

Handle Execution results in clang-repl

Jun Zhang jun@junz.org Mentors: Vassil Vassilev & David Lange

1

About me

I'm Jun Zhang, 3rd year undergraduate student major in Software Engineering.

Working in the compiler research team since May 2022.

Last year Google Summer of Code student, Clang/LLVM contributor. (Land ~70 patches)



Project Background



An introduction to clang-repl

clang-repl is an interactive C++ interpreter that allows for incremental compilation, based on Clang and LLVM Orc JIT.

clang-repl> #include <iostream>
clang-repl> std::cout << "Hello, world!\n";
Hello,world!</pre>



Project Goals



Value pretty printing

clang-repl> int x = 42;

clang-repl> x

(int) 42

clang-repl> std::vector<int> v{1,2,3};

clang-repl> v

(std::vector<int> &) { 1, 2, 3 }



Compiled/Interpreted code interop

int Global = 42;

void setGlobal(int val) { Global = val; }

int getGlobal() { return Global; }

Interp.ParseAndExecute("void setGlobal(int val);");

Interp.ParseAndExecute("int getGlobal();");

Value V;

Interp.ParseAndExecute("getGlobal()", &V);

std::cout << V.getAs<int>() << "\n"; // Prints 42</pre>

The implementation (Highly inspired by Cling)



Overview





Top level expressions extension

clang-repl> int x = 42; clang-repl> x // Missing semicolon. Invalid in standard C++, but fine in the incremental C++.

Omit the semi tells clang-repl we want to capture the value of the DeclRefExpr (x)



Annotation token: annot_repl_input_end

945 + // Annotation for end of input in clang-repl.

946 + ANNOTATION(repl_input_end)

947 +

When parsing an ExprStmt and the last semi is missing, we'll pretend that there's one and set a marker for the late use.

Token *CurTok = nullptr; 547 // If the semicolon is missing at the end of REPL input, consider if 548 // we want to do value printing. Note this is only enabled in C++ mode 549 550 // since part of the implementation requires C++ language features. 551 // Note we shouldn't eat the token since the callback needs it. 552 if (Tok.is(tok::annot repl input end) && Actions.getLangOpts().CPlusPlus) 553 CurTok = &Tok; 554 else 555 // Otherwise, eat the semicolon. 556 ExpectAndConsumeSemi(diag::err expected semi after expr); 557 StmtResult R = handleExprStmt(Expr, StmtCtx); 558 559 if (CurTok && !R.isInvalid()) CurTok->setAnnotationValue(R.get()); 560 561 562 return R; 563



AST transformation

5460	+	
5461	+	<pre>if (Tok.is(tok::annot_repl_input_end) &&</pre>
5462	+	Tok.getAnnotationValue() != nullptr) {
5463	+	ConsumeAnnotationToken();
5464	+	<pre>cast<toplevelstmtdecl>(DeclsInGroup.back())->setSemiMissing();</toplevelstmtdecl></pre>
5465	+	}
5466	+	

Sema sets a flag after seeing the special token, so we know we should transform the AST before the real CodeGen process.

41	+	bool handleropLeverbect(DectGroupker DGk) override final {
42	+	<pre>if (DGR.isNull())</pre>
43	+	return true;
44	+	if (!Consumer)
45	+	return true;
46	+	
47	+	<pre>for (Decl *D : DGR)</pre>
48	+	<pre>if (auto *TSD = llvm::dyn_cast<toplevelstmtdecl>(D);</toplevelstmtdecl></pre>
49	+	TSD && TSD->isSemiMissing())
50	+	<pre>TSD->setStmt(Interp.SynthesizeExpr(cast<expr>(TSD->getStmt()));</expr></pre>
51	+	
52	+	<pre>return Consumer->HandleTopLevelDecl(DGR);</pre>
53	+	}

heal HandleTerlevelDeel(DeelCreveRef DCR) evennide fi



Code synthesis

clang-repl> x \rightarrow _clang_Interpreter_SetValue(x); // pseduo code.

