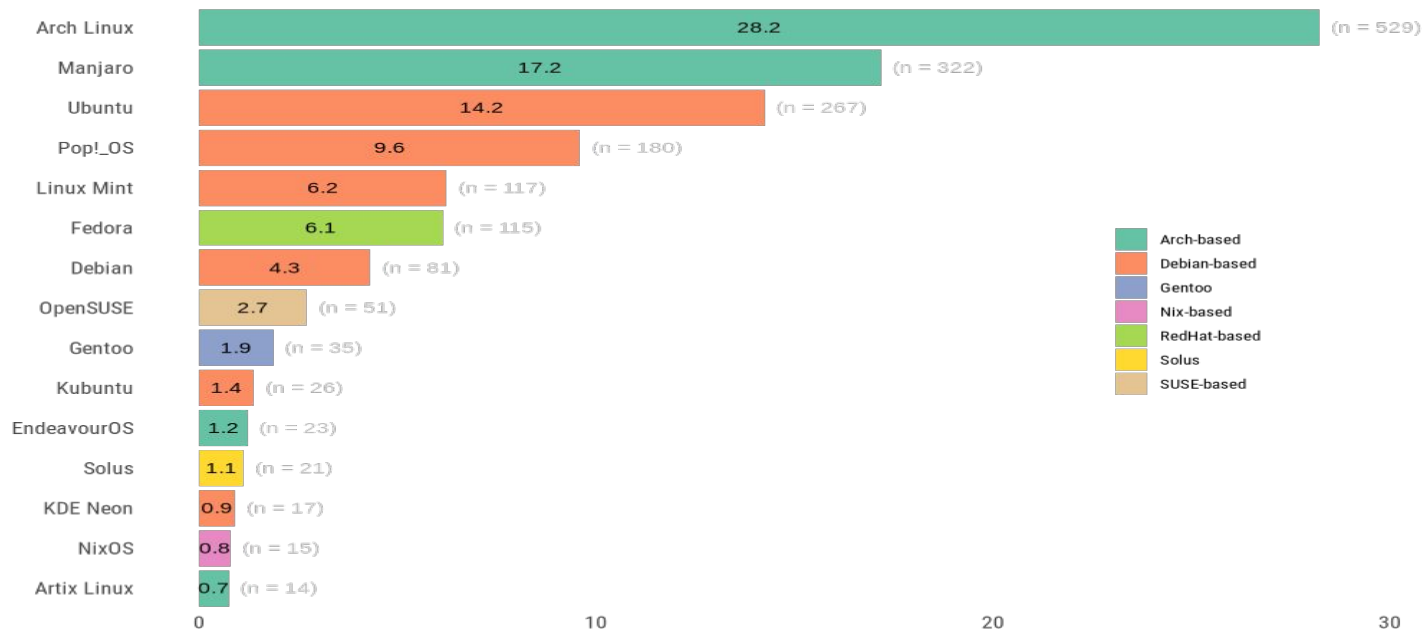
Linux-distributioner



Linux-distributioner

Representation of Each Linux Distribution in the Survey (in %) - Top 15

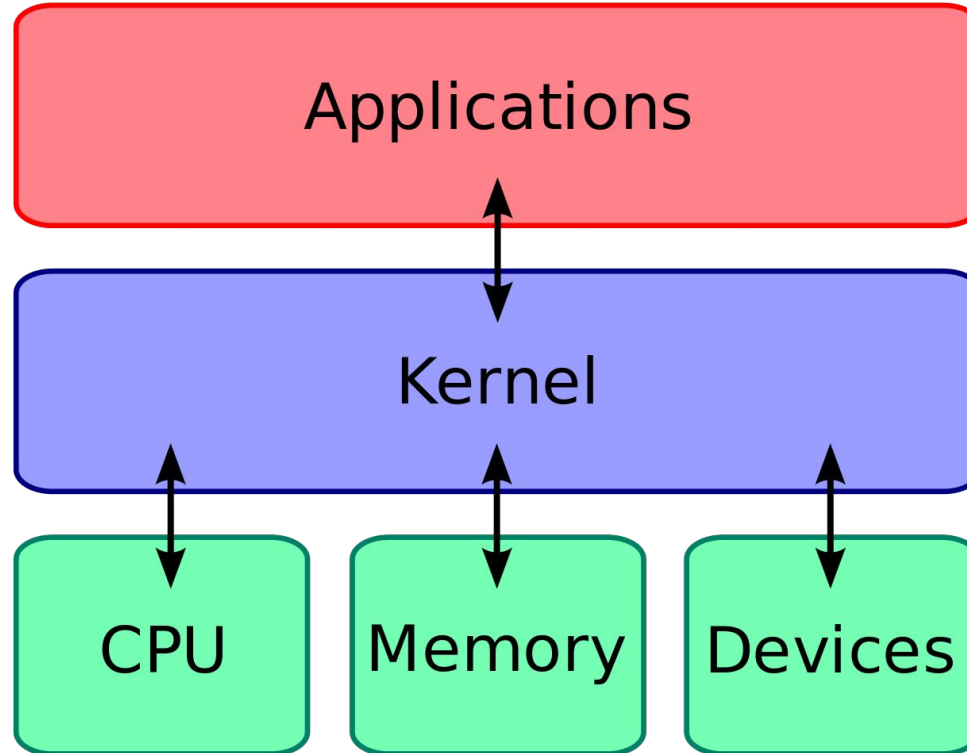
Source: Linux Gamers Survey Q2 2021 by Boiling Steam (n=1874)



Linux

- Linux-distributioner är lika förutom detaljer om installation, paketering och konfiguration. Standard gränssnitt är också olika i olika distributioner.
- “Kernel” som är kärnprogramvaran är samma i olika distributioner.

