

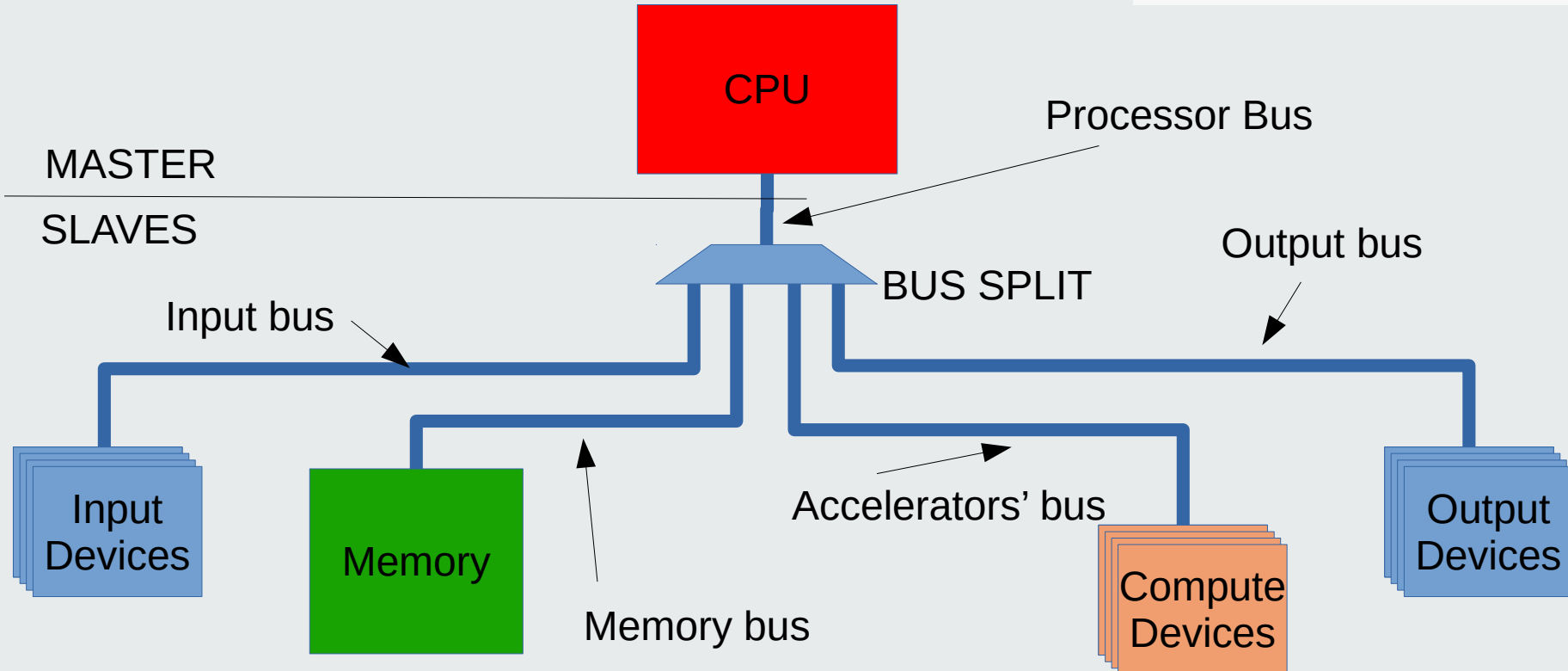
# Computer Electronics

## Lecture 2: Introduction to Iob-SoC, Git, Make



John von Neumann

# Recap of last lesson

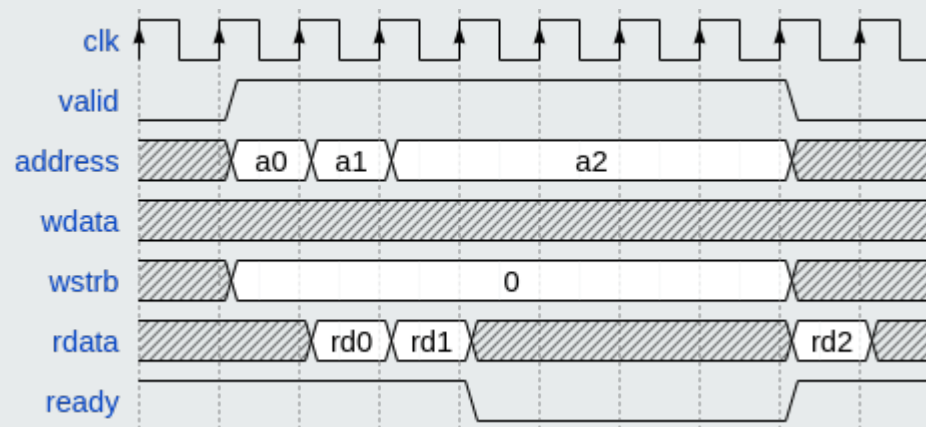


# The Processor Bus

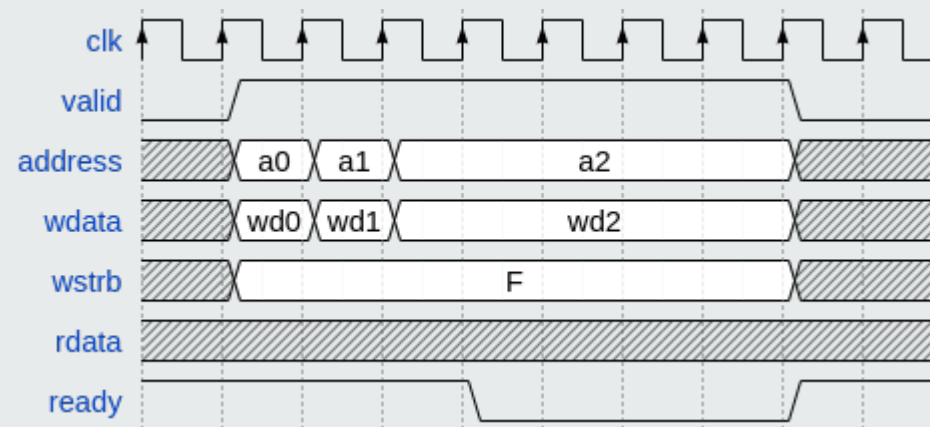
- Processor buses can be complex, for example ARM's AXI
- In this course a minimalistic yet effective 6-signal bus called the Native Bus will be used:

|          | Signal Name | Direction wrt CPU | Width in bits (wires) | Description                              |
|----------|-------------|-------------------|-----------------------|--|
| REQUEST  | Valid       | Output            | 1                     | CPU wants to access data or instructions |
|          | Address     | Output            | 32                    | Address of the data                      |
|          | Wdata       | Output            | 32                    | Data to write                            |
|          | Wstrb       | Output            | 4                     | Byte write                               |
| RESPONSE | Rdata       | Input             | 32                    | Data read                                |
|          | Ready       | Input             | 1                     | Slave is ready                           |