 $\mathsf{DeclRefExpr} \to \mathsf{CallExpr}$

- 619 + // This synthesizes a call expression to a speciall 620 + // function that is responsible for generating the Value. 621 + // In general, we transform: 622 + // clang-repl> x 623 + // To: 624 + // // 1. If x is a built-in type like int, float. + // __clang_Interpreter_SetValueNoAlloc(ThisInterp, OpaqueValue, xQualType, x); 625 + // // 2. If x is a struct, and a lvalue. 626 + // clang Interpreter SetValueNoAlloc(ThisInterp, OpaqueValue, xQualType, 627 + // &x); 628 + // // 3. If x is a struct, but a rvalue. 629 + // new (__clang_Interpreter_SetValueWithAlloc(ThisInterp, OpaqueValue, 630 + // xQualType)) (x); 631 632 +
- 633 + Expr *Interpreter::SynthesizeExpr(Expr *E) {



Value runtime

A value is a container that can carry the arbitrary result of an expression in an endian-independent way with small buffer optimization.

class Value {
public:
 clang::QualType* getType(); // Obtain the type information of the expression.
 template<typename T>
 T castAs(); // Cast the value to corresponding type.
 void printType(llvm::raw_ostream& OS);
 void printOata(llvm::raw_ostream& OS);
 void print(llvm::raw_ostream& OS);
 void dump() const; // Dump the value, called print(llvm::outs()) internally.
};

The value that holds the information of the expression can be passed around after construction.



Pretty print implementation

If the user asks for the Value, we pass it as output parameter.

Or we perform pretty printing: Invoke Value::dump()

```
+ llvm::Error Interpreter::ParseAndExecute(llvm::StringRef Code, Value *V) {
306
307
      +
          auto PTU = Parse(Code);
308
          if (!PTU)
309
            return PTU.takeError();
310
          if (PTU->TheModule)
311
      +
            if (llvm::Error Err = Execute(*PTU))
312 +
             return Err;
313 +
314 +
          if (LastValue.isValid()) {
315
            if (!V) {
316
     +
317 +
             LastValue.dump();
             LastValue.clear();
318
     +
319
            } else
              *V = std::move(LastValue);
320
      +
321
      +
          return llvm::Error::success();
322
323 + 1
324 +
```



Implementation of Value::dump

If the type is a builtin type:

Just print it directly.

Else:

Synthesize a call to another runtime function: PrintValueRuntime(const T*)

519	+	} else {
5 <mark>2</mark> 0	+	// All fails then generate a runtime call, this is slow.
521	+	<pre>SS << SynthesizeRuntimePrint(V);</pre>
522	+	}
523	+	return Str;



std::string PrintValueRuntime(const T*)

All overloads live in a header, which are included at runtime.

So print a std::vector is equivalent to: PrintValueRuntime(&v);

271	<pre>+ static std::string SynthesizeRuntimePrint(const Value &V) {</pre>
272	<pre>+ Interpreter &Interp = const_cast<interpreter &="">(V.getInterpreter());</interpreter></pre>
273	<pre>+ Sema &S = Interp.getCompilerInstance()->getSema();</pre>
274	+ ASTContext &Ctx = S.getASTContext();
275	+
276	+ // Only include this header once and on demand. Because it's very heavy.
277	<pre>+ static bool Included = false;</pre>
278	+ if (!Included) {
279	+ Included = true;
280	+ llvm::cantFail(
281	<pre>+ Interp.Parse("#include <clang_interpreter_runtime_printvalue.h>"));</clang_interpreter_runtime_printvalue.h></pre>

This means users can write their own overload for their types:

clang-repl> struct S{}; clang-repl> std::string PrintValueRuntime(const S* s) { return "My printer!"; } clang-repl> S{} (S) "My Printer!"

How we make it



Submit the RFC

RFC: Handle Execution Results in clang-repl





1 🖋 Feb 15

TL;DR: Synthesize automatic printf to print execution results in clang-repl and generalize the approach to use an object used to bridge compiled/interpreted code taking inspiration from what was done in Cling.

Introduction

The Cling interpreter is a unique interpretative technology for C++ based on Clang developed by highenergy physics (HEP). It is used to deliver reflection and type information for exabytes of scientific data and is heavily used during data analysis of particle physics data from the Large Hadron Collider (LHC) and other particle physics experiments.