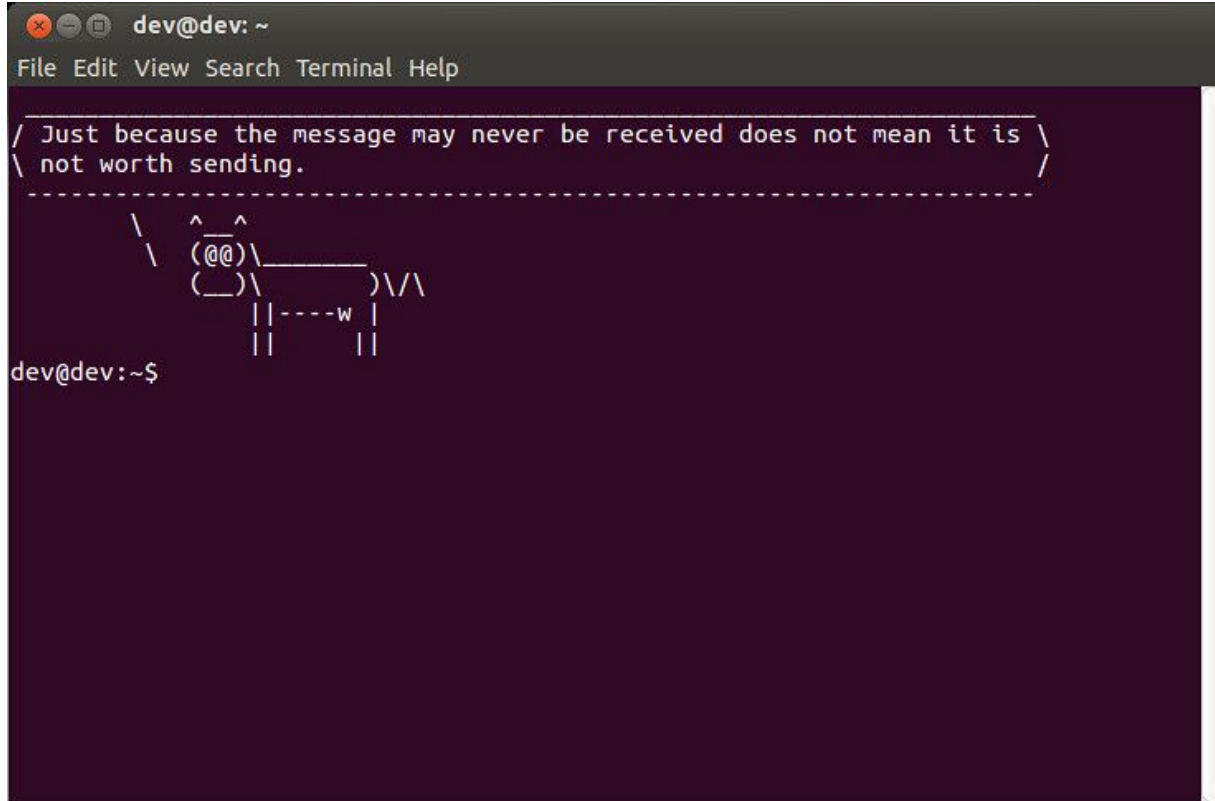
Linux-kernel



Linux-terminalen

- De flesta operativsystem innehåller ett program där man kan skriva kommandon att exekveras.
- I Linux och Mac OSX kallas det för “Terminal” eller “Shell”
- I Windows kallas det för “Command prompt”.

Linux-terminalen



```
dev@dev: ~  
File Edit View Search Terminal Help  
/ Just because the message may never be received does not mean it is \  
\ not worth sending. \  
-----  
      ^ ^  
      (@@)\_____  
      (___)\      )\ \  
          ||----w  ||  
          ||      ||  
dev@dev:~$
```

The screenshot shows a Linux terminal window with a dark background. The title bar at the top reads "dev@dev: ~" and includes standard window control buttons. Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main content area displays a quote enclosed in a box defined by slashes and a dashed line. Below the quote is a diagram consisting of a horizontal line with a vertical line extending downwards from its center, and a wavy line extending to the right from the vertical line. The prompt "dev@dev:~\$" is visible at the bottom left of the terminal.

