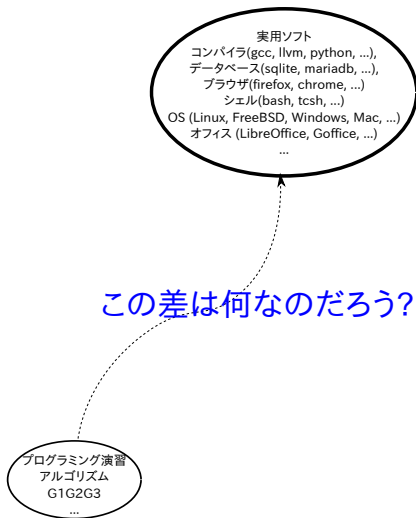


結びに変えて

田浦
遠藤，鳴海

- ▶ あっという間に終わりが近づいていきます...



この演習で得てほしかったもの

- ▶ 中規模以上のソフトで常識的なこと (分割コンパイル, ライブラリ, 各種ビルドツール, configure, make, ...)

この演習で得てほしかったもの

- ▶ 中規模以上のソフトで常識的なこと (分割コンパイル, ライブラリ, 各種ビルドツール, configure, make, ...)
- ▶ ソフトの中身を自信を持って追跡する方法 (gdb)

この演習で得てほしかったもの

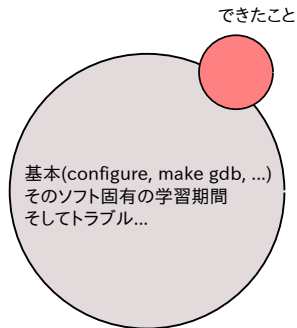
- ▶ 中規模以上のソフトで常識的なこと (分割コンパイル, ライブラリ, 各種ビルドツール, configure, make, ...)
- ▶ ソフトの中身を自信を持って追跡する方法 (gdb)
- ▶ トラブルのあれこれの対処方法 (原理をその場で学習. Google 先生)

この演習で得てほしかったもの

- ▶ 中規模以上のソフトで常識的なこと (分割コンパイル, ライブラリ, 各種ビルドツール, configure, make, ...)
- ▶ ソフトの中身を自信を持って追跡する方法 (gdb)
- ▶ トラブルのあれこれの対処方法 (原理をその場で学習. Google 先生)
- ▶ 最低限の知識を身につけたあとは, 「やってみればなんとかなる」という精神・負けない根性

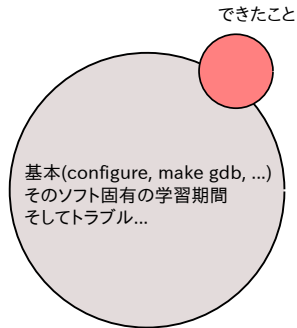
...だが持ったかも知れない感想

- ▶ 少しの拡張を施すのに苦勞が多すぎ?



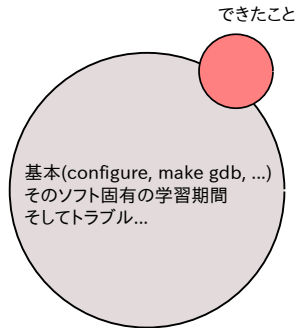
...だが持ったかも知れない感想

- ▶ 少しの拡張を施すのに苦勞が多すぎ?
- ▶ 望むらくは, 再利用可能な初期投資



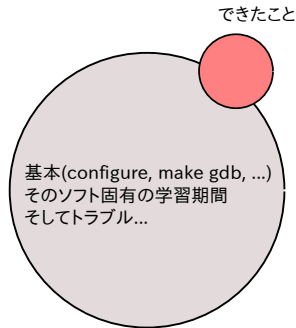
...だが持ったかも知れない感想

- ▶ 少しの拡張を施すのに苦勞が多すぎ?
- ▶ 望むらくは、再利用可能な初期投資
 - ▶ どんなソフトでも共通の事柄
(次のソフトはもっとさっといじれる)



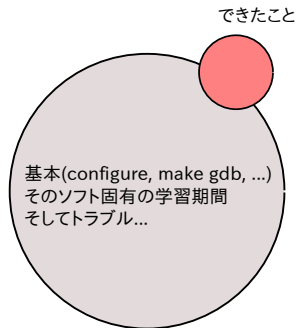
...だが持ったかも知れない感想

- ▶ 少しの拡張を施すのに苦勞が多すぎ?
- ▶ 望むらくは, 再利用可能な初期投資
 - ▶ どんなソフトでも共通の事柄
(次のソフトはもっとさっといじれる)
 - ▶ 一旦あるソフトに精通すれば,
そのソフトの拡張は**加速度的に容易になる**



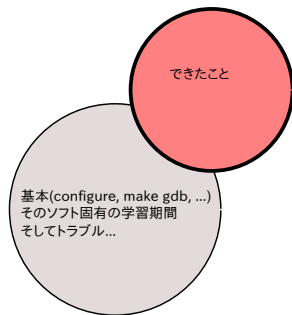
...だが持ったかも知れない感想

- ▶ 少しの拡張を施すのに苦勞が多すぎ?
- ▶ 望むらくは、再利用可能な初期投資
 - ▶ どんなソフトでも共通の事柄
(次のソフトはもっとさっといじれる)
 - ▶ 一旦あるソフトに精通すれば、そのソフトの拡張は**加速度的に容易になる**
 - ▶ 10の拡張を施すのに必要な労力は、1の拡張を施すのに必要な労力の10倍ではない



