

1. Október 8.:

– Általános bevezető, C++11 áttekintés.
– Ízelítő és kedvcsináló a C++11 és újabb szabványok kényelmi-biztonsági feature-jeihez:

- initializer_list, Uniform initialization (előnyök, hátrányok)
- Range-based for loop
- Type inference alapok
- Lambdák bevezető
- Move semantics bevezető

2. Október 22.: RAII idióma részletesen, és a smart pointerek részletezése.

- A move semantics jelentősége a unique_ptr, shared_ptr és weak_ptr esetében.
- A unique_ptr és a referenciaszámlálás egy lehetséges (egyszerűsített) implementációja.

3. November 5.: A kód újrafelhasználásának globális szintje, avagy a library-k jelentősége és használata.

- Egy egyszerű példa library használatára (és ismerkedés a GCC vagy clang command-line felületének releváns részeivel).
- A C és C++ fordítási folyamat. Header file-ok, object file-ok, statikus és dinamikus linkelés.
- Kód dinamikus betöltése a dlopen() API-val.
- Library-k készítése. API-tervezési szempontok. (milyen a jó API? pl. naming conventions, típusok helyes megválasztása, stb.)

4. November 19.: "Csoportmunka" a programozásban: maintenance és eszközei. FOSS-kollaborációs elvek.

- De facto build rendszerek Unix-szerű OS-eken. Az autoconf/automake konfigurációs rendszerek, illetve a "make" rövid áttekintése, egy példaprojekten keresztül.
- Verziókezelés és jelentősége. Néhány népszerű verziókezelő szoftver megemlítése; a Git használatának alapjai.

5. December 3.: "Command line culture" – az általános programozói műveltséghez tartozó néhány, leggyakrabban használt parancs áttekintése, és _jó minőségű_ learning resource-ok bemutatása (merthogy rossz minőségűből sok van, és könnyű fals információkhoz jutni... ilyen rossz példa: cplusplus.com a cppreference.com helyett... meg tutorialspoint, ahol a JavaScript tutorial ikonja egy Java-bögre, tehát láthatólag fogalmuk sincs a különbségről, stb...)

- konkrét példák:
 - szövegfeldolgozás. AWK, sed, grep – ezek kapcsán reguláris kifejezésekbe is kicsit mélyebben belemerethetünk: mire jók, mire NEM jók, ... sort-uniq-cut
 - bc, a parancssoros számológép
 - tipikus/gyakran használt shell-ek, pl. BASH bemutatása. Környezeti változók kezelése, shell scripting trükkök. (subshell expansion, avagy hogy lopjuk el feldolgozásra egy parancs kimenetét, és hasonló apróságok.)

6. December 10.: Toolchain-ek (fordítóprogram, assembler, linker, debugger, profiler) működési elve (magasszintű áttekintés), compiler stages (ebben alapoznánk az előzőek során elmondott "C és C++ fordítási folyamat"-ra).

- Néhány gyakorlati szempontból fontos érdekesség tárgyalása, pl. hibaüzenetek értelmezése céljából (miért is "undefined symbol" az az undefined symbol hiba, ...).
- Egy command line debugger (pl. GDB vagy LLDB) és profiler (gprof vagy Google Performance Tools) alapvető használatának bemutatása.

7. December 17.:

- Algoritmusok vizsgálata hierarchikus memóriájú rendszerekben (cache és memóriabejárési minták kapcsolata, rekordok tömbje vs. tömbök rekordja, folytonos és csúcs alapú adatszerkezetek kapcsolata a cache-sel, false sharing, kevés mozgatus(inkább helyben módosítás)

- adatszerkezetek az algoritmusok ellen, utasítás-gyorsítótár, problémák párhuzamos és elosztott rendszerekben).

- Aszimptotikus komplexitás vs. konstans faktorok jelentősége – mikor mit optimalizáljunk? Egy példa: az Apple open-addressing hash table-je, maximális probe sequence length tárolásával.