In RFC Moving (parts of) the Cling REPL in Clang 1 we discussed and shipped the initial incremental compilation facilities into LLVM mainline, called clang-repl.

In this RFC we propose two distinct features and their interaction: automatic printf and connecting compiled and interpreted C++ through a class called Value as an abstraction layer used to carry expression results and support value pretty printing in clang-repl.

Goals

Automatic printf

One of the key aspects of interactive C++ is exploratory programming which encourages showing execution results on screen easily. Typing every time printf or similar is too laborious and too annoying. Taking inspiration from Cling, we could achieve this effect by an extension that lives purely in libclangInterpreter. We propose to have a special mode to indicate when we want to do value pretty printing: A expression in the global scope (without the semicolon). Coincidently Rust takes a similar approach:

Spend almost one month writing the RFC and discussing the implementation with the community

Submit p	atches to Phabricator
Differential > D148997	
🌣 [clang] Add a new	annotation token: annot_repl_input_end
Closed OPublic	Differential > D141215
Authored by junaire on Apr 2	 3 2023, 00:06 Clang-repl] Introduce Value to capture expression results Closed OPublic
Made 2 patches in but still one l	eft! Authored by junaire on Jan 8 2023, 14:58.
Status Author Revision	
 Needs Review junaire D146809 [clang-rep Closed junaire D141215 [clang-rep] Implement Value pretty printing

What I learned



Large patches are hard to get it in!

Crazy long revision history!

Diff 1	508073	1c18b17	
Diff 2	509306	f8d8cb5	Update + Rebase
Diff 3	510168	0c983b8	Update
Diff 4	510288	08aad49	CHANGELOG:
Diff 5	511871	457bf61	Rebase
Diff 6	513024	d40a321	Update
Diff 7	513927	1494a32	Sync
Diff 8	513971	a08b090	Rebase
Diff 9	514160	a08b090	Update
Diff 10	517836	3ad0351	Update
Diff 11	518266	2575ad3	
Diff 12	518287	fcdedcf	
Diff 13	518289	fcdedcf	
Diff 14	518458	84312e2	Fix AutoType sugar
Diff 15	520912	11 [405a	Fix
Diff 16	520913	11f405a	Remove unused code.
Diff 17	520914	11f405a	Add some comments
Diff 18	520998	11f405a	Export symbols in Windows.
Diff 19	521961	11f405a	Don't use C++17 because Clang on Windows is not default to that :(
Diff 20	521962	11f405a	Add macro guard.
Diff 21	521981	11 <mark>f</mark> 405a	* include <tuple> to avoid incorrect lookup on Windows</tuple>
Diff 22	521985	bf4ea60	Make and use our own std::void_t
Diff 23	524088	d20afbd	
Diff 24	524091	d20afbd	
Diff 25	524093	d20afbd	add 'caas' namespace
Diff 26	524096	d20afbd	
Diff 27	525420	cf1ef41	Address comments.
Diff 28	525425	cf1ef41	Remove 'Interpereter::getParser' + More clean up