Linux-terminalen



Top 50+ Linux Commands You MUST Know

Top 50 Linux Commands you must know

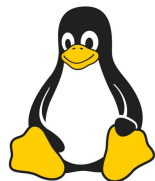


1. is	1. clear	1. diff	1. kill and killall	1. apt, pacman, yum, rpm
2. pwd	2. echo	2. cmp	2. df	2. sudo
3. cd	3. less	3. comm	3. mount	3. cal
4. mkdir	4. man	4. sort	4. chmod	4. alias
5. mv	5. unman	5. export	5. chown	5. dd
6. cp	6. whoami	6. zip	6. ifconfig	6. whereis
7. rm	7. tar	7. unzip	7. traceroute	7. whatis
8. touch	8. grep	8. ssh	8. wget	8. top
9. in	9. head	9. service	9. ufw	9. useradd
10. cat	10. tail	10. ps	10. iptables	10. passwd

<https://www.digitalocean.com/community/tutorials/linux-commands>

Linux-terminalen

- Vissa kommandon är bara ord som förstås av terminalen. Sådana kommandon kallas för “interna/inbyggda kommandon” eller på engelska “internal/built-in commands”. Till exempel, `pwd`, `cd`, `exec`, `logout` osv.
- De flesta kommandona är program/applikationer som ligger på hårddisk och kan köras i terminalen. Sådana kommandon kallas för “externa kommandon” eller på engelska “external commands”.
- Externa kommandon ligger mest i `/bin`, `/sbin`, `/usr/sbin` mappar.
- Varje Java program som vi utvecklar i kursen exekveras i terminalen som ett externt kommando.



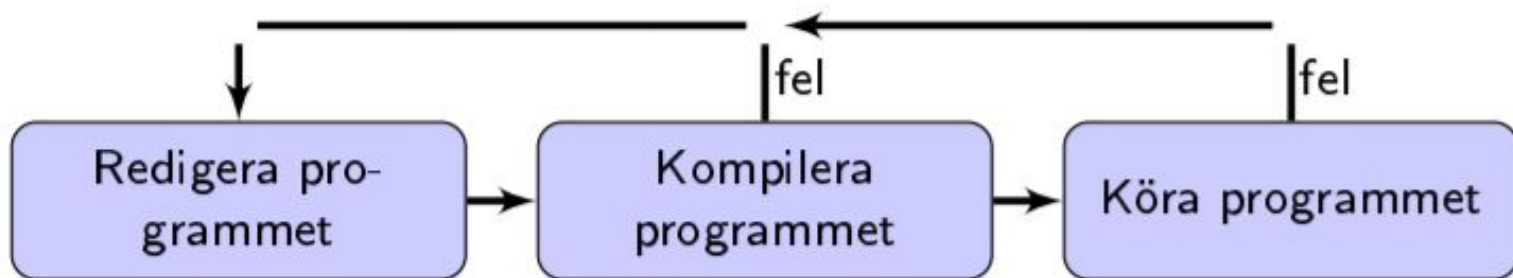
Demo



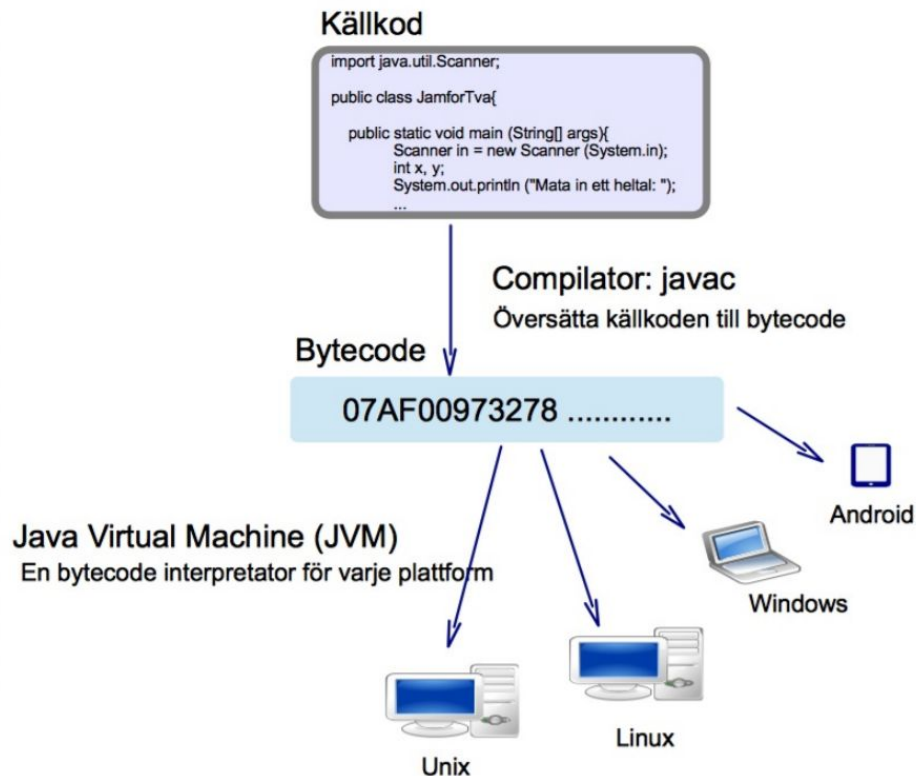
Java™ Development Kit(JDK)

Kompilera och köra program

1. Skapa `Hej.java`, skriv programmet och sedan spara filen.
2. Kompilera med `javac Hej.java`
3. Rätta fel och repetera tills du lyckas med att kompilera programmet.
4. Kör programmet med `java Hej`



Kompilera och köra program



Java Virtuell Maskin

- En virtuell maskin är en mjukvara som simulerar en dator.
- Java Virtuell Maskin är en simulerad dator som förstår och exekverar Bytecode.
- OS sitter mellan JVM och hårdvaran.

