



Java

TDDC77

Objektorienterad Programmering

Föreläsning 0

Sahand Sadjadee
IDA, Linköpings Universitet
Hösttermin, 2020

Outline

- Kursinfo
- Introduktion till Programmering
- Java Development Kit(JDK)
- Linux
- Git

Kursinfo● 1

Hemsida

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

Organisation

- 14 föreläsningar under HT1 och HT2.
- 34 laborationstillfällen, 5+1 obligatoriska labbuppgifter:

Examination

- Kursmoment:
 - 6 laborationsuppgifter (HT1 och HT2). 3hp. LAB (U, G)
 - 1 programmeringsprojekt (HT2). 2hp. UPG(U,3,4,5)
 - 1 datorbaserad tenta. 1hp. DAT (U,3,4,5)
 - Basgruppsarbete kring 9 vinjetter. 2hp. BAS (U,G)
- Ett godkänt på alla moment krävs för att få slutbetyget.
- Slutbetyget bestäms av UPG och DAT.
- Examinationen sker alltid individuellt.
- **Arbetet på LAB och UPG sker i grupper av två personer.**

Examination-regler

- Ni får **INTE** lämna in kopierade, även modifierade, lösningar.
- Varje student måste kunna visa att hen förstått **ALLA** detaljer i lösningen. Varje student ska frågas individuellt under redovisning.
- Varje student ska redovisa till läraren som är ansvarig för gruppen som studenten har registrerat sig i på Webreg. Frågor kan ställas till nån av lärarna som finns på plats.
- 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 är på plats.
- Föreläsaren löser uppgifter på varje föreläsning.
- Håll avstånd till varandra när ni är på plats. Det ska finnas en tom sittplats mellan varandra.
- **Det är starkt rekommenderat att vara påläst i förväg på ämnena som tas upp på varje föreläsning.** 🙌



Labbtillfällen



- Sker på distans via Teams. 😊
- Adam, Anton och Max handleder er under hela vägen.
- 4 labbuppgifter i första perioden, 2 labbuppgifter i andra perioden.
- Labbarna ska göras i ordning.
- Vi har även bokat labbsalar för varje labbtillfälle som kan användas. Handledarna finns dock inte på plats.
- Det är inte tillåtet att sitta bredvid varandra vid samma dator i labb-salarna. Varje student ska använda en separat dator.
- **Det är ett krav att jobba i par.** 😡

Teams

- Teams är kommunikationsverktyg från Microsoft.
- Teams ska användas oavsett du labbar i våra labbsalar eller hemifrån.
- Teams ska installeras på egen dator om du tänker på att labba hemifrån.
- Teams är installerad på datorerna i våra labbsalar. Du ska dock ta med egna hörlurar för att kunna prata med din handledare via Teams.
- Teams ska bara användas när det finns ett pågående labbtillfälle.

Webreg

- Webreg rapportering av LAB, UPG:
 - LAB: i grupper av två personer innan 2020-09-02,
<https://www.ida.liu.se/webreg3/TDDC77-2020-1/LAB>
 - UPG: i grupper av två personer innan 2020-09-02,
<https://www.ida.liu.se/webreg3/TDDC77-2020-1/UPG>

Frågor

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

Frågor?

Programmering { }

Datorer överallt



Hårdvara VS Mjukvara

- **Hårdvara** är 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. I andra ord, styrs datorer av mjukvara. Till exempel, Adobe Photoshop, Linux, Chrome och så vidare.
- Mjukvara existerar **INTE** fysiskt.

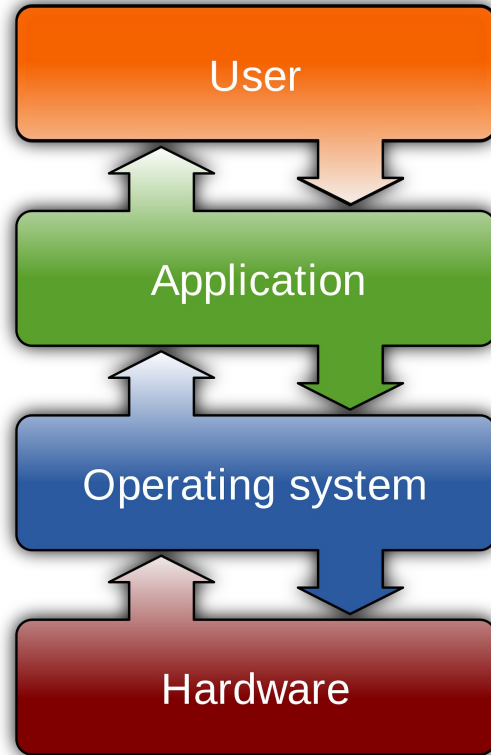
Operativsystem

- 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.