HY122 S10838 411934 Update DH13 S1084 All screenstaldSConumer DH14 S1084 All screenstaldSConumers DH14 S1084 All screenstaldscreenst	Diff 31	510286	3624769	Fix known memory issues.
NYI 2 S11934 411924 Add boremental/ATConsumer DNY 3 S11934 411934 Menve old code DNY 3 S11934 41094 Menve old code DNY 3 S11934 40016 Commental/ATConsumer DNY 3 S1187 40016 Commental/ATConsumer DNY 3 S1187 40016 Address comments DNY 3 S1187 40016 Address comments DNY 4 S1207 91545 us indice cover if we don't have it DNY 4 S1208 91545 us indice cover if we don't have it DNY 4 S1208 91545 us indice cover if we don't have it DNY 4 S1208 91545 tot indice cover if we don't have it DNY 4 S1208 91545 tot indice cover if we don't	Diff 32	510885	4110934	Update
DH1451000410004610004DH135100040000Cons up the patchDH135110040000Cons up the patchDH135110040000Cons up the patchDH135110040000Amore some unnectsaary code.DH145110040000Address one connects.DH145110040000Address one connects.DH145120040000Address one connects.DH145120060000Backte fiben/Manajor/repl.pop.end parts on form toDH145120060000Backte fiben/Manajor/repl.pop.end parts on form toDH155120061200Address one connects.DH155120061200Address one connects. <th>Diff 33</th> <th>511084</th> <th>4110934</th> <th>Add IncrementalASTConsumer</th>	Diff 33	511084	4110934	Add IncrementalASTConsumer
DH153 511537 44001b Update DH154 51180 4801b Cone up the patch. DH154 51180 4801b Robers cone uncessary code. DH154 51180 4801b Robers cone uncessary code. DH144 51190 4801b Robers cone uncessary code. DH144 51190 4801b Robers cone uncessary code. DH144 51190 4801b Robers cone uncessary code. DH144 51201 4801b Robers cone uncessary code. DH145 51201 4801b Robers cone uncessary code. DH145 51201 915455 ded include snew if we don't have it and uncessary code. DH145 51201 915455 ded include snew if we don't have it and uncessary code. DH145 51201 915455 ded include snew if we don't have it and uncessary code. DH145 51201 915455 for code moments DH145 51202 Address cone conneents DH145 51203 915203 for code moments pont code moments pont code moments <th>Diff 34</th> <th>511090</th> <th>4110934</th> <th>Remove old code</th>	Diff 34	511090	4110934	Remove old code
DH16 S11000 Allonia Allonia DH17 S1100 Allonia Barvas consuets DH19 S1100 Allonia Barvas consue consectavic code. DH14 S1100 Allonia Barvas consue consectavic code. DH14 S1100 Allonia Barvas consue consectavic code. DH14 S1100 Allonia Address conse consentis DH14 S1100 Allonia Kostow stow DH14 S1200 Allonia Kostow stow DH14 S1200 S1300 Allonia Kostow stow DH14 S1200 S1300 Allonia Kostow stow DH14 S1200 S1300 Allonia Kostow stow Kostow stow DH14 S1200 S1300 Allonia Kostow stow Kostow Kostow DH14 S1200 S1300 Allonia Kostow Kostow Kostow DH14 S1200 Kostow Kostow Kostow Kostow DH14 S1200 Kostow Kos	Diff 35	511857	4f8f01b	Update
DH12 2H301 4H011 Review some unnexesson code. DH13 5H107 4H011 Review some unnexesson code. DH14 5H107 4H011 Review some unnexesson code. DH14 5H207 4H011 Review some unnexesson code. DH14 5H207 4H011 Review some unnexes DH14 5H207 4H011 Kosen e nits DH14 5H207 4H011 Kosen e nits DH14 5H207 4H011 Kosen e nits DH14 5H207 9H345 Kosen e nits DH14 5H207 9H345 Kosen e nits DH14 5H208 9H345 Kosen e nomments DH14 5H209 9H345 Kosen e nomments DH14 5H209 9H345 Kosen e nomments DH14 5H204 Revisions buildoits Revisions buildoits DH14 5H204 Revisions buildoits Revisions buildoits DH14 5H204 Revisions buildoits Revisionsoutildoits DH14<	Diff 36	511862	4f8f01b	Clean up the patch
