C++ in the developing world, why it matters

Mathew Benson

About Me

- I like C++!
- I am a Graduate Computer Scientist(since 2007)
- I have been working with and researching on computers and programming and how to practically apply it well for several years, still learning and enjoying the journey.
- I am passionate about energy efficient computing and making the most out of older hardware, researching different platforms, tools and how to bridge different worlds and peoples together.

Contacts

- Twitter : @bensonorina
- mastodon : @mathewbenson.hachyderm.io
- email : benson.orina@live.com
- Github : github.com/mathewbensoncode
- Discord : #CppAfrica -> mathewbensonke

Agenda

- The talk is an exploration of the importance of C++ as a language for writing efficient code for older hardware which is easier to access in the developing world.
- It will also look into the C++ language as a platform for education about software due to the large amount of "real world" software that has already been written in the language.
- It will also touch on tooling and resources for programming on resource constrained systems.

Definitions

C++

- What is C++?

- Probably a silly thing to be defining what C++ is at a C++ developer Conference
- however if you think about it... What is C++ really?
- It is difficult to answer this without talking about C.
- As you may be aware, C++ is an evolution of the C programming language. This evolution is analogous to a journey.
- C was necessary as a step up from assembly. It was born of a need to better express programming intent.
- C++ was a necessary step up from C.

Developing World

- The Wikipedia definition of "Developing World" is from the perspective of income levels and access to money.
- I will take a slightly different trajectory in this talk.
- Let's First Discuss what Developed Means in Terms of Computers

Developed

- To have a processors with enough cores that will not *FEEL SLOW* in the expected tasks
- To have Internet Access That doesn't *FEEL SLOW*
- To Have uninterrupted Power(Electricity)
- To Have uninterrupted connectivity. (Its available when the User Needs It)
- To Have adequate KNOWLEDGE, to make the most of the computing resources available
- To have All of the Above at a *COMFORTABLE COST*

Developing

- The process of "Growing Closer To" / "Moving Towards" an envisioned, state that is ideal and comfortable -> "DEVELOPED"
- So in Terms of Computers, I mean an experience that is lacking in ANY of the above requirements.

World

- My emphasis is on a world, not a country.
- This means any person, group, region.
- It wouldn't be fair to assume that everyone from a "Poor" country is "Poor" and everyone from a "Rich" Country is "Rich"

Older Hardware

- As *TIME* has progressed, Techology has improved to make SMALLER electrical components.
- Meaning that it takes LESS POWER To do the SAME tasks as before.

- It may also mean, MORE Tasks with the SAME POWER Levels.
- This Technological Improvement does come at quite a significant cost, Meaning that Newer Computer Hardware Costs More, out of Reach for many people.
- However, This Technological Improvement, does push down the cost for the “Previous Generations” of Hardware,

Efficient Code for Budget/Older Hardware

- With a large number of people with less to spend on computer hardware, energy and connectivity, access to computing must come at a lower cost.

Access To Computer Hardware At Lower Cost

- One way is that computers come in a differant form, namely budget mobile phones.
- The second way is that older, “Obsolete” computers from the “Developed” World find their way to the “Developing” World.
- To hit the lower price points on “Newer Budget” Offerings, manufacturers will likely use older technology rebranded as new.

Difference Between The CONNECTED and the UNCONNECTED

- Chromebooks, have been deployed in some countries as a solution to low spec, mass deployment of computing resources.
- This however comes with the precondition of connectivity.
- This solution cannot work in the “Developing” World.

Solving The Riddle With Software

- In the “Developing World”, the systems available to most users is usually underpowered or very power inefficient.
- Therefore Software Perfomance is Crucial.
- It is important to always think about the low end, because this is where the people that need it the most will lie.
- It is also where scale matters, for example, the moment you are dealing with large deployments, e.g. to schools, governments,

- Projects to enable access to Computing will usually fail because of the COST of ADEQUATE computing.

How Does C++ Solve this Problem?

By Design

- It is one of the core fundamentals of the language. The "Zero OverHead Principle"

Compile Time Computing

- The less work that will be done on the resulting client PC, the better. Meaning, the less translation to machine code, the better. (I am aware that there are ways to "precompile" many Interpreted languages, but it is not the natural order of things.)
- Constexpr support. With more done on developer machines, only the necessary run time computations need be done on the "client" machines and this has major potential to unleash power in areas that had not been considered.
- The stack based memory model for computing is better idiom. Being in control of what is happening on the stack is a more natural way to think about computing.

Evolution From C but Still Compatible with it

- The backwards compatibility with C is crucial since a lot of critical sofware at the lower level is actually written in C.
- To allow chances for improvement of code by more developers in a safer way; We can use safer tools to experiment, optimize and interact with our systems.
- It may be the safest way to access system level features of our operating systems in a safer way. Alot of the software and libraries commonly used in the tasks you would want to use are written in it.
- For example libraries like opengl, vulkan, direct3d that provide access to hardware graphics capabilities require the use of C level constructs, but when wrapped in C++ containers, types, with the use of RAII

Learning/Teaching C++

- Now that we have talked a little about the "Why" I would like to touch on the "How"
- I am not a teacher of the C++ language, just someone who happens to have learnt through effort.