...だが持ったかも知れない感想

- ▶ 少しの拡張を施すのに苦勞が多すぎ?
- ▶ 望むらくは、再利用可能な初期投資
 - ▶ どんなソフトでも共通の事柄(次のソフトはもっとさっといじれる)
 - ▶ 一旦あるソフトに精通すれば、そのソフトの拡張は加速度的に容易になる
 - ▶ 10の拡張を施すのに必要な労力は、1の拡張を施すのに必要な労力の10倍ではない
- ▶ 研究の場合、目標は最初からもっと高めかつ自由な発想なのでその心配は(きっと)ない
 - ▶ 既存ソフトの利用はあくまで手段。目的ではない



研究とソフトウェアいじり・作り

- ▶ 世の中のソフトウェアには，もともとは一研究者 (グループ) の研究成果であったものも多い
 - ▶ LLVM, postgresql, BSD (OS), Cilk, MPI, Hadoop, ...

研究とソフトウェアいじり・作り

- ▶ 世の中のソフトウェアには、もともとは一研究者 (グループ) の研究成果であったものも多い
 - ▶ LLVM, postgresql, BSD (OS), Cilk, MPI, Hadoop, ...
- ▶ 研究をする時に既存のソフトを利用しないし拡張していることは、枚挙に暇がない
 - ▶ コンパイラの新しい最適化手法の研究 → LLVM
 - ▶ 新しいメモリ管理 (ガベージコレクション) 手法の研究 → Java 処理系 (JVM)
 - ▶ データベースの高速化研究 → postgresql, mysql, ...
 - ▶ 新しいファイルシステム → Linux OS, BSD, ...
 - ▶ ...

研究とソフトウェアいじり・作り

- ▶ 世の中のソフトウェアには、もともとは一研究者 (グループ) の研究成果であったものも多い
 - ▶ LLVM, postgresql, BSD (OS), Cilk, MPI, Hadoop, ...
- ▶ 研究をする時に既存のソフトを利用しないし拡張していることは、枚挙に暇がない
 - ▶ コンパイラの新しい最適化手法の研究 → LLVM
 - ▶ 新しいメモリ管理 (ガベージコレクション) 手法の研究 → Java 処理系 (JVM)
 - ▶ データベースの高速化研究 → postgresql, mysql, ...
 - ▶ 新しいファイルシステム → Linux OS, BSD, ...
 - ▶ ...
- ▶ アイデアが主役, 動機の源泉. そのアイデアを「しっかり実現」するために, 既存ソフトを借りる (こともある)

田浦研でも...

- ▶ 拡張機能を利用した拡張:
 - ▶ 強力なベクトル化機能 (LLVM)
 - ▶ 不揮発メモリを有効利用できるファイルシステム (Linux OS を拡張; 島津パイセン)
 - ▶ 既存のプログラムを SQL から呼び出し並列実行 (SQLite)
 - ▶ どんなデータにでも直接 SQL query がかけられる SQLite (島津パイセン)

田浦研でも...

- ▶ 拡張機能を利用した拡張:
 - ▶ 強力なベクトル化機能 (LLVM)
 - ▶ 不揮発メモリを有効利用できるファイルシステム (Linux OS を拡張; 島津パイセン)
 - ▶ 既存のプログラムを SQL から呼び出し並列実行 (SQLite)
 - ▶ どんなデータにでも直接 SQL query がかけられる SQLite (島津パイセン)
- ▶ 一から:
 - ▶ 超軽量スレッドライブラリ
 - ▶ ノード間で、擬似的なメモリ共有をするライブラリ (遠藤パイセン)
 - ▶ 並列分散シェル, make

田浦研でも...

- ▶ ソース修正:
 - ▶ 並列 GC (ゴミ集め; Boehm GC library)
 - ▶ C/C++に並列化機能を追加 (今の OpenMP の走り; EDG C/C++ frontend)
 - ▶ 通信に迅速に反応する Linux スケジューラ

いろいろないじり方・作り方

- ▶ ちなみに、実際にソースコードに手を出さなくてはいけないケースは、意外と少ない
- ▶ というよりも、人に使わせるにはいじらず済ませられたほうがよい
 - ▶ 拡張機能を利用 (LLVM パス, SQLite virtual table, Linux OS kernel module, ...)
 - ▶ コマンドとして利用
- ▶ それでも中身をしっかりと探れる能力は中身をいじる場合と同じく重要
 - ▶ 本格的な拡張機能は一体でデバッグする必要あり
 - ▶ ファイル形式
 - ▶ ドキュメント化されていない機能
 - ▶ etc.

- ▶ 以上が，この演習をやろうと思いついた人の「思い」
（「正当化」）
- ▶ 成功しているか否かはみなさんの評価・感想を待ちたいので，よろしくお願いします。