DH18 SH100 440016 Remote some unnecessary code DH19 SH104 440016 Address some unnecessary code DH14 SH109 440016 Address some unnecessary DH14 SH109 440016 Address some unnecess DH14 SH109 913456 red DH14 SH109 480468 Address more conneces DH14 SH109 Address more conneces Bette IndemNational (reg. Unput_red parts out from the DH145 SH1205 Address	Diff 37	511867	4/8/016	Address comments
NY 39 S1994 48016 Redicts how we comple a destructor DW 40 S1999 48016 Address some comments DW 41 S1207 48016 Address some comments DW 42 S1207 48016 Known his DW 43 S1207 48016 Known his DW 44 S1207 48016 Known his DW 43 S1208 91545 Gott includer crews if we don't have it cluding the r DW 44 S1209 915456 Kol Montess some comments DW 47 S1919 GA4546 Address more comments Montess DW 43 S1930 GA4548 Address more comments Montess DW 43 S1930 GA1538	Diff 38	511870	4f8f01b	Remove some unnecessary code.
DH140 S10900 Alfeins some comments DH141 S1007 40016 Scome comments DH142 S1007 40016 Scome comments DH143 S1007 40016 Scome comments DH144 S1007 40016 Scome comments DH143 S1007 40016 Scome comments DH144 S1007 40016 Scome comments DH143 S1028 S10456 Get include scneen if we don't have it DH144 S1028 GetS446 Address some comments DH145 S1028 GetS446 Address some comments DH145 S1028 GetS426 Address some comments DH145 S1028 Forsilow bubbbs GetS146 DH154 S1028 S10228 GetS406 GetS406 DH155 S1028 S10228 GetS406 GetS406 DH154 S1028 S10278 GetS406 GetS406 DH155 S1028 S10278 GetS406 GetS406 GetS40	Diff 39	511984	4f8f01b	Refactor how we compile a destructor
DH14 S1027 44001s Kackens some comments DH142 S1027 44001s Kackens nik DH143 S1027 44001s Kackens nik DH144 S1028 915845 Las Incomparing Deparational Advanced Statistics, twhen including the r DH145 S1287 915845 Las Incomparing Deparational Advanced Statistics, twhen including the r DH145 S1289 915845 Each Include - nears of twe don't have it DH146 S1289 915845 Kackens more comments DH149 S1393 464564 Address some comments DH149 S1393 464564 Address some comments DH149 S10301 490318 Result plasse. DH149 S10302 915220 Result plasse. DH159 S10203 915723 Orige nablealong Interpret_Strahluk-Copylor support in DH159 S10203 915723 Orige nablealong Interpret_Strahluk-Copylor support in DH159 S10203 915723 Orige nablealong Interpret_Strahluk-Copylor support in DH159 S10203	Diff 40	511999	4f8f01b	Address some comments
DH142 Storm effective DH143 Storm 440016 Value increased induces developmental dev	Diff 41	512072	4/8/016	Address some comments
NH 41 51037 4801b. Use incrementalPursur-SickMarsgirdName including the re- Dirf 43 51030 915345. Good Include crees in fead of statistic, is who including the re Dirf 43 51030 915345. Good Include crees in fead of statistic, is who including the re Dirf 43 51030 915445. Good Include crees in fead of statistic, is who including the re Dirf 43 51030 63045. Good Include crees in fead of statistic, is who including the re Dirf 43 510301 63045. Fead of statistic, is who including the regulary	Diff 42	512073	4f8f01b	fix some nits
DH1 4 51201 915455 due unsigned long instand of statising when including her at DH1 45 51203 915455 due include scene if we don't have it DH1 45 51209 915455 fc du Include scene if we don't have it DH1 45 51209 624545 Address some comments Include scene if we don't have it DH1 45 51201 624546 Address more comments Include scene it is DH1 49 51201 624546 Address more comments Include scene it is DH1 49 51201 624548 Address more comments Include scene it is DH1 50 51201 696348 fbesche IsGemitkaling/repLiput_end parts out from the Dh1 50 Dh1 50 Fbesche IsGemitkaling/repLiput_end parts out form the Dh1 50 Fbesche IsGemitkaling/repLiput	Diff 43	512075	4f8f01b	Use IncrementalParser::GetMangledName instead