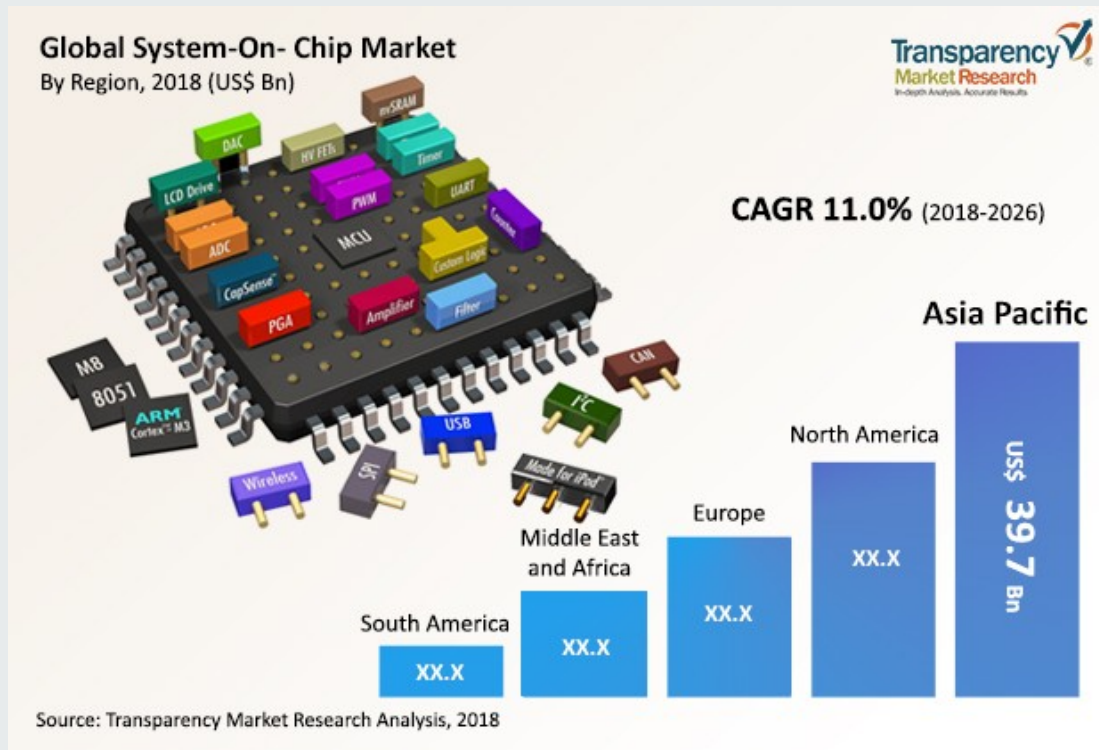
# Master reading from slave



# Master writing to slave



# System on Chip & IP Cores

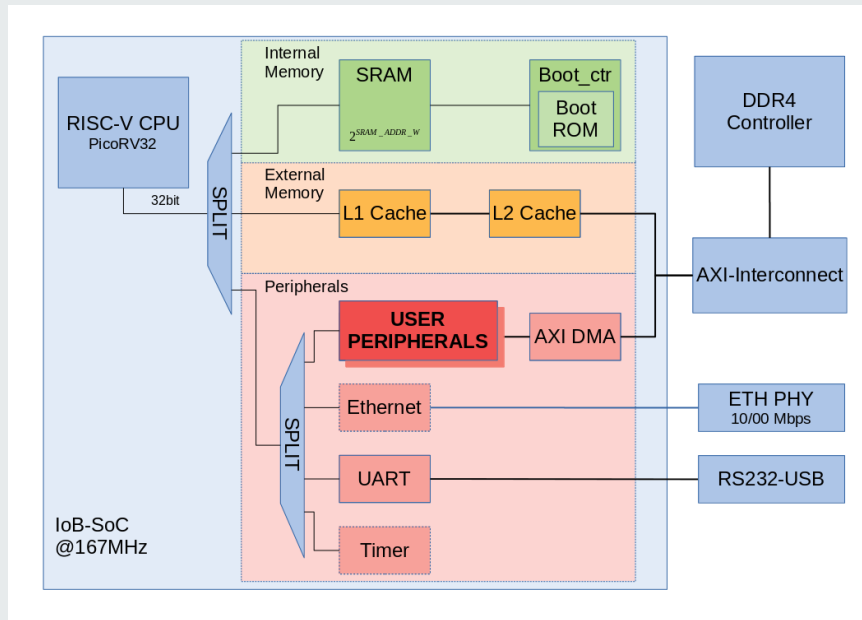


- **IP Core** providers design IP cores: ADC, MCU, PWM, etc
- **Fabless** Integrated Circuit providers integrate IP cores and make the **SoC** (AMD, Xilinx, etc)
- **Foundries** fabricate SoCs and other chips

# System on Chip

- *The SoC is an IP core assembly*
- The whole system sits on a single chip
- Less components on Printed Circuit Boards
- Less weight and volume
- More performance
- Lower power consumption

# IOB-SoC block diagram



- Developed by IP core provider IObundle Lda with the help of master students students (Roque, Silva, Ratinho, Mário, ..., Internships/Dissertations)
- Simple SoC comprising
  - The PicoRV32 RISC-V CPU (open-source)
  - Boot ROM (has small program to load user programs)
  - SRAM (on-chip memory for running small programs or storing fast access data)
  - Instruction and Data L1 Caches + shared L2 cache
  - UART interface for communication
  - Other optional modules: Ethernet, DMA
  - Ready to attach user modules



# Getting hands dirty on IOb-SoC

- Computer systems are inherently complex: use suitable tools!



