

Posix Basics

What is POSIX?

- POSIX는 Portable Operating System Interface의 약자이며 운영체제의 호환성을 유지하기 위해 IEEE에서 지정한 표준
- POSIX 표준을 준수하는 소프트웨어는 POSIX 표준을 준수하는 다른 운영 체제와 호환되어야 함.
- ex: ps명령을 사용하는 경우 OpenBSD, Debian, MacOS에서 동일하게 작동해야 함
- Q: window 에서 간혹 ls대신 dir를 사용하는데, 그럼 윈도우는 POSIX 표준을 준수하지 않는 건가?

A little History

- 정보화 시대 초기에 프로그래머들은 서로 다른 소프트웨어 작성 → 마이그레이션시 많은 비용 발생
- 이러한 문제를 극복하기 위해 POSIX가 탄생
- POSIX는 1980년 UNIX 시스템 인터페이스를 표준화 하기위해 시작

POSIX Versions

- 현재 POSIX.1의 버전은 IEEE Std 1003.1-2017
- POSIX 1은 애플리케이션의 이식성뿐만 아니라 프로세스 생성 및 종료, 프로세스 환경 조작, 파일 및 디렉토리 액세스, 심플한 I/O 시스템 서비스 동작까지 정의함.
- POSIX 2는 명령 인터프리터, 휴대용 셸 프로그래밍, 유저 환경 및 관련 유틸까지 포함함.
- 두 표준 둘다 기존 UNIX의 영향을 받음

POSIX Defined Standards

1. The C API

- a. POSIX는 C 언어로 표준을 정의
- b. ANSI(American National Standards Institute) C Standards에 더 많은 함수를 추가함
 - i. 파일 작업
 - ii. 프로세스, 스레드, 공유 메모리 및 스케줄링 매개변수
 - iii. 네트워킹
 - iv. 메모리 관리
 - v. 정규 표현식

2. General Concepts

- a. C API 이외에도 포인터 유형의 초기화 및 동시 실행에 대한 안정성과 같은 프로그램 작성 규칙도 추가
- b. 메모리 동기화에 대한 규칙도 적용

3. File Formats

- a. 파일, 표준 출력, 표준 에러, 표준 입력에 대한 문자열 서식 지정 규칙도 정의 ? : 형식 지정자
- b. ex:


```
printf "Today's Date: %d %s, %d" 18 September 2021
```

Today's Date: 18 September, 2021

4. Environments Variables

- a. 환경 변수는 로그인 시 환경 파일에 정의할 수 있는 변수
- b. 일반적으로 변수 이름은 대문자와 밑줄을 포함해야 함 숫자를 넣는 것을 권장하지 않음
- c. 환경 변수의 값은 이식 가능한 문자 집합에 정의된 문자 문자열이어야 함
- d. 예약된 환경 변수를 따르는 것이 좋음
 - i. COLUMN, HOME, LOGNAME, LINES, PATH, PWD 등등

5. Locale

- a. 로케일은 사용자 환경에서 사용되는 언어 및 문화 규칙을 정의
- b. 구현시 일관성을 유지하기 위해 정의된 로케일 환경 변수를 사용

c. POSIX Standards는 다음과 같은 환경 변수를 정의

- i. LC_TYPE → 문자분류
- ii. LC_NUMERIC → 화폐 서식
- iii. LC_NUMERIC → 숫자 서식
- iv. LC_TIME → 날짜 및 시간 서식
- v. LC_MESSAGES → 정보 메시지 및 로그와 같은 프로그램 메시지

6. Character Set

- a. POSIX에서는 구현에 적어도 하나의 문자 집합과 이식 가능한 문자 집합을 포함할 것을 권장

7. Regular Expressions

- a. POSIX를 준수하는 구현은 기본 정규 표현식 또는 확장 정규 표현식을 사용 가능

8. Directory Structure

- a. Linux 배포판들은 FHS를 준수
- b. FHS는 트리형 디렉토리 구조를 정의
- c. ex:

```
> tree / -d -L 1
/
├── Applications
├── Library
├── System
├── Users
├── Volumes
├── bin
├── cores
├── dev
├── etc -> private/etc
├── home -> /System/Volumes/Data/home
├── opt
├── private
├── sbin
├── tmp -> private/tmp
├── usr
└── var -> private/var
```

9. Utilities

- a. POSIX에서는 유틸리티 프로그램에서 다음과 같은 argument syntax를 구현할 것을 권장

utility_name [-a][-b][-c option_argument]
[-d]-e)[-f[option_argument]][피연산자...] <매개 변수 이름>

Operating Systems and POSIX Compliancy

1. Linux

- a. POSIX를 완전히 준수하는 Linux 기반 OS를 만드는 것은 가능 ex: EulerOS
- b. 그러나 대부분의 최신 프로그램들은 부분적으로 표준을 준수하지 않거나 전부 표준을 준수하지 않을 때도 있다. ex: bash 셸(POSIX를 준수했다가 최신버전에선 준수하지 않음)

2. Darwin

- a. macOS 및 iOS 같은 OS에 코어 셋
- b. 부분적으로 POSIX를 준수
- c. 최근 나온 macOS는 POSIX를 완전히 준수

3. Windows NT

- a. Microsoft Windows는 UNIX 운영체제와 설계가 완전히 다르기 때문에 표준을 전혀 준수하지 않음
- b. WSL 호환성 레이어 또는 Cygwin을 사용하여 POSIX 호환 환경 설정 가능

4. Others

- a. POSIX Certified
 - i. AIX
 - 1. HP-UX
 - ii. Oracle Solaris