DH143 S12337 915456 dott include snews if we don't have if DH144 S12389 915456 field DH144 S12389 915456 field DH144 S12389 64564 Address some connents DH149 S13381 464564 Address some connents DH149 S13080 959388 Beak the IdemMining' regLinput, end parts out from the DH153 DH154 S12030 959388 Beak the IdemMining' regLinput, end parts out from the DH153 DH153 S12031 Fischione buildots DH154 S12226 Advalut, please, some connents DH154 S12235 Printaly address some connents DH154 S12235 S12236 Only enablediagleinterpreter_StriblacCopylor support in DH154 S12335 S12723 Only enablediagleinterpreter_StriblacCopylor support in DH154 S12335 S12732 . DH154 S12935 Idfolia Advance conneents. DH154 S12935 Idfolia Advance conneents. DH154 S129337<	Diff 44	512801	915a45c	use unsigned long instead of std:size_t when including the r
DH14 51289 915455 K cal DH147 51991 64846 Address since comments DH14 51993 64846 Address since comments DH14 51993 64846 Address since comments DH15 51000 690336 Beak the IdsemMinang/ reg.Lipst.ext parts out from the DH15 51000 690336 Kenkla jbass. Final jbass. DH15 51001 690336 Kenkla jbass. Final jbass. DH15 51002 Final jbass.ext build/obt. Final jbass.ext build/obt. DH15 51230 6004 wablestop_linter/step.Striblau/Copyler support in DH15 51230 95723 C DH15 51230 16471 Adversa comments from datenshiman, theats! DH15 51230 74194 Adversa comments.	Diff 45	512837	915a45c	dont include <new> if we don't have it</new>
DH147 51919 cx48-64 Address none comments DH14 51939 tx4934 Address none comments DH15 51931 tx4934 ktability jetasse. DH15 51930 tx444 ktability jetasse. tx4934 DH15 519310 tx41114 jetasse. tx4934 DH15 512230 Advantsers for void & member pointers types tx4934 DH15 512230 Advantsers for void & member pointers types tx4934 DH15 512230 Advantsers for void & member pointers types tx4934 DH15 512320 4dvantsers for void & member pointers types tx4934 DH15 512320 4d4 unitters for void & member pointers types tx4934 DH15 512320 4d4 more comments tx49344 DH15 51232 4d4 more comments tx49344 DH15<	Diff 46	512869	915a45c	fix ci
DH14 51303 ca44-34 Address more comments DH149 51305 41544 Resale DH15 51000 690316 Resale to Edeminising regul, pupul, end parts out from the DH15 51000 690316 Resale to Edeminising regul, pupul, end parts out from the DH15 51000 690316 Resale to Edeminising regul, pupul, end parts out from the DH15 51001 612020 Resale to Edeminising regul, pupul, end parts out from the DH15 51020 612020 Add uniteration to comments 000 mobile_calon_torroret_setHubacCopyArr suport in DH150 512020 957720 . . DH15 512030 612010 . . DH150 512030 464982 comments from Gaaron-bailman; thankel . DH151 512030 7x1140 Address comments. . DH154 512030 7x1140 Add comments. . DH154 52030 7x1140 Add comments. . DH154 52237 535584 Add comments. <td< th=""><th>Diff 47</th><th>513918</th><th>ca4fb46</th><th>Address some comments</th></td<>	Diff 47	513918	ca4fb46	Address some comments
DH149 51954 ke1044 Relaxet DH150 51000 990380 Beak the Istem Milaniny' repLipsyL and parts out from the DH151 51000 990380 Beak the Istem Milaniny' repLipsyL and parts out from the DH153 51001 900380 Beak the Istem Milaniny' repLipsyL and parts out from the DH153 51001 500380 Add uniteets for word on member pointers types DH154 51225 Finality address some comments Only and the some comments Only and the some comments DH154 51225 951723 - Only and the some comments Only and the some comments DH154 51250 951723 - Address comments from Quarchalimain; thanks1 DH154 51250 616117 Address comments Address comments DH154 52030 701104 - - DH154 52037 730546 Address comments - DH154 52037 730546 Address comments - DH154 52037 730546 Address comments - DH154 522547 530546 Address comments	Diff 48	513926	ca4fo46	Address more comments