# Simple IOb-SoC assignment

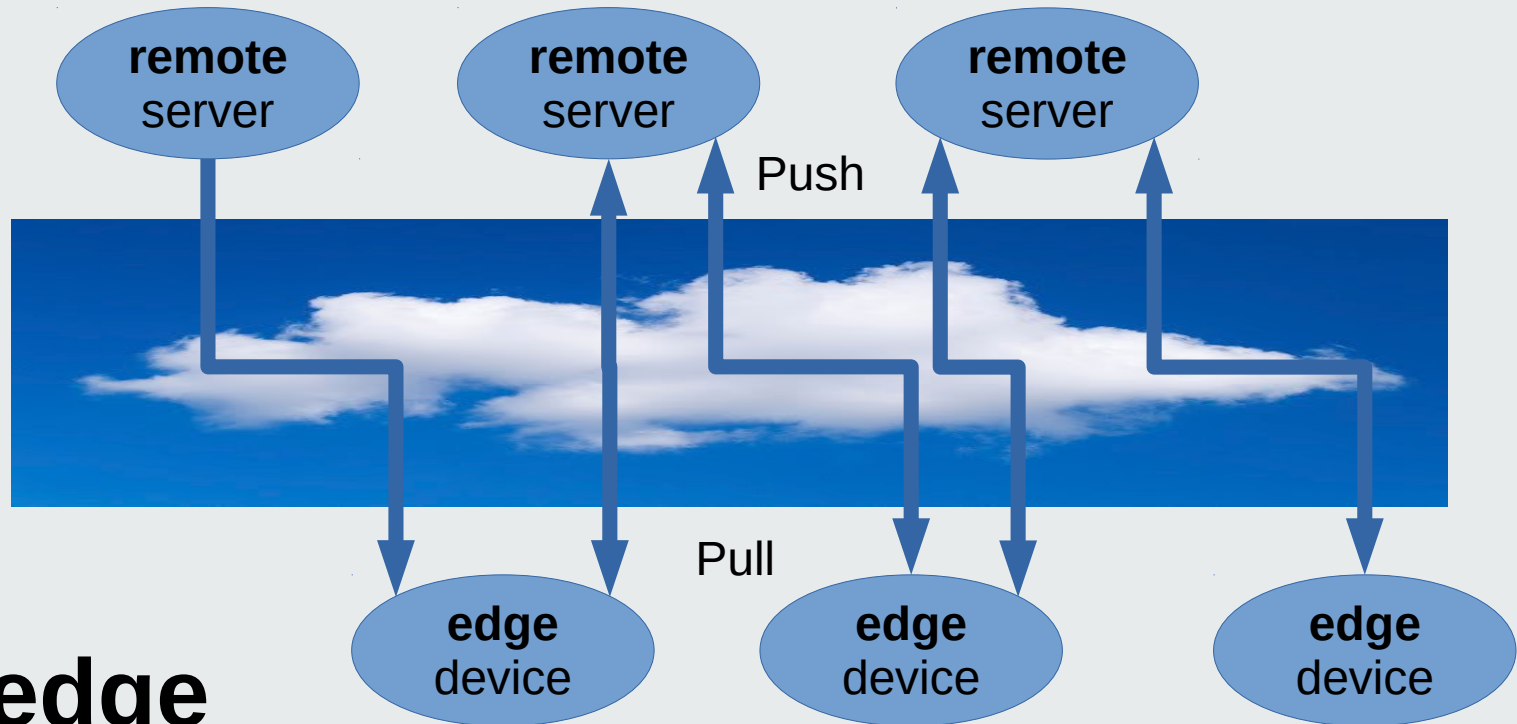
- Hosted by Github
- Read the README file
- Clone the repository
- Move into its directory
- Simulate the design and visualize simulation waveforms
- Make the document



