

au VimEnter \* call Bye2024()

# 대학생의 터미널 생활 적응기

은은수 (ewhk9887@gmail.com)

# Intro

은은수

- 한국외국어대학교 재학중
- Vim 사용 경험:
  - 現: Arch + Neovim
  - 舊: IdeaVim

# Intro

**Topic: 대학생과 터미널과 Vim**

- **Vim 사용기: JetBrains에서 터미널로**
- **Vim을 어떻게 쓰고 있나요?**

# Vim 사용기: JetBrains에서 터미널로

- 터미널 이전에는:
  - JetBrains의 제품들을 주로 이용했고, 편의성을 위해 IdeaVim 이용
  - 개발 시작 시점부터 이미 vscode와 JetBrains가 주류였다!

# Vim 사용기: JetBrains에서 터미널로

- 터미널로 이사가기:
  - 로컬 환경 변화: 윈도우 → 우분투 → 아치리눅스
    - 많은 아치 유저가 Neovim을 이용!
  - IDE에서 Vim모드를 쓰다보니 Native Vim에 대한 갈망이
  - 훌륭한 커스텀과 가벼움 → 정착

# Vim을 어떻게 쓰고 있나요?

- IDE 겸, 메모장 겸, 검사검사 모든 걸 수행하는 텍스트 편집기
- 개발도, 과제도, 필기도, CTF 풀이도 모두 Neovim에서

# Vim을 어떻게 쓰고 있나요?

주 사용 용도: 당연하게도 코딩입니다.



```
1  / SPDX-License-Identifier: MIT
2
3  pragma solidity ^0.8.21;
4
5  // Define a Tweet Struct with author, content, timestamp, likes
6  // Add the struct to array
7  // Test Tweets
8
9  contract Twitter {
10     // define our struct
11     struct Tweet {
12         address author;
13         string content;
14         uint256 timestamp;
15         uint256 likes;
16     }
17
18     // add our code
19     mapping(address => Tweet[]) public tweets;
20
21     function createTweet(string memory _tweet) public {
22         Tweet memory newTweet = Tweet({
23             author: msg.sender,
24             content: _tweet,
25             timestamp: block.timestamp,
26             likes: 0
27         });
28
29         tweets[msg.sender].push(newTweet);
30     }
31
32     function getTweet(
33         address _owner,
```

# Vim을 어떻게 쓰고 있나요?

neovim은 아주 훌륭한 메모장입니다!

오른쪽 사진은 동아리 인수인계 자료인데,  
render-markdown을 이용하였어요.

```
43
44 정도 물어보고
45 너무 아니다 싶은 게 아니면 회비 납부 후 디스코드 서버에 초대해주시면 됩니다.
46 디코 서버 초대할 때 오프라인으로 동방 방문해서 명부 작성할 것, 디코 자기소개할 것 요청해주세요.
47
48 # 1-5. 기존 부원 명부 작성 및 회비 납부
49
50 위의 포맷대로
51 ● 단과대학: (총정시에 있는 그대로)
52 ● 학번 9자리: (총정시 기준)
53 ● 성명:(개명 시 총정시 기준)
54 ● 연락처: (010-0000-0000 포맷)
55 ● 재학/휴학 여부(군휴학, 일반휴학, 교환): (재학, 일반휴학 다 말로 써줄것)
56 ● 현재 학기: (0-0학기 포맷)
57 ● 개인정보 수집 동의여부: (0, X 포맷)
58
59 디엠으로 받아놓으시면 되고 유의사항은
60 ● 졸업유예생, 졸업생, 대학원생, 군휴학생, 국내외 학점 교류학생은 명부에 올릴 수 없습니다.
61 ● 명부는 받아놓으시되 해당 명단을 동연에 제출하는 명단에서 제외시켜야합니다.
62 ● 학부/학과/전공을 총정시에 있는 그대로 작성해야합니다.
63   ○ 예) Language & Trade, 경영학부, 경영학전공
64   ○ 안 되는 예) 경영, LT, 일언문 이런 식으로 명단에 바로 올릴 수 없게 주면 다시 적어달라고 해주세요
65   ○ 그렇지 않으면 나중에 다 공식명칭을 찾아봐야합니다.
66   ○ 저 명단 여부가 오탈자나 사실과 다른 부분이 있으면 동연에서 연락이 옵니다.
67
68 해당 과정을 디코봇으로 자동화할 수 있는지는 생각중에 있습니다.
69 개강 후 1주일까지 받는데, 개강 전부터 받기 시작하면 좋습니다.
70
71 @ 1-6. 편집장 선출
72
73 편집장을 선출해야합니다. 인수인계는 이전 편집장에 의해서 이루어지며, 미리 뽑아놓기만 하면 됩니다.
74 개강 이후 뽑아도 무관합니다.
```

**NORMAL** 해갈 회장 인수인계 자료.md

# Vim을 어떻게 쓰고 있나요?

```
20
21 EncoderDecoderModel은 사전 학습된 자동 인코딩(autoencoding) 모델을 인코더로, 사전 학습된 자가 회귀(autoregressive) 모델을 디코더로 활용하여 시퀀스->
   투-시퀀스(sequence-to-sequence) 모델을 초기화하는 데 이용됩니다.
22
23 사전 학습된 체크포인트를 활용해 시퀀스-투-시퀀스 모델을 초기화하는 것이 시퀀스 생성(sequence generation) 작업에 효과적이라는 점이 Sascha Rothe, Shashi N
   arayan, Aliaksei Severyn의 논문 Leveraging Pre-trained Checkpoints for Sequence Generation Tasks에서 입증되었습니
   다.
24
25 EncoderDecoderModel이 학습/미세 조정된 후에는 다른 모델과 마찬가지로 저장/불러오기가 가능합니다. 자세한 사용법은 예제를 참고하세요.
26
27 이 아키텍처의 한 가지 응용 사례는 두 개의 사전 학습된 BertModel을 각각 인코더와 디코더로 활용하여 요약 모델(summarization model)을 구축하는 것입니
   다. 이는 Yang Liu와 Mirella Lapata의 논문 Text Summarization with Pretrained Encoders에서 제시된 바 있습니다.
28
29 ② 모델 설정에서 EncoderDecoderModel을 무작위 초기화하기 Randomly initializing `EncoderDecoderModel` from model configurations.
30
31 EncoderDecoderModel은 인코더와 디코더 설정(config)을 기반으로 무작위 초기화를 할 수 있습니다. 아래 예시는 BertModel 설정을 인코더로, 기본 Bert
   ForCausalLM 설정을 디코더로 사용하는 방법을 보여줍니다.
32
33 python
34 >>> from transformers import BertConfig, EncoderDecoderConfig, EncoderDecoderModel
35
36 >>> config_encoder = BertConfig()
37 >>> config_decoder = BertConfig()
38
39 >>> config = EncoderDecoderConfig.from_encoder_decoder_configs(config_encoder, config_decoder)
40 >>> model = EncoderDecoderModel(config=config)
41
```

# Vim을 어떻게 쓰고 있나요?

CTF에서 Neovim은 아주 훌륭한 도구입니다.

- 모든 것을 터미널 안에서 해결하기:
  - gdb로 프로그램 분석
  - 익스플로잇 코드 → Neovim으로 작성
  - Write-up → Neovim으로 작성

# Vim을 어떻게 쓰고 있나요?

- tmux는 훌륭한 도구입니다!

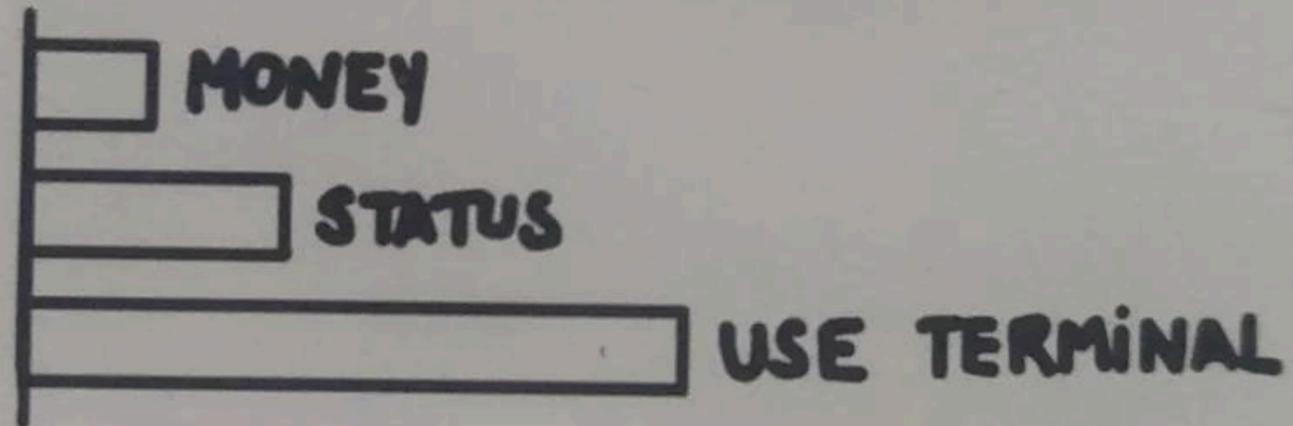
```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int compare(const void *a, const void *b) {
4     if (*(int *)a > *(int *)b) {
5         return 1;
6     } else if (*(int *)a < *(int *)b) {
7         return -1;
8     } else {
9         return 0;
10    }
11 }
12
13 int main(void) {
14     int n;
15     scanf("%d", &n);
16
17     int *arr = (int *)malloc(n * sizeof(int));
18
19     for (int i = 0; i < n; i++) {
20         scanf("%d", &arr[i]);
21     }
22
23     int x;
24     scanf("%d", &x);
25
26     int cnt = 0;
27
28     qsort(arr, n, sizeof(int), compare);
29     int i = 0;
30     int j = n - 1;
31     for (; i < j; ) {
32         if (arr[i] + arr[j] == x) {
```

```
1
2 undefined8 main(void)      ■ Unknown type name 'und
3
4 {
5     long in_FS_OFFSET;
6     int local_30;
7     int local_2c;
8     int local_28;
9     uint local_24;          ■ Use of undeclared identifie
10    int local_20;
11    int local_1c;
12    void *local_18;
13    long local_10;
14
15    local_10 = *(long *)(&in_FS_OFFSET + 0x28);
16    __isoc99_scanf(&DAT_00102004, &local_30);      ■
17    local_18 = malloc((long)local_30 << 2);      ■ Ca
18    for (local_28 = 0; local_28 < local_30; local_28
= local_28 + 1) {
19        __isoc99_scanf(&DAT_00102004,          ■ Use of und
20            (void *)((long)local_18 + (long
)local_28 * 4));
21    }
22    __isoc99_scanf(&DAT_00102004, &local_2c);      ■
23    local_24 = 0;
24    qsort(local_18, (long)local_30, 4, compare);
25    local_20 = 0;
26    local_1c = local_30 + -1;
27    while (local_20 < local_1c) {
28        if (*(int *)((long)local_18 + (long)local_20 *
4) +
29            *(int *)((long)local_18 + (long)loc@0@
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 // Function to compare two integers for qso
rt
5 int compare(const void *a, const void *b) {
6     return (*(int *)a - *(int *)b); }
7
8 int main(void) {
9     int array_size;        // Size of the input
array
10    int target_sum;        // Target sum to fin
d pairs
11    int pair_count = 0;    // Count of pairs ma
tching the target sum
12    int left = 0, right;  // Two-pointer varia
bles
13    int *array;           // Pointer to dynami
cally allocated array
14
15    // Read the size of the array
16    scanf("%d", &array_size);
17
18    // Allocate memory for the array
19    array = (int *)malloc(array_size * sizeof
(int));
20    if (array == NULL) {
21        fprintf(stderr, "Memory allocation fail
ed\n");
22        return 1;
23    }
```

# 그래서 결론이?

HOW TO IMPRESS NORMAL PEOPLE?



`au VimEnter * call Bye2024()`

**Thank you**



`ewhk9887@gmail.com`



`maximizemaxwell`