

SWPP 2024-01

# Sprint : 0 proposal

서종환 / 임재환 / 김민지

# 팀원 소개

## 서종환

- 전기·정보공학부  
수리과학부 복수전공

## 임지환

- 지역시스템공학과
- 컴공 부전공

## 김민지

- 자유전공학부
- 컴공 + 정보문화학  
연합전공

# Project Objectives

- 대규모 프로젝트 협업 및 의사소통 능력 함양
- 새롭게 접한 아키텍처를 최적화하는 경험을 통한 개발 능력 향상

## 6 optimizations with example codes

1. 특별한 연산 없이 변수 저장만 하는 조건문 삼항연산자로 대체
2. 조건문 분기 최적화
3. loop를 통해 반복되는 연산 벡터 레지스터 연산으로 대체
4. Shift operation을 Multiplication/Division operation으로 변환
5. 코드 이동
6. 함수 인라이닝

# 1. 특별한 연산 없이 변수 저장만 하는 조건문 삼항연산자로 대체

```
int a = 0, b = 0, c = 0;  
b = func1();  
c = func2();  
if (b < c) a = 3;  
else a = 9;
```



```
bool cond = b < c;  
a = cond ? 3 : 9;
```

; Before optimization

```
cmp eax, ebx  
je .L1  
mov ecx, eax  
jmp .L2  
.L1:  
mov ecx, ebx  
.L2:
```

; After optimization

```
cmp eax, ebx  
cmovne ecx, eax  
cmovne ecx, ebx
```

## 2. 조건문 분기 최적화

```
int d;  
bool cond1, cond2, cond3;
```

```
if (cond1) {  
    if (cond2) {  
        if (cond3) d = 1;  
        else d = 2;  
    }  
    else {  
        if (cond3) d = 3;  
        else d = 4;  
    }  
}  
else {  
    if (cond2) {  
        if (cond3) d = 5;  
        else d = 6;  
    }  
    else {  
        if (cond3) d = 7;  
        else d = 8;  
    }  
}
```



```
if (cond1 && cond2 && cond3) d = 1;  
else if (cond1 && cond2 && !cond3) d = 2;  
else if (cond1 && !cond2 && cond3) d = 3;  
else if (cond1 && !cond2 && !cond3) d = 4;  
else if (!cond1 && cond2 && cond3) d = 5;  
else if (!cond1 && cond2 && !cond3) d = 6;  
else if (!cond1 && !cond2 && cond3) d = 7;  
else d = 8;
```



```
int ret;  
unsigned int n;  
if (n == 1) ret = 1;  
else if (n == 2) ret = 2;  
else if (n == 3) ret = 3;  
else ret = 4;  
  
n = ()  
switch(n) {  
    case 1:  
        ...  
        break;  
    case 2:  
        ...  
        break;  
    case 3:  
        ...  
        break;  
    default:  
        ...  
}
```

### 3. loop를 통해 반복되는 연산 벡터 레지스터 연산으로 대체

```
// Before vectorization
for (int i = 0; i < 4; i++) {
    a[i] = b[i] + c[i];
}

// After vectorization (pseudo-code)
for (int i = 0; i < 4; i += 4) {
    vector_a[i:i+4] = vector_b[i:i+4] + vector_c[i:i+4];
}
```

### 4. Shift operation을 Multiplication/Division operation으로 변환

```
int a = 1;
a <<= 1;    →    a *= 2;
```

## 5. 코드 이동

```
vector<int> v(n);  
for (int i = 0; i < n - 1; ++i) {  
    int temp = v[i];  
    v[i + 1] = v[i];  
    v[i] = temp;  
}
```



```
int temp;  
for (int i = 0; i < n - 1; ++i) {  
    v[i + 1] = v[i];  
    v[i] = temp;  
}
```

## 6. Function inlining

Module &M

for (Function &F : M)

if (F.empty() || F.isDeclaration()) -> continue

if (shouldinline(F)) -> inlineFunction(F)