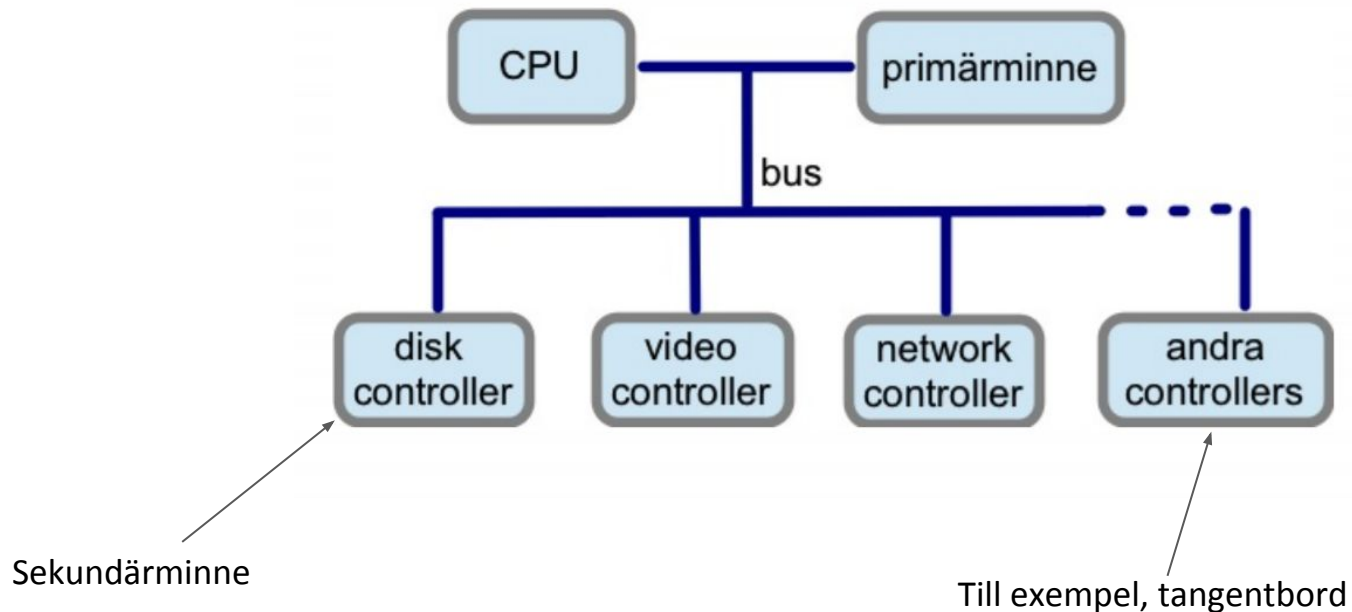
Skillnaden mellan att köra och att exekvera?