# Git in a few slides

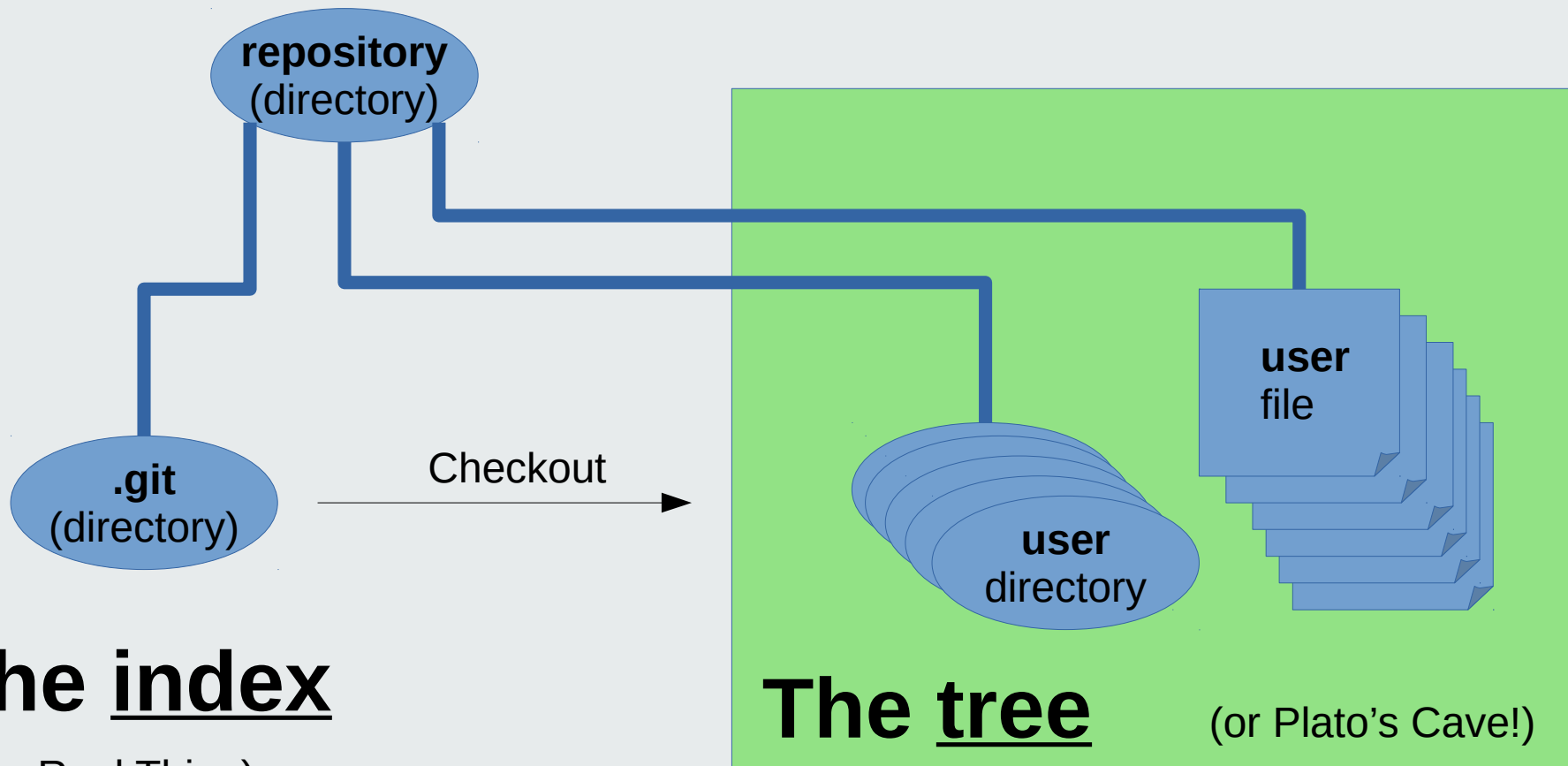
- Developed by Linus Torvalds (also Linux creator)
- Managing your project files efficiently
- Reduce storage and backup needs
- Manage different versions and development branches in large teams
- Access the project files anywhere
- Successor of CVS, SVN but better

# The cloud



# The edge

# Meanwhile and the Edge



## **The index**

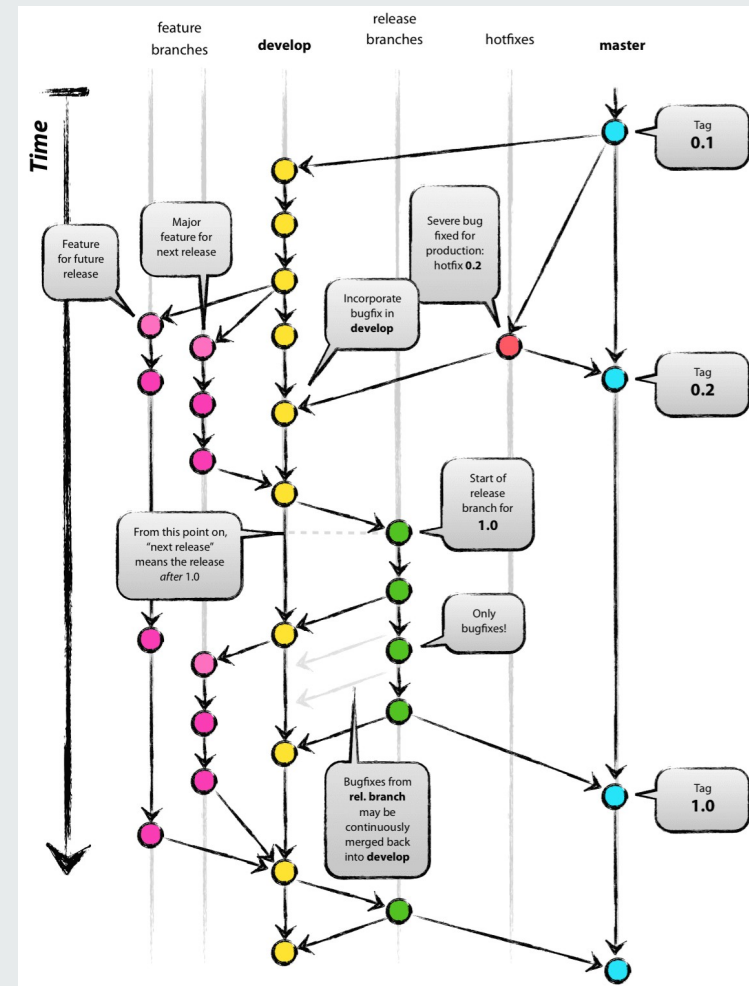
(or Real Thing)

# Common git commands

- **git config** –global user.name “your name”
- **git config** –global user.email “your@fantastic.email”
- **git init**: turns directory into git repository
- **git status**: shows differences to index
- **git add**: stages the files to be committed in index
- **git commit**: creates a new development node (commit or hash)
- **git push**: sends your commits to a remote
- **git branch**: creates a development branch

# Development branches

- Develop each feature in a branch
- Test / Fix
- Merge with master branch
- Retest / Fix
- Release!



# Common git commands

- **git log**: shows commit history
- **git fetch**: downloads branches from a remote
- **git pull**: same as **git fetch + git merge**
- **git show**: shows contents in any commit
- **git remote**: shows remote servers in use
- After you get up: eat breakfast then do **git pull** (update your repo)
- Before you go to bed: do **git push** (save your work) and brush your teeth





# The Make utility program

- Make is a program that automates the process of building programs (compiling)
- Created by Steward Feldman in 1976
  - Also wrote the first Fortran 77 compiler
  - Part of the Bell Labs team who created the Unix operating system, the precursor of Linux for masses
- Make is still one of the best build automation tools to date!

# The Make utility program

- Make programs are called Makefiles
- Makefile is based on rules. Example rule:

