Dash - YES

yes, please

TEAM 나

Summary: YES, we gonna make yes

Version: 1

```
#include <stdio.h>
int main()
    while (1)
        printf("y\n");
```

### Faster

#### Much Faster

/usr/bin/yes

#### **About Yes**

yes(예스)는 유닉스 및 유닉스 계열 운영 체제의 명령어의 하나로, 긍정적인 반응이나 사용자 지정 텍스트 문자열을 kill 시점까지 계속 출력한다.

### Ex\_01

```
void main() {
   while(write(1, "y\n", 2)); [5.94MiB/s]
```

# 내장된 GNU의 YES

```
THE RESIDENCE THE PROPERTY.
   Ti unage (Del status)
 to production todactions:

or of an area productions
   the beautiful nature a line with all payoffine children, or "v", but
     V/i
"), black();
*/pata (OELP_SPTIDE_BESCHITTER, statut);
*/pata (OERSIDE_SPTIDE_BESCHITTER, statut);
***soft_menillary_loris (PRODRAW_MAPS);
}
        /* Degree performance by using a buffer size greater than Buffill / 2. 1/
      ther "but a reuse operand strings ? "operants a smaller (bufaller);
```

[5.76GiB/s]

# Ex\_01과 GNU 결과값 비교

- 왜 차이가 발생할까?
- 우리는 얼마나 더 빠르게 할 수 있을까?

### Ex\_02

```
#define LEN 2
#define TOTAL LEN * 1000
int main() {
    char yes [LEN] = \{'y', '\setminus n'\};
    char *buf = malloc(TOTAL);
    int used = 0;
    while (used < TOTAL) {
        memcpy(buf+used, yes, LEN);
        used += LEN;
while(write(1, buf, TOTAL));
return 1;
```

#### [2.96GiB/s]

#### Ex\_03

```
#define LEN 2
#define TOTAL 8192
int main() {
    char yes [LEN] = \{'y', '\n'\};
    char *buf = malloc(TOTAL);
    int bufused = 0;
    while (bufused < TOTAL) {
        memcpy(buf+bufused, yes, LEN);
        bufused += LEN;
    while(write(1, buf, TOTAL));
    return 1;
```

[5.99GiB/s]

# 프로젝트 제안 목적

같은 기능의 프로그램도 코드 최적화에 따라 발생하는 성능차이를 배운다.

## Team NA

