Down the rabbit hole...

Array copy - “real C”

```c
void copy (char *to, char *from, int count)
{
   while (count > 0)
       *to++ = *from++, count--;
}
```

Op Cache

OpCache

- **CFG-based Optimization** (0x0010): Utilizes Control Flow Graphs for code analysis and optimization, enhancing performance by restructuring the execution flow for efficiency.
- **SSA-based Optimization** (0x0020): Employs Static Single Assignment form to simplify variable usage and improve code analysis, leading to more efficient execution paths.
- **Sparse Conditional Constant Propagation** (0x0080): Optimizes code by evaluating and propagating constants at compile time, reducing runtime computation needs.

OpCache

- **Temporary Variable Optimization** (0x0100): Reduces overhead by optimizing the usage and allocation of temporary variables within the code.
- **Dead Code Elimination** (0x2000): Removes code segments that do not affect the program's outcome, streamlining execution and reducing resource consumption.
- **Literal Optimization** (0x0400): Enhances performance by optimizing the handling and storage of literal values within the code.

OpCache

- Debugging option that produces an opcode dump after individual passes.

```
php -d opcache.opt_debug_level=0x20000 test.php
```

```
$_main:
        ; (lines=5, args=0, vars=0, tmps=0)
        ; (after optimizer)
        ; /home/dev/Benchmark/test.php:1-82
0000 INIT_FCALL 0 80 string("benchsingle")
0001 DO_UCALL
0002 INIT_FCALL 0 80 string("benchdouble")
0003 DO_UCALL
0004 RETURN int(1)

benchSingle:
        ; (lines=1, args=0, vars=0, tmps=0)
        ; (after optimizer)
        ; /home/dev/Benchmark/test.php:3-38
0000 RETURN string("GoodbyeCruelWorld hello World Today")

benchDouble:
        ; (lines=1, args=0, vars=0, tmps=0)
        ; (after optimizer)
        ; /home/dev/Benchmark/test.php:43-78
0000 RETURN string("GoodbyeCruelWorld hello World Today")
```

Short-circuit evaluation

Short-circuit evaluation

if ($a && $b) { … }

```
cmp    a, 0

jne    LABEL_A

---more code---

LABEL_A:

cmp    b, 0

jne    RETURN_LABEL

 ---more code---
```

if ($a & $b) { … }

```
and   a, a, b

cmp   a, 0

jne   RETURN_LABEL

---more code---
```

Short-circuit evaluation

- Compiler to rescue?

  https://gcc.godbolt.org/z/x4c19z3zz

Short-circuit evaluation

- All CPU are superscalar.
- Can do a multiple things at the same time.
- Speculative execution can actually execute both parts.