Operativsystem



En vanlig dator

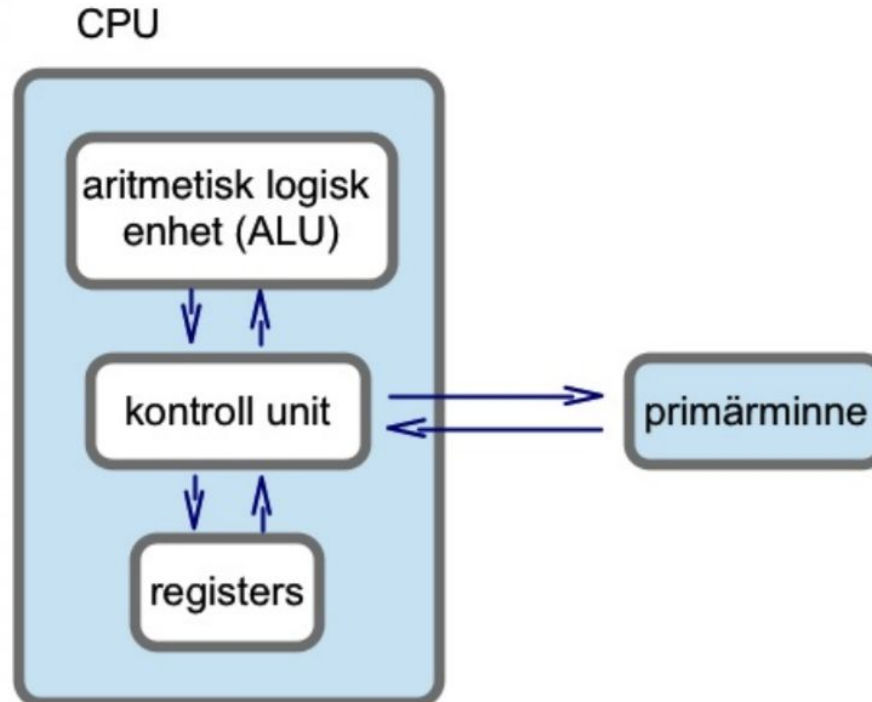


Minnet... 8 bits ger 1 byte

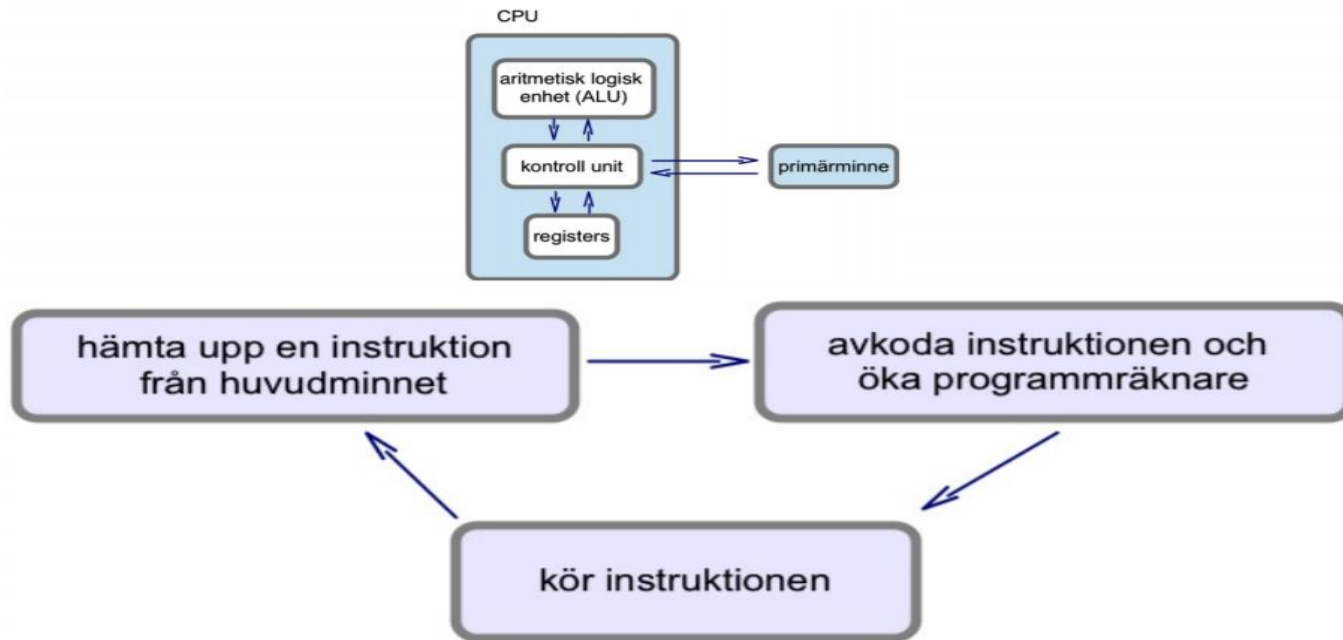


byte	B	$2^0 = 1$	10^0	B	byte
Kibibyte	KiB	$2^{10} = 1024$	10^3	KB	Kilobyte
Mebibyte	MiB	$2^{20} = 1,048,576$	10^6	MB	Megabyte
Gibibyte	GiB	$2^{30} = 1,073,741,824$	10^9	GB	Gigabyte
Tebibyte	TiB	$2^{40} = 1,099,511,627,776$	10^{12}	TB	Terabyte
Pebibyte	PiB	$2^{50} = 1,125,899,906,842,624$	10^{15}	PB	Petabyte

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 skillnad när det kommer till Java? Vad är det?
- Det behövs inte kunna processorns detaljer för att kunna skriva ett program genom att använda högnivåspråk.

