FT_Preprocessor

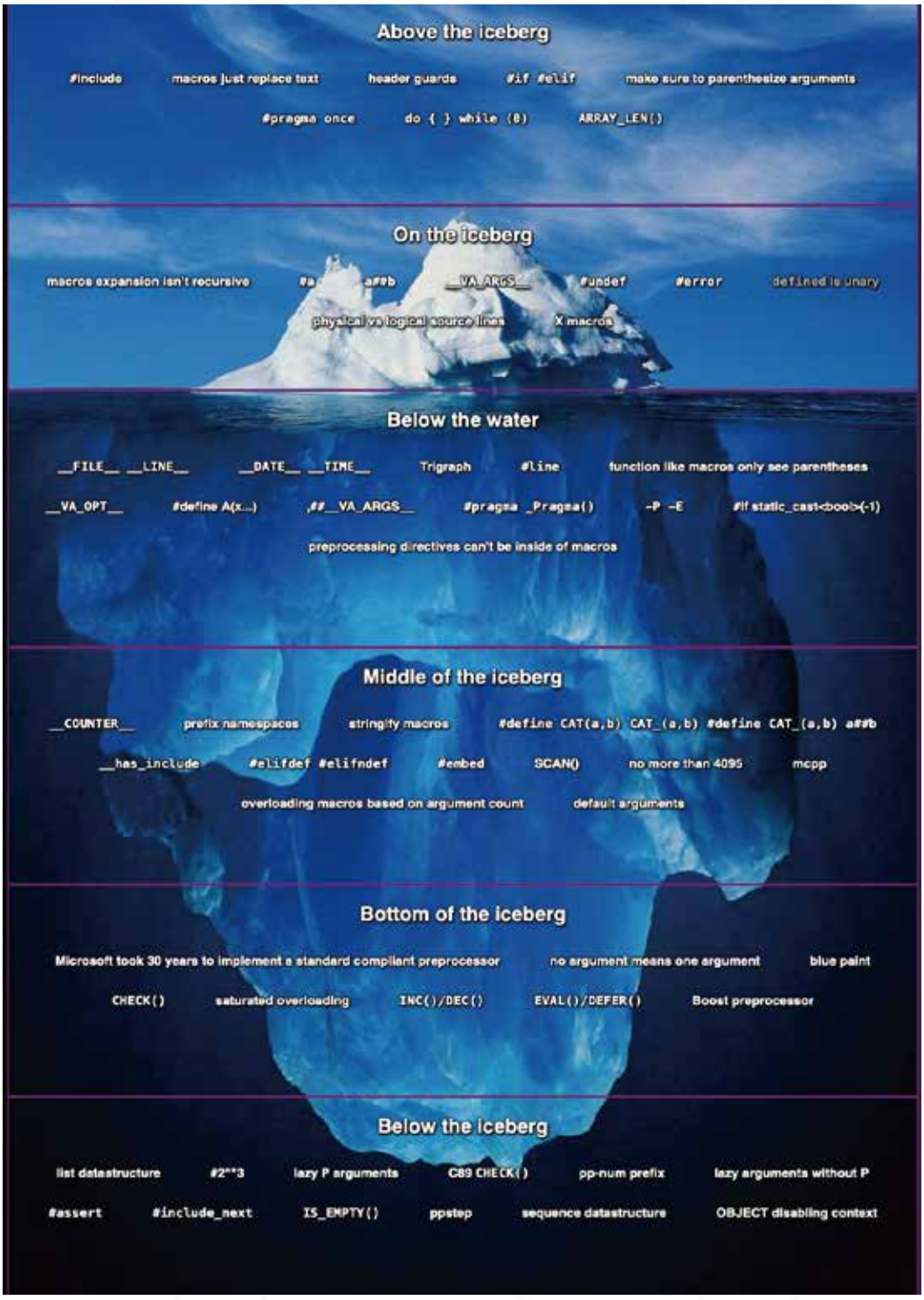
Summary:
This document contains the exercises of preprocessorc

Contents

I	Preamble	2
II	Introduction	3
III	General rules	4
IV	Exercise 00: Get File Name	5
V	Exercise 01: Endianness	7

Chapter I

Preamble



Chapter II

Introduction

This project aims to broaden your insight of preprocessor by using C. You will implement macro expansion, line control, and conditional compilation to your program serve multi environments.

Chapter IV

Excersize 00: FT_Get_Filename

Function name	ft_get_filename
Prototype	char* ft_get_filename(const char * path)
Parameters	path : path for search
Return value	filename: correct behavior null : an error occurred
Turn-in directory	ex00/
Files to turn in	ft_get_filename.c
Allow functions	in string.h, stdlib.h

Windows와 Unix 기반 OS는 파일 경로 표기법이 다릅니다.
전처리 지시기를 사용하여 하나의 함수로 두 운영체제에서 경로 상 파일 이름을 가져오는 함수를 작성하세요.

- 운영체제는 Windows와 macOS만 고려됩니다. 그 외의 운영체제에서 실행되는 경우 NULL을 반환합니다.
- .c 파일 내에서 코드에 전처리 지시기를 사용하여 운영체제 기준으로 분기하도록 구현하여야 합니다.
- path는 반드시 정상적인 경로가 입력됩니다.(경로의 마지막 문자가 구분자인 경우, 디렉토리로 판단합니다.)
- 경로가 디렉토리로 판단되는 경우, NULL을 반환합니다.
- 경로 상 파일 구분자가 없는 경우, NULL을 반환합니다.
- 함수 내에서 새로운 문자열을 할당해서 반환해야합니다.

Preprocessor에서 OS를 판단하는 방법을 찾아보세요

윈도우에서 파일의 경로를 표기하는 방법과 unix에서 파일을 표기하는 방법은 아래와 같습니다.

WINDOWS :

```
C:Users\su\Desktop\42.txt
```

UNIX :

```
/users/su/Desktop/42.txt
```

Chapter V

Excersize 01 : FT_HOSTCMP

Macro name	FT_HOSTCMP
Parameters	A : value of host B : value of network
Return value	0 : if equal non zero :
Turn-in directory	ex01/
Files to turn in	ft_hostcmp.h
Allow functions	none

컴퓨터의 메모리공간에 바이트를 배열하는 방법 또한 아키텍처에 따라 달라집니다. 이번엔 FT_HOSTCMP를 구현하면서 이를 알아볼 것입니다.

- 전달된 A와 B가 같으면 0으로 확장되고 아니면 다른 값이 나오는 Macro를 작성하세요.
- endian과 운영체제의 메모리 저장 방식을 고려하여 모든 경우에 동일한 동작을 할 수 있도록 작성하세요.
- Preprocessing 시에 FT_USE_LITTLE_ENDAIN, FT_USE_BIG_ENDIAN으로 endian을 비교하세요.
- Preprocessing 시에 FT_USE_32, FT_USE_64로 메모리 저장 방식을 비교하세요.

Big Endian과 Little Endian의 차이점을 생각해보세요

- 아래 제시된 메인문이 동작할 수 있도록 구현되어야 합니다.

```c
#define FT_USE_LITTLE_ENDIAN    1
#define FT_USE_32_BIT           1
#include "ft_hostcmp.h"

int main() {
    printf("%d\n", FT_HOSTCMP(8080, htonl(8080))); // print "1"
    return (0);
}
```

Conditional compilation에 대해 알아보세요