Oh, the Humanity, the Diversity

- This talk is really not about statistics, facts, figures and measurements. It was born of my reflections after a long hard journey working with software both as a scientist, developer and as a user.

- So I may not have some hard numbers, but I seek to appeal to our reflection on an individual basis on these issues.
- I believe we all people are conversant with the concept of our diversity.
- This diversity is visible in our different personalities, races, colors, genders, religious views, talents, etc.

Comparison to human language

- C++ is a language, So Let's Compare how we learn our Human Languages.

According to the FAQ Section on Language Acquisition on the Website for the American Linguistic Association

- When Children are born they have a natural ability to pick up language, usually the language they interact with first.
- Children Acquire Language by interaction with the language through talking with adults and other children.
- There are certain stages that are similar in acquiring language.
- This learning is gradual and happens step by step.
- Adults Help to Improve the Learning of Children by Stepping Down A level Closer to Children Through "Baby Talk"

How Does this Map to Learning C++

- We need to bring the human interaction into it.
- Through our Communities, Our Work Environments
- I believe this to be one of the benefits of our C++ Conferences, User Groups, Discord Servers, etc.
- The more we speak about code, interract with our code, read our code the better we get at using our computers

Re-Evaluating "WHY" We Use Computers and Programming

- As with many things in life, many people use(learn) computers and programs due to reasons that often conflict with the intended design of said computers and programs.
- This is different for each individual
- So we all have our different starting points

RoadMap for Learning C++

- There may not be a one-size-fits-all answer to this for everyone as we are all diverse
- You may need to try different teachers, books, resources & techniques before you find the right mental model for your unique mind, talent, etc.
- We are all learning, just that some of us have been learning for longer.

"C" The Story

- One of the greatest, in my opinion, reasons behind my preference for C++ in particular is that it is part of a rich history.
- Like a movie that is part of a series of movies where the story just couldn't fit in one movie length e.g Lord Of the Rings, The Star Wars Story, etc.
- Explore why things are the way they are, many features come about as a result of rich discussions, arguments, testing, trial and error

One Step At A Time

- "The Journey of a thousand miles begins with a single step"
- Keep a Reference Close At Hand, and I mean Really Close At Hand. (http://cppreference.com)

Study and Research the basic concepts and keywords starting with these:

- expression
- statement
- variable
- object
- Declaration
- Definition
- Function

Tooling Tips

For older or Constrained Computing Scenarios

- To be able to boost interactivity with the Language, there must be access to computers, power, books, connectivity, etc.
- It is important to crawl before you walk and walk before you run.
- In this regard, I think the answer may be to look into the past a little and Look into how computers worked in times past.

Don't Start with Integrated Development Environments

- When Starting out it may be cheaper, and more effective to avoid relying on IDEs when starting to learn.
- In my opinon the IDEs like visual studio, clion and the rest are great when you are already in professional environments and with more powerful hardware, and when time is of the essence in professional environments.
- The value proposition of IDEs really shines once you are aware of what it is they really do for you.

The Terminal

- I believe that it is important to first understand where computing is coming from.
- The Terminal is essential to this.
- Knowing about Plain Text Files, compiler commands, linker commands is essential.
- It is also the lowest cost model for teaching programming to more people, in more situations.
- Learn Bash(Posix Shell) or Powershell. These are invaluable

Keep As Much as You can Locally

- Having all you need on your developer machine is better than always having to rely on resources on the internet. Especially in places with expensive/limited connectivity.
- It is possible to download the entire cppreference.com website locally. It is available as a package on arch-linux.
- Get Reference material like the Pdf of the microsoft developer website C++ documentation on PDF.
- Buy Good Books On C++ When you Can Afford To.

Share Compute Resources On Local Networks Where Possible

- You can have a local network version of compiler explorer.
- This is To Reduce Connectivity Costs and Use the Limited Connectivity for the most Crucial Things.

SSH

- This is one of the benefits of Terminal Based Computing.
- Try To Learn About Vim/Neovim or Emacs
- Or share a single computer with multiple users who may use basic android phones with terminals via ssh-clients. This works with windows PCs.
- I use this personally as my personal machine is Linux based, but always ssh into my wife's laptop to try things out on windows using neovim on both.

Conclusion

- For our Language C++ to thrive and be of greater benefit, Every One of us Must be as welcoming as possible to new users.
- We need the perfomance benefits of C++ applied to where it matters most i.e. on Older, Cheaper Hardware to make computing more accessible to more people in more places.

- Older Tooling and Software such as the Terminal that would be considered by some to be obsolete can, when applied appropriately b e the key to creating opportunities for people who may not have had a chance otherwise.
- The Personal Computer Can Do so much more.
- Lets Upgrade our Users before We upgrade our Hardware.