DH150 51000 993338 Denak the EdemiNikaning' reg. Lipuk_end parts out from the DH151 51000 993338 Denak the EdemiNikaning' reg. Lipuk_end parts out from the DH151 51000 993338 Denak the EdemiNikaning' reg. Lipuk_end parts out from the DH153 51000 993338 Denak the EdemiNikaning' reg. Lipuk_end parts 6000 DH153 510338 5122328 Knishinass bulkhols: 60000 60000 60000 <th>Diff 49</th> <th>513954</th> <th>fe1f344</th> <th>Rebase</th>	Diff 49	513954	fe1f344	Rebase
DHY 5 S16011 990338 Rebuild, please. DHY 2 517329 b12230 fit initiation buildback DHY 3 512430 b12230 fit initiation buildback DHY 3 512430 b12230 Advention buildback DHY 3 512430 b12230 Advention brown boundback DHY 3 512430 b12230 Advention brown boundback DHY 3 512430 b12230 Advention brown boundback DHY 3 512430 b127230 Advention brown boundback DHY 3 512430 b127240 Advention brown boundback DHY 3 512430 b157240 Advention brown boundback DHY 3 512430 b167140 Addess comments brown balance, thankal DHY 4 51241 Addess comments brown balance, thankal Boundback DHY 4 51241 Addess comments brown balance, thankal Boundback DHY 4 51241 Addoments Boundback Boundback DHY 4 51241 Addoments Boundback Boundback	Diff 50	516080	f99b38b	Break the IsSemiMissing/ repl_input_end parts out from the
DHY 20 217230 b12/230 K windows buildbots DHY 30 511245 b12/230 Advallay address some comments DHY 50 512245 b12/230 Advallay address some comments DHY 50 512245 b12/230 Advalunters for void & member pointers types DHY 50 512245 b12/230 Advalunters for void & member pointers types DHY 50 512245 b12/230 Advalunters for void & member pointers types DHY 50 512245 b12/230 advalunters for void & member pointers types DHY 50 512545 clint1 Add more comments DHY 50 512541 clint1 Add more comments DHY 50 512541 Add more comments clint1 DHY 50 512541 Add comments clint1 DHY 50 52257 553554 Add comments clint1 DHY 50 52257 553554 Add comments clint1 DHY 50 522565 2413ccd memore whitepace clint1	Diff 51	516081	f99b38b	Rebuild, please.
DH15 Stilleta N12220 Patralay address some comments DH15 Stilleta N12230 Add unifiestin for void & member pointer types DH15 Stilleta Stilleta Only mable_clang_hoterpoter_SettabueCopyArr support in DH15 Stilleta Stilleta Address comments from Guaenchabilman; thanks: DH16 Stilleta Address comments Stilleta Address comments DH164 Stilleta Address comments Stilleta Address comments DH164 Stilleta Address comments Stilleta Address comments DH164 Stilleta Stilleta Address comments Stilleta Address comments	Diff 52	517829	b12c23b	fix windows buildbots
DHY 54 S12226 Add unitests for void & member pointers types DHY 55 S1226 PS/2723 Origy enablesleng_interpreter_S4tNukCepyArr support in DHY 57 S1228 PS/7723 . DHY 57 S1239 PS/7723 . DHY 57 S1230 PS/7723 . DHY 57 S1250 Fd/6117 Address comments from Quaeronbalman; thanks1 DHY 58 S1230 Pd/6117 Add more comments. DHY 54 S12302 Pd/1147 Repaire - Comments. DHY 54 S12302 Pd/1147 Address comment from Vasal, for DHY 54 S12302 S13544 Address comment from Vasal, for DHY 54 S12455 S145442 Address comment from Vasal, for	Diff 53	518166	b12c23b	Partially address some comments
DH153 518285 977722 Only enable_slang_interpreter_SettAbleCopyArs support in DH156 518295 977723 . DH153 518295 9578723 . . . DH154 518295 9578723 . . . DH153 518295 9578723 . . . DH154 518395 9578723 . . . DH154 518395 9578723 . . . DH155 518305 616417 Address comments from Quaronbailman; thanks1 . DH164 518991 effek17 For yoos . DH164 51991 effek17 For yoos . DH164 520328 761104 . . DH164 52237 538544 Address comments from Quaronbailman; then Quar	Diff 54	518265	b12c23b	Add unittests for void & member pointers types
DHY 5 313330 977723 . DHY 57 518330 957973 . DHY 58 595972 . . DHY 59 51830 167376 . DHY 59 519302 rdf4171 Address comments from Quaronhaliman, thankal DHY 69 51990 efd417 Add more comments. DHY 64 51951 efd417 Fix types DHY 64 52032 7x11946 . DHY 64 52237 53554 Address comments from Vasal, thr DHY 64 52257 535544 Address comments from Vasal, thr DHY 64 522545 2aflcsac remove whitespace	Diff 55	518285	957d723	Only enableclang_Interpreter_SetValueCopyArr support in
DHY 57 518232 8757723 . DHY 58 518435 461704 A . DHY 59 518532 46147 A dat more comments from Guaron-bailman, thanktal DHY 59 518932 eff417 A dat more comments. DHY 61 51891 eff417 Fix types DHY 62 52022 Yathye Rebase + Update DHY 64 520325 7stlibe - DHY 64 52357 558454 Address comments from Vasal, thr DHY 64 252455 2afleset yathers from Vasal, thr	Diff 56	518290	957d723	
DHY 50 514520 (H2170) . DHY 50 518920 (Hd17) Address comments from Quaero.bailman; thanks! DHY 60 518900 (Hd17) Address comments. DHY 61 518900 (Hd17) Address comments. DHY 62 520320 741940 Rebase + Update DHY 63 530320 7a0194 . DHY 64 520327 538454 Address comments from Vasal, the DHY 64 522457 538454 Address comments from Vasal, the DHY 64 522458 2aflcase remove whitepace	Diff 57	518292	957d723	
DHY 59 518932 cft6117 Address comments from @uaron.balmam, thanisal DHY 60 51899 cft6141 Add more comments. DHY 61 51899 cft6141 Fix typos DHY 62 520214 All more comments. DHY 63 520204 All more comments. DHY 64 520235 Antine 4. DHY 64 522545 Add comments. DHY 64 522545 Add comments. DHY 64 522545 Address comments from Vasol, thw. DHY 64 522545 2draces.	Diff 58	518450	1f6370b	
DHY 60 318900 effel17 Add more comments. DHY 61 S1980 effel17 Fix spos DHY 62 S20232 7419% Rebase + Update DHY 64 S20236 7419% Add comments DHY 64 S20236 S598454 Add comments DHY 65 S22350 S598454 Add comments DHY 64 S22355 S40dress comments from Vasol, thx DHY 64 S22454 241cacd remove whitegace	Diff 59	518952	cffd417	Address comments from @aaron.ballman, thanks!
DHY 61 510951 effet17 Fix types DHY 62 520324 7401% Rease - Update DHY 64 520328 7611% Add comments DHY 64 520329 550454 Add comments DHY 64 522545 24dcase comment from Vesal, thr DHY 64 522545 24fcase remove whitegase	Diff 60	518960	cffd417	Add more comments.
DHY 62 S20224 74/HW Rebase - Update DHY 63 S30303 7cl/bek . DHY 64 S20237 S35854 Address comments DHY 64 S20545 Zdderss comments from Vesal, thr DHY 64 S21545 Zdderss comment from Vesal, thr	Diff 61	518961	cffd417	Fix typos
DWF 06 S20320 Crol10e1 . DWF 06 S22327 S358d54 Add comments DWF 06 S22530 S358d54 Address comment from Vacal, thx DWF 06 S22545 2datasc remove whitespace	Diff 62	520324	743ff9c	Rebase + Update
DHF 64 322527 5558d54 Add comments DHF 65 522530 5558d54 Address comment from Vasal, thx DHF 66 522545 2d1cacd remove whitespace	Diff 63	520326	7cd10e4	
DHF 65 522530 5558d54 Address comment from Vasal, thx DHF 66 522545 2d1cacd remove whitespace	Diff 64	522527	5558d54	Add comments
Diff 66 522545 2d1cacd remove whitespace	Diff 65	522530	5558d54	Address comment from Vassil, this
	Diff 66	522545	2d1cacd	remove whitespace



Writing portable code is hard!



To revert, or not to revert, it's a question...

aosp-O3-polly-before- vectorizer-unprofitable	10752	
openmp-clang-x86_64-linux- debian	37790	
clang-hexagon-elf	11870	
llvm-clang-win-x-armv7l	12097	
sanitizer-x86_64-linux- autoconf	37728	
ppc64le-flang-rhel-clang	70187	
sanitizer-aarch64-linux- bootstrap-msan	2731	
llvm-clang-win-x-aarch64	13329	

Thank you!

Special thanks to:

Aaron Ballman Axel Naumann Lang Hames Richard Smith Stefan Gränitz