En bra artikel om JVM:

<https://www.javaworld.com/article/3272244/core-java/what-is-the-jvm-introducing-the-java-virtual-machine.html>

JDK(Java Development Kit)

- JDK är en kit/mapp som innehåller verktyg för att:
 - Kompilera
 - Köra
 - Debugga
 - Övervaka Java virtuella maskiner
 - Generera dokumentation
- Innehåller bibliotek för att utföra grundläggande uppgifter. Till exempel:
 - Utmaning, Inmatning
 - Matematiska beräkningar
 - Processera filer
- Innehåller JVM

Version	class file format version ^[8]	Release date	End of Free Public Updates ^{[9][10][11][12][13][14][15][16]}	Extended Support Until
JDK 1.0	45	23rd January 1996	May 1996	—
JDK 1.1	45.3	2nd February 1997	October 2002	?
J2SE 1.2	46	4th December 1998	September 2003	?
J2SE 1.3	47	8th May 2000	October 2010	?
J2SE 1.4	48	13th February 2002	October 2008	February 2013
Java SE 5	49	29th September 2004	November 2009	April 2015
Java SE 6	50	11th December 2006	April 2013	December 2018 for Oracle ^[9] December 2026 for Azul ^[12]
Java SE 7	51	28th July 2011	September 2022 for OpenJDK Maintained by Oracle until May 2015 ^[17] , Red Hat until August 2020 ^[18] and Azul until September 2022 ^[19]	July 2022 for Oracle ^[9] June 2020 for Red Hat ^[13] December 2027 for Azul ^[12]
Java SE 8 (LTS)	52	18th March 2014	(OpenJDK currently maintained by Red Hat) ^[20] March 2022 for Oracle (commercial) December 2030 for Oracle (non-commercial) December 2030 for Azul ^[12] May 2026 for IBM Semeru ^[14] At least May 2026 for Eclipse Adoptium ^[10] At least May 2026 for Amazon Corretto ^[11]	December 2030 for Oracle ^[9] November 2026 for Red Hat ^[13]
Java SE 9	53	21st September 2017	March 2018 for OpenJDK	—
Java SE 10	54	20th March 2018	September 2018 for OpenJDK	—
Java SE 11 (LTS)	55	25th September 2018	(OpenJDK currently maintained by Red Hat) ^[21] September 2026 for Azul ^[12] October 2024 for IBM Semeru ^[14] At least October 2024 for Eclipse Adoptium ^[10] At least September 2027 for Amazon Corretto ^[11] At least October 2024 for Microsoft ^{[22][15]}	September 2026 for Oracle ^[9] September 2026 for Azul ^[12] October 2024 for Red Hat ^[13]
Java SE 12	56	19th March 2019	September 2019 for OpenJDK	—
Java SE 13	57	17th September 2019	(OpenJDK currently maintained by Azul) ^[23] March 2023 for Azul ^[12]	—
Java SE 14	58	17th March 2020	September 2020 for OpenJDK	—
Java SE 15	59	16th September 2020	(OpenJDK currently maintained by Azul) ^[24] March 2023 for Azul ^[12]	—
Java SE 16	60	16th March 2021	September 2021 for OpenJDK	—
Java SE 17 (LTS)	61	14th September 2021	(OpenJDK currently maintained by SAP) ^[25] September 2029 for Azul ^[12] October 2027 for IBM Semeru ^[14] At least September 2027 for Microsoft ^[15] At least September 2027 for Eclipse Adoptium ^[10]	September 2029 or later for Oracle ^[9] September 2029 for Azul ^[12] October 2027 for Red Hat ^[13]
Java SE 18	62	22nd March 2022	September 2022 for OpenJDK and Adoptium	—
Java SE 19	63	20th September 2022	March 2023 for OpenJDK	—
Java SE 20	64	21st March 2023	September 2023 for OpenJDK	—
Java SE 21 (LTS)	65	19th September 2023	September 2028	September 2031 for Oracle ^[9]
Java SE 22	?	March 2024	September 2024 ^[16]	—
Legend: Old version Older version, still maintained Latest version Future release				

JDK

<https://docs.oracle.com/en/java/javase/11/docs/api/index.html>

- jdk1.7.0
 - bin
 - java*
 - javac*
 - javap*
 - javah*
 - javadoc*
 - lib
 - tools.jar
 - dt.jar
 - jre
 - bin
 - java*
 - lib
 - applet
 - ext
 - localedata.jar
 - fonts
 - security
 - sparc
 - server
 - client
 - rt.jar
 - charsets.jar

Att jobba med hemkatalogen

- Hemifrån under windows/mac/linux:
 - Du kan använda Thinlinc för att logga in på ditt konto på universitet
 - läs mer om thinlinc på IDA genom att [klicka här](#).
- Hemifrån under windows/mac/linux:
 - Du kan också använda en ssh klient istället för thinlinc.
 - [klicka här](#) för mer information.
- I labbsalarna har ni tillgång till hemkatalogen automatiskt utan att behöva använda något ytterligare verktyg.