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

- Webb applikationer
- Mobil applikationer
- **Desktop applikationer(Terminal-baserade och GUI-baserade program)**
- Bibliotek
- Embedded program
- ...

- 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

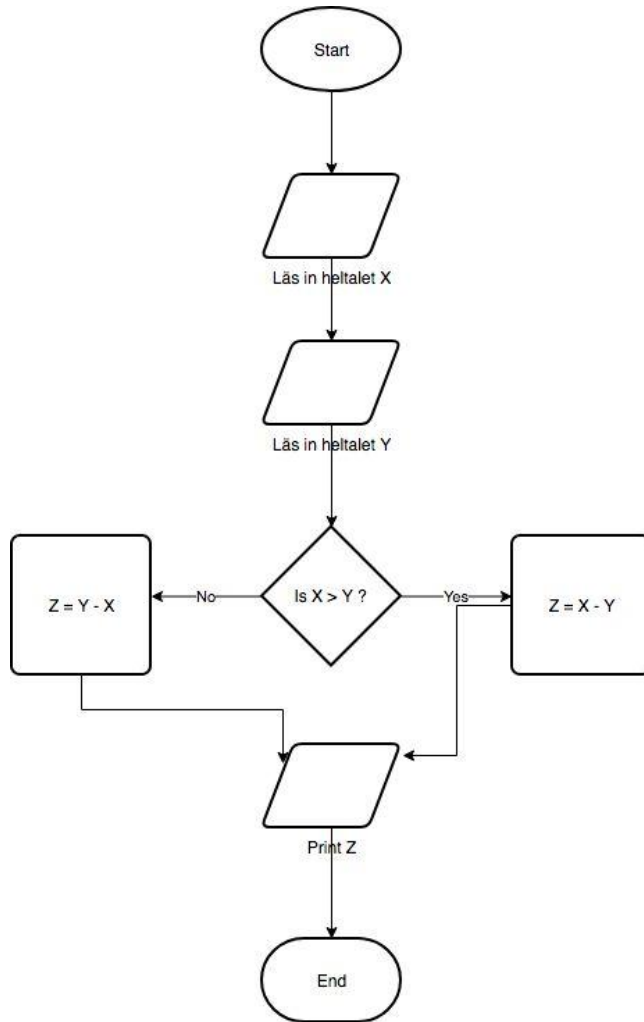
Indicates a list of information with a standard structure that allows for searching and sorting.



Internal storage

Indicates an internal storage device.

Flowchart

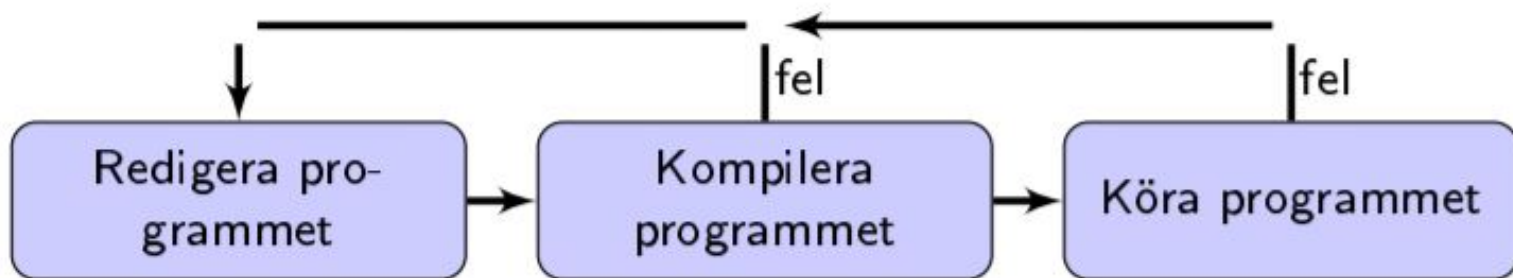




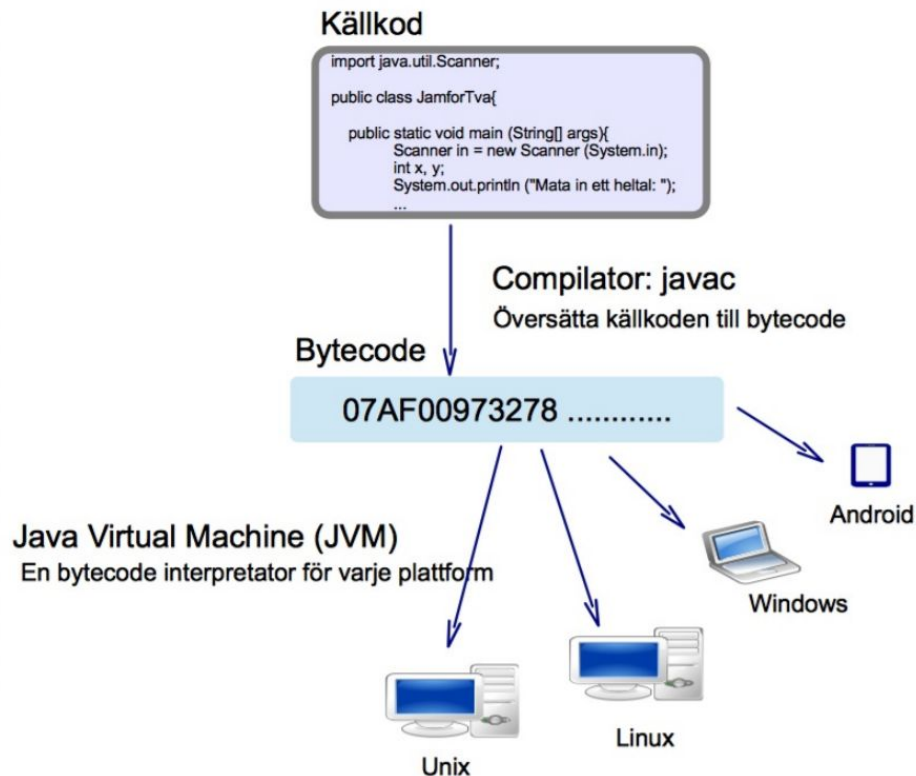
Java™ Development Kit(JDK)

Kompilera och köra programmet

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



Kompilatorer och JVMer



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	Release date	End of Free Public Updates ^{[6][7]}	Extended Support Until
JDK Beta	1995	?	?
JDK 1.0	January 1996	?	?
JDK 1.1	February 1997	?	?
J2SE 1.2	December 1998	?	?
J2SE 1.3	May 2000	?	?
J2SE 1.4	February 2002	October 2008	February 2013
J2SE 5.0	September 2004	November 2009	April 2015
Java SE 6	December 2006	April 2013	December 2018
Java SE 7	July 2011	April 2015	July 2022
Java SE 8 (LTS)	March 2014	January 2019 for Oracle (commercial) December 2020 for Oracle (personal use) At least September 2023 for AdoptOpenJDK	March 2025
Java SE 9	September 2017	March 2018 for OpenJDK	N/A
Java SE 10	March 2018	September 2018 for OpenJDK	N/A
Java SE 11 (LTS)	September 2018	At least September 2022 for AdoptOpenJDK	September 2026
Java SE 12	March 2019	September 2019 for OpenJDK	N/A
Java SE 13 (Early-Access Builds 31) ^[8]	25 July 2019	TBA	TBA
Java SE 14 (Early-Access Builds 7) ^[9]	24 July 2019	TBA	TBA
Legend: Old version Older version, still supported Latest version Latest preview version Future release			

JDK

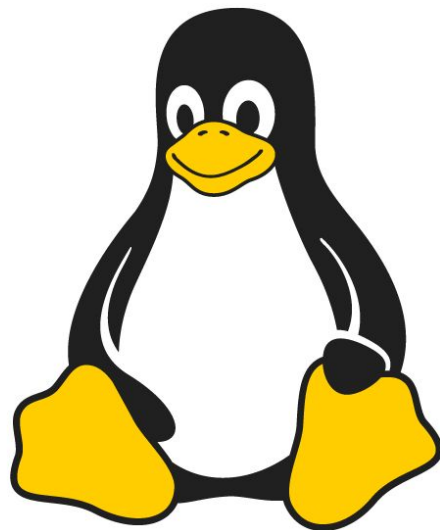
<https://docs.oracle.com/en/java/javase/11/docs/api/index.html>

- 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.

Linux



Linux

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

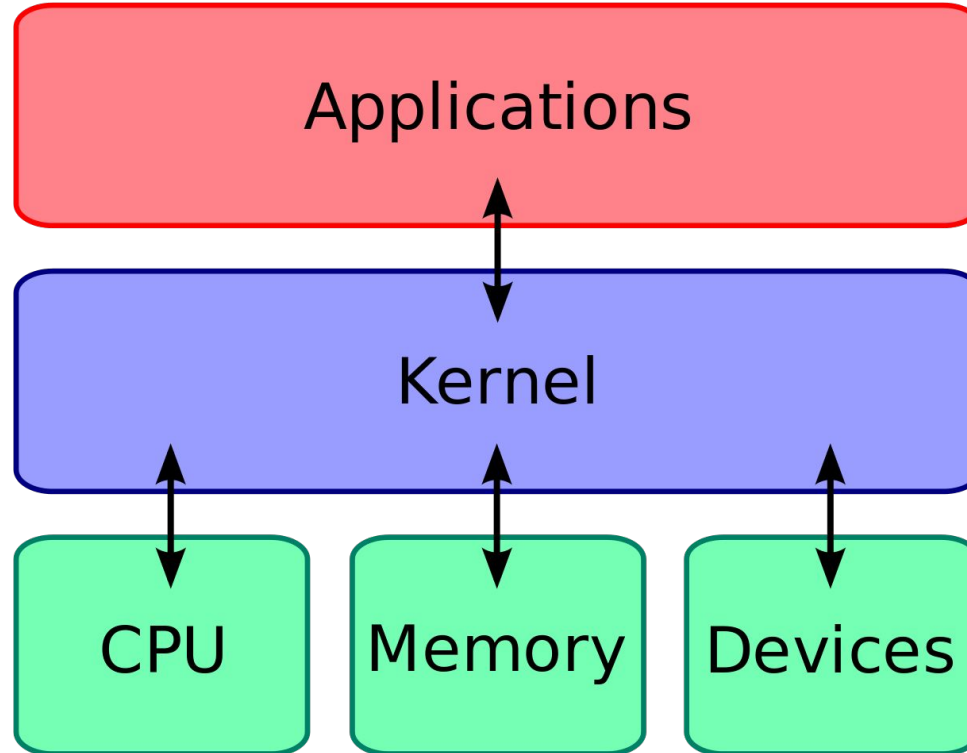
Linux-distributioner



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.

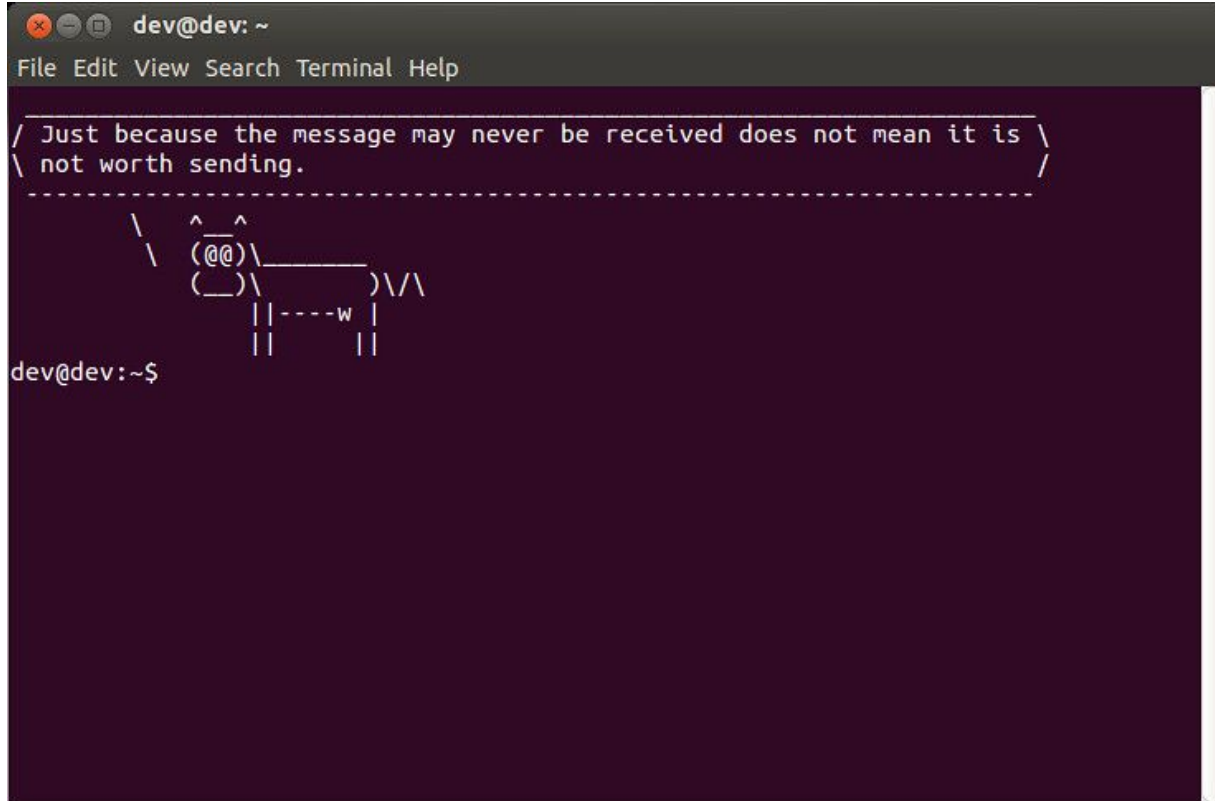
Linux-kernel



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 message enclosed in a box defined by slashes and a dashed line. The message reads: "Just because the message may never be received does not mean it is not worth sending." Below the message is a diagram consisting of a horizontal line with a dashed section in the middle, and a vertical line on the right side. The diagram is drawn with backslashes and underscores. The prompt "dev@dev:~\$" is visible at the bottom left of the terminal window.

Linux-terminalen

- Vissa kommandon är bara ord som förstås av terminalen. Sådana kommandon kallas för “interna/imbyggda kommandon” eller på engelska “internal/built-in commands”. Till exempel, `pwd`, `cd`, `exec`, `logout` osv
- De flesta kommandona är program/applikationer som ligger på hårddisken 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.

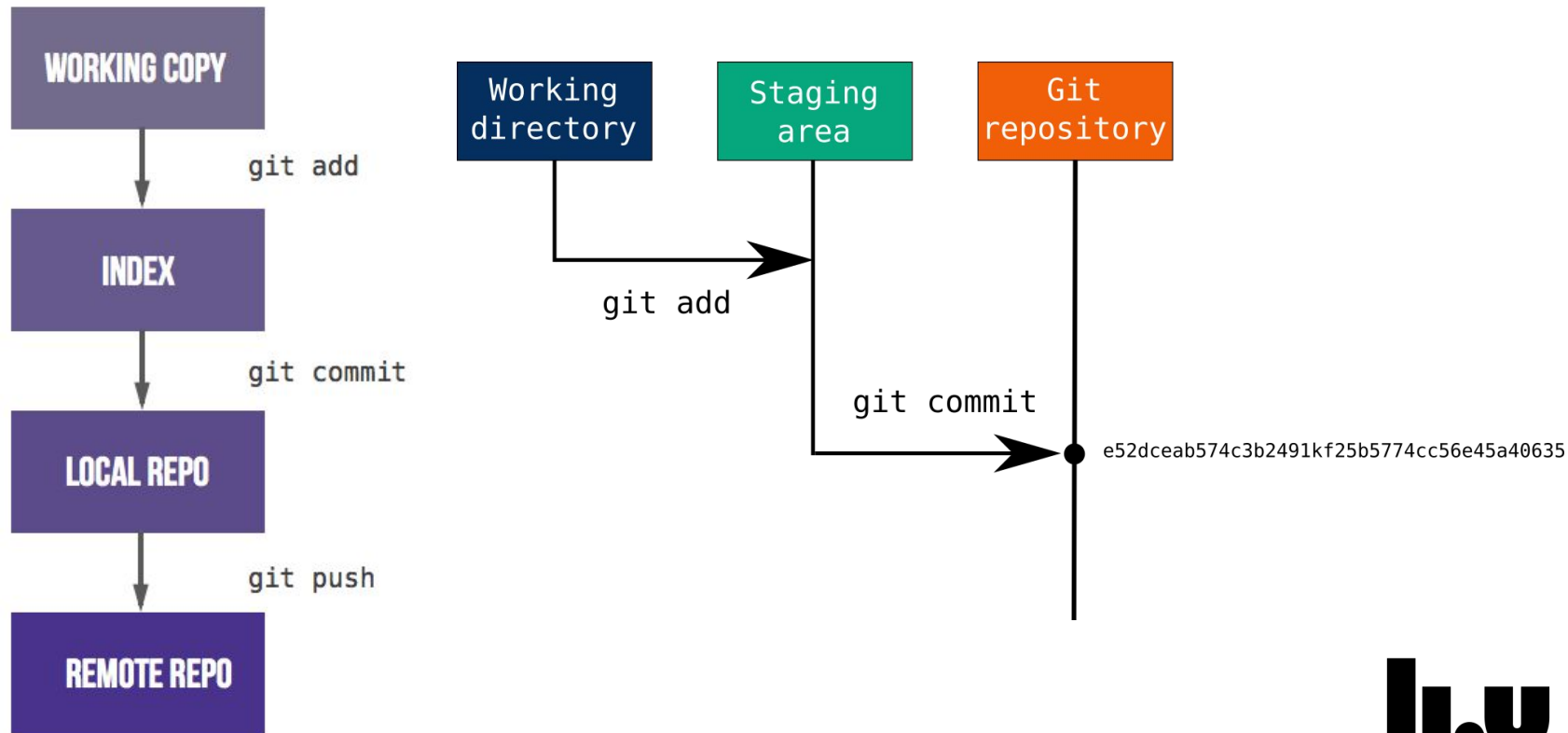


Git

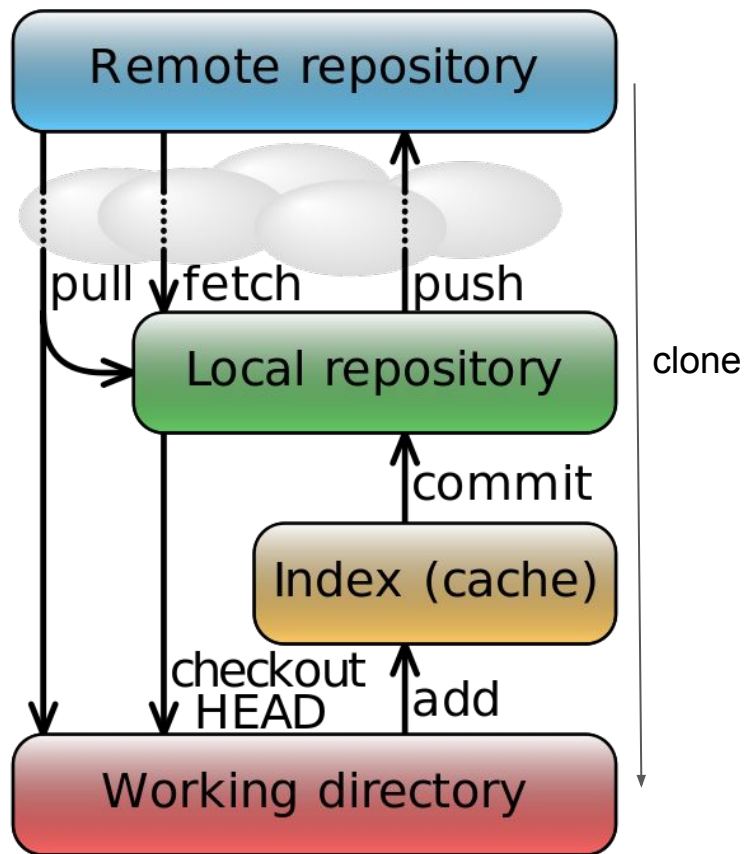
- Git är ett versionshateringsprogram gjort av Linus Torvald 
 - “Versionshantering innebär att tidigare versioner av dokument, källkodsfiler, program eller webbsidor kan återskapas, och ändringar gjorda i dessa tidigare versioner kan spåras.”
 - Med hjälp av Git kan man dela kod med andra.
 - Med hjälp av Git kan man ha en backup på sin kod.
 - Med hjälp av Git kan flera utvecklare jobba på samma projekt.
- Git är ett program som ska först laddas ner och installeras:

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

Git



Git



Git

Local Version Control Systems vs Centralized Version Control Systems vs Distributed Version Control Systems

<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>



GitLab

- Gitlab är en hemsida som tillhandahåller repo-hanterare.
- Vi använder `gitlab.liu.se` i kursen.
- **Det är inte tillåtet att använda Github.**



Tack för att ni lyssnade!