목차

alternative 로 실행 명령어 버전 관리 3
alternatives 명령어의 주요 기능 3
주의사항 3
Master - Slave 관계 설정 4
Master-Slave 관계 설정 예제 4
1. 기본 자바 설치 경로 4
2. Master-Slave 등록 4
3. Master-Slave 작동 방식 4
Master-Slave 등록 상태 확인 5
결론 5
alternatives 명령어로 python 버전 관리하는 방법 5

alternative 로 실행 명령어 버전 관리

— *이강우 2024/12/20 11:06*

alternatives 명령어는 리눅스 시스템에서 여러 버전의 프로그램이 설치되어 있을 때, 기본적으로 사용할 프로그램을 관리하는 데 유용합니다. 예를 들어, java, editor, (x)emacs, (n)vi(m) 등의 프로그램에 대해 여러 버전이 존재할 때, alternatives를 통해 기본 프로그램을 설정하고 관리할 수 있습니다.

alternatives 명령어의 주요 기능

- **대체 프로그램 설정:** 여러 버전의 프로그램 중에서 기본으로 사용할 프로그램을 선택할 수 있습니다.
- **우선순위 지정:** 각 프로그램에 우선순위를 부여하여, 기본 프로그램을 자동으로 선택하도록 설정할 수 있습니다.
- **심볼릭 링크 관리:** 프로그램 실행 파일에 대한 심볼릭 링크를 생성하고 관리하여, 시스템 업데이트 시에도 설정이 유지되도록 합니다.

사용 예시:

em이라는 텍스트 편집기가 기본으로 설정되어 있으나, 유니코드를 지원하는 nem으로 변경하려는 경우:

- 기존 em 바이너리를 /opt/em-legacy/em2로 이동하고, 새로운 nem 바이너리를 /usr/local/bin/nem에 위치시킵니다.
- 다음 명령어를 사용하여 uemacs라는 이름으로 대체 프로그램을 등록합니다:

```
alternatives --install /usr/bin/