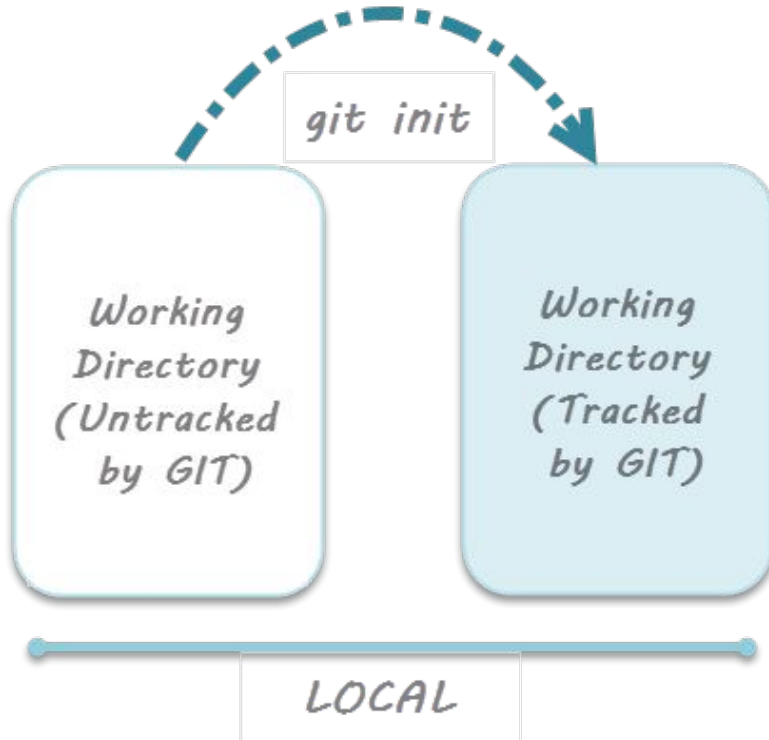


Git

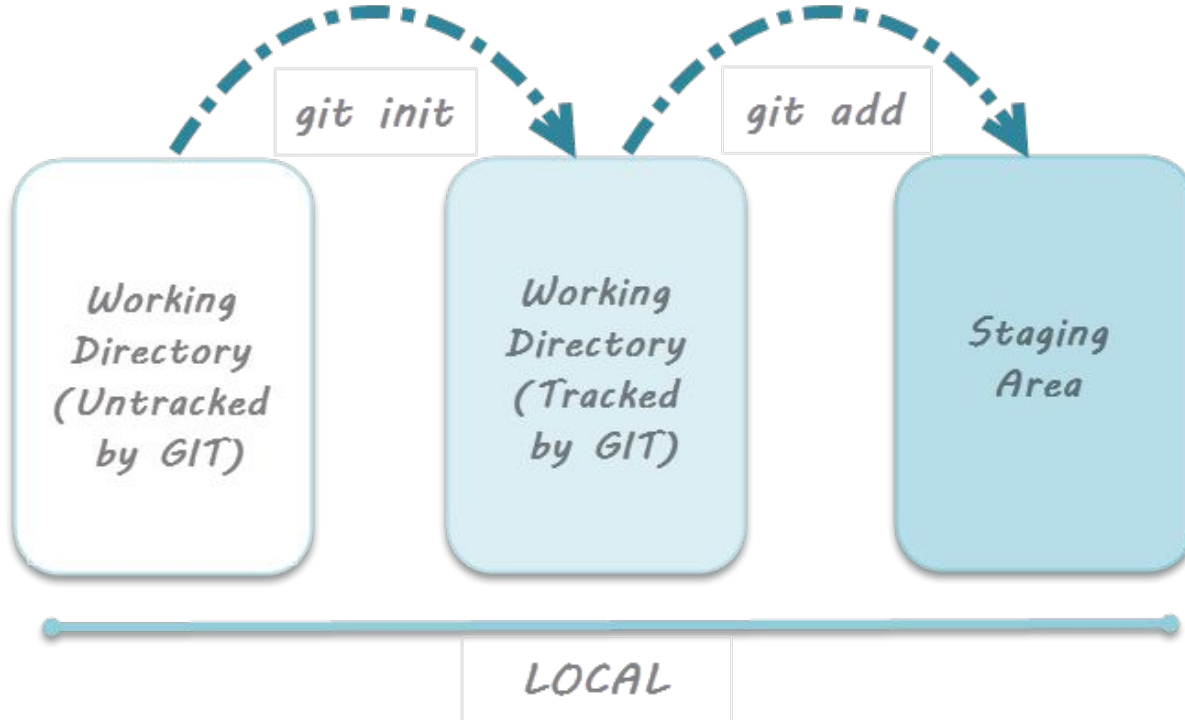
- “Git är en mjukvara som spårar ändringar i filer under utveckling. Det används främst inom samarbeten där flera utvecklare vill kunna arbeta med samma kodbas men går också bra att använda för att versionshantera sina egna projekt.”
- Git är ett program som ska först laddas ner och installeras:

<https://git-scm.com/downloads>

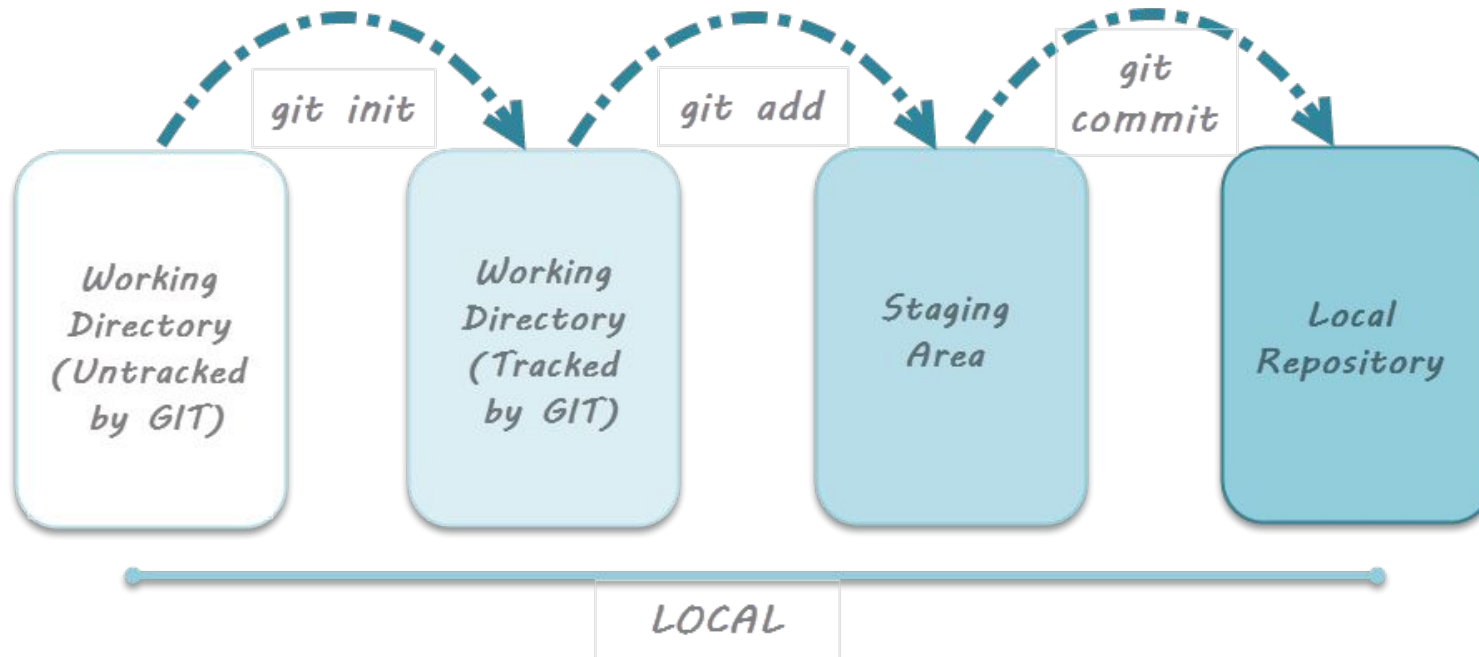
Git



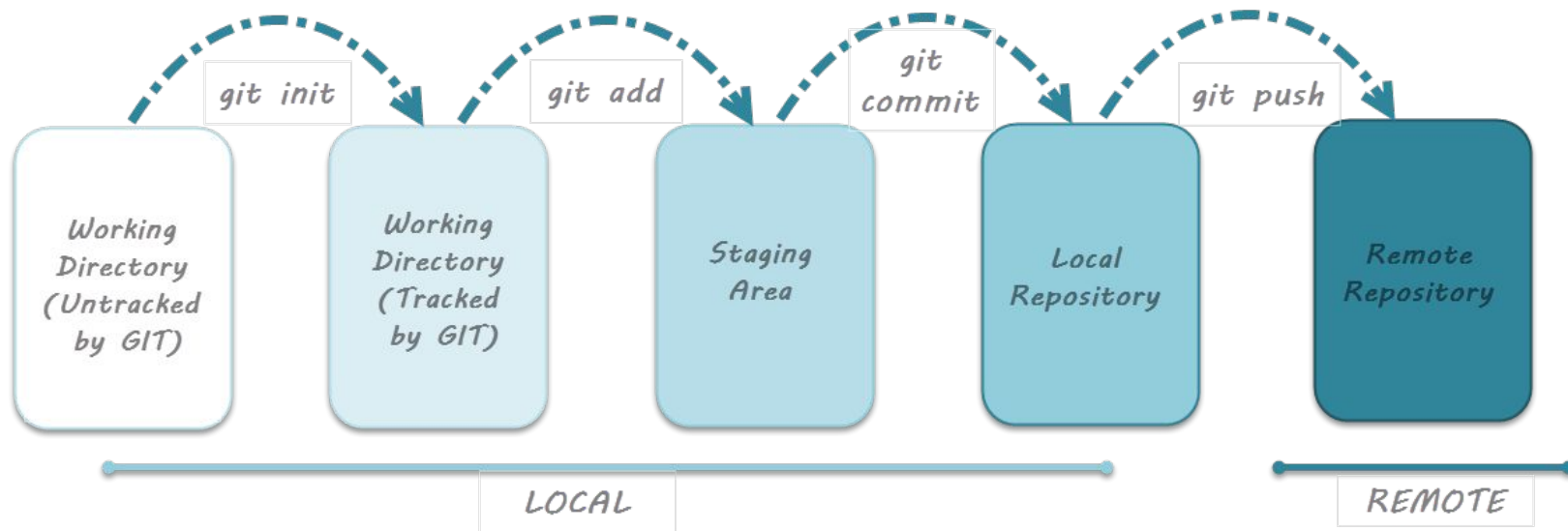
Git



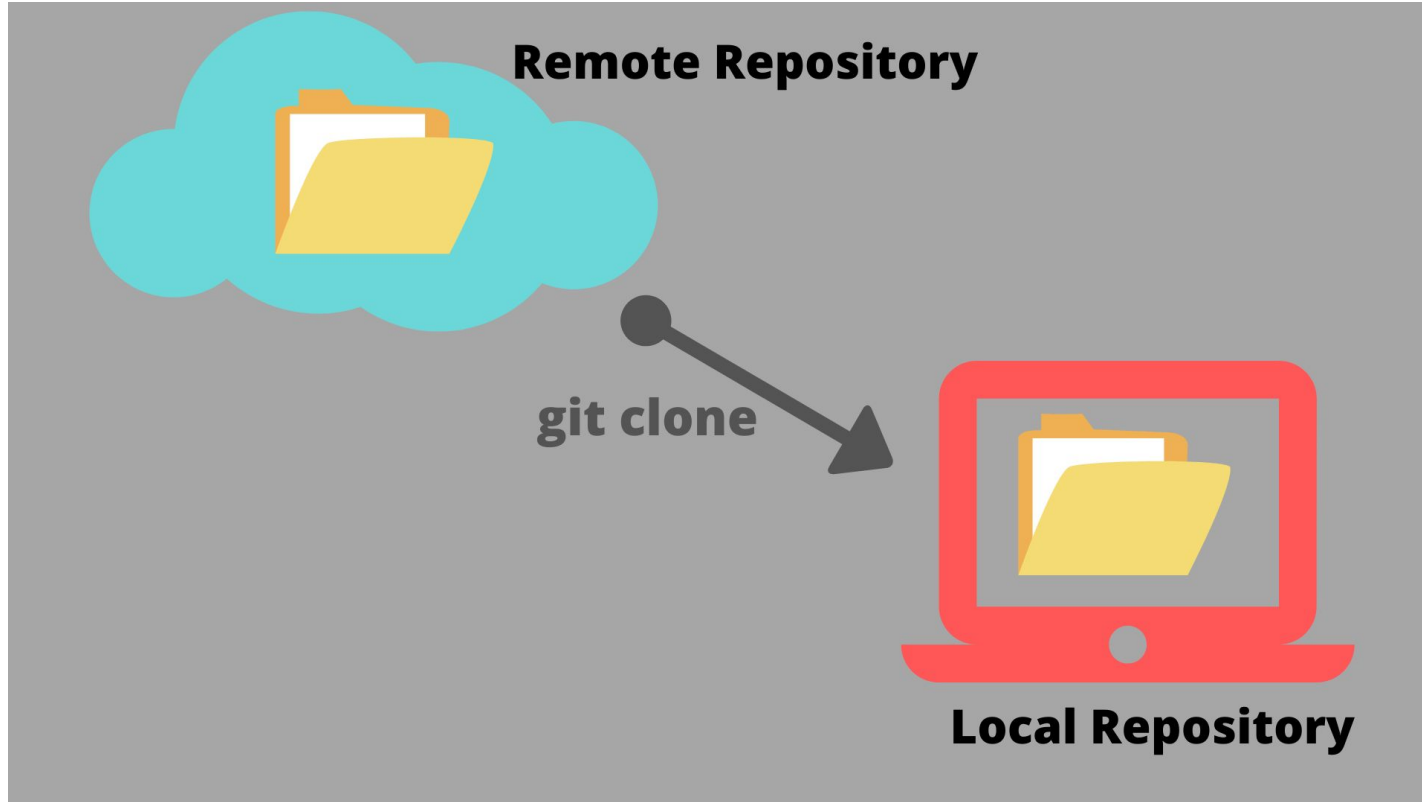
Git



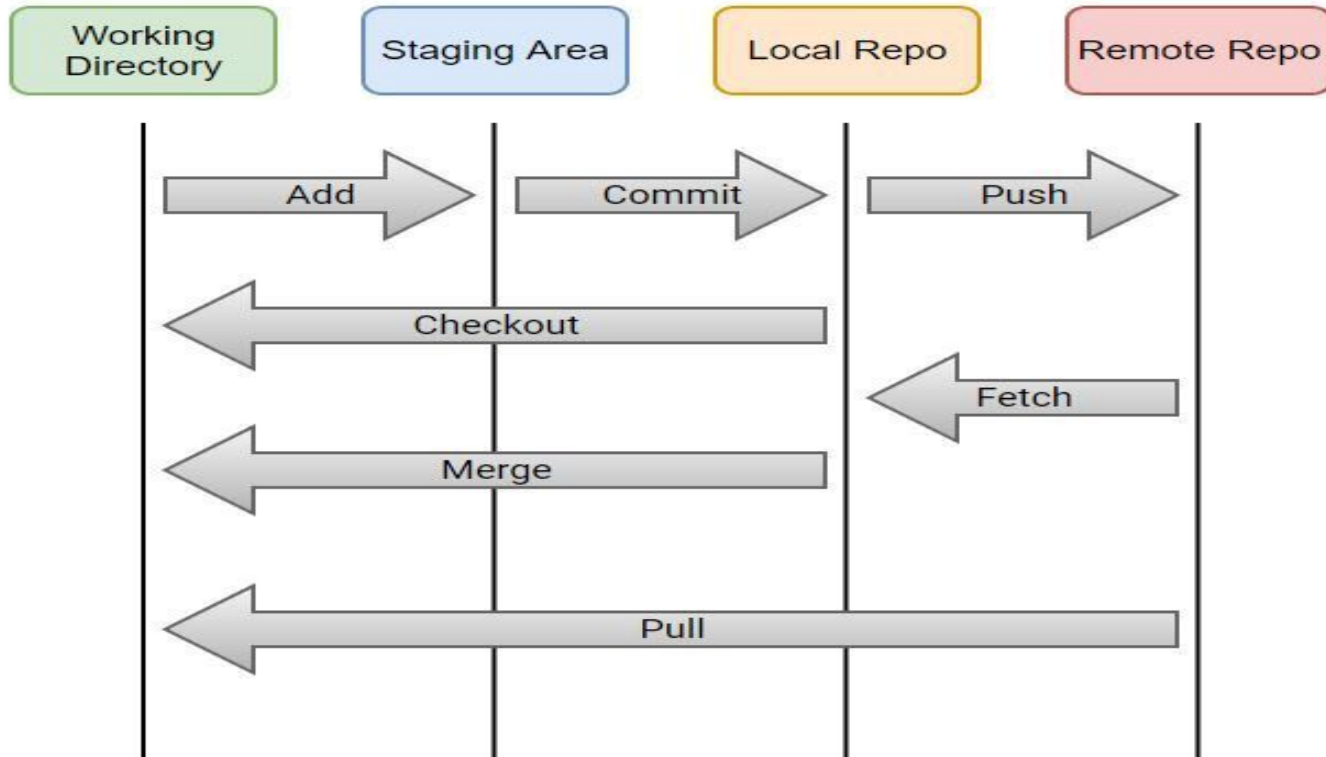
Git



Git



Git



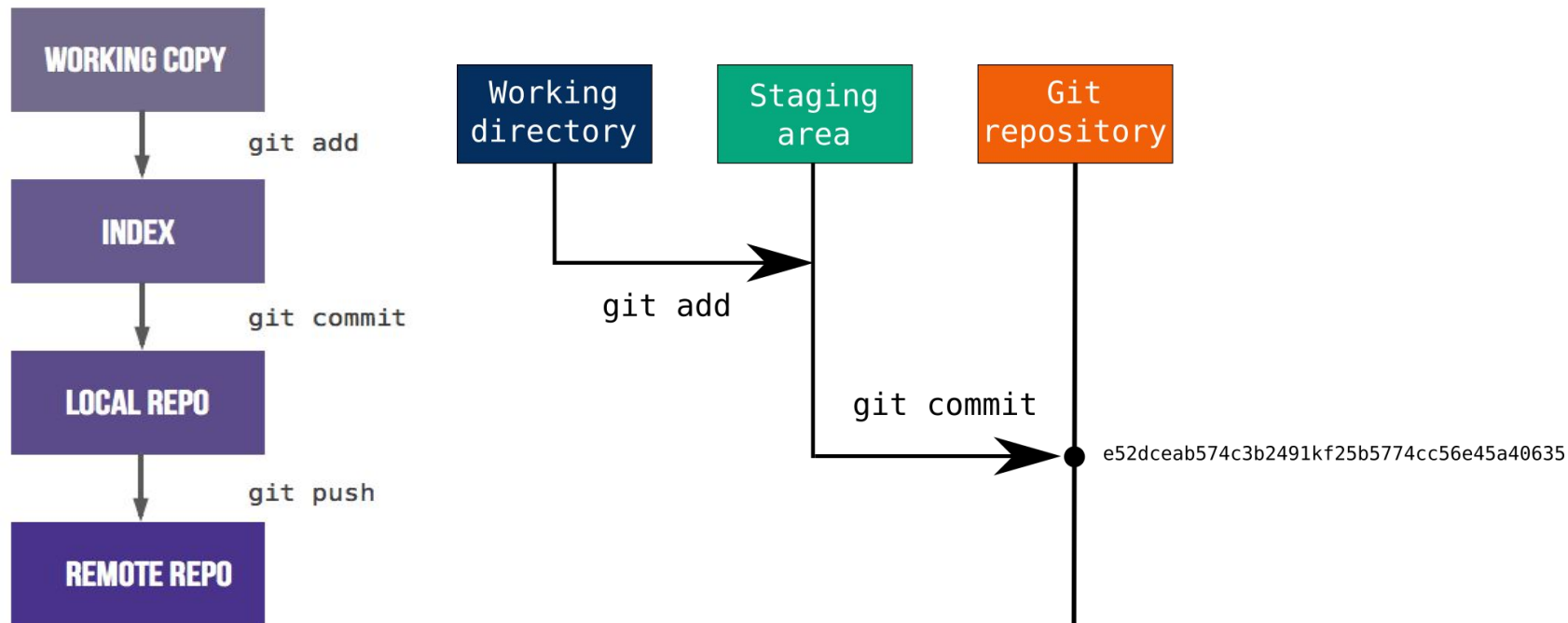
Git - avancerad

git-branch - List, create, or delete branches

git-checkout - Switch branches or restore working tree files

git-merge - Join two or more development histories together

Git



<https://examples.javacodegeeks.com/software-development/git/git-commands-tutorial/>



- Gitlab är en hemsida som tillhandahåller repo-hanterare.
- Vi använder `gitlab.liu.se` i kursen.
- **Det är inte tillåtet att använda Github.**
- Alla skapade repo:n i kursen ska vara privata.



Demo



Tack för att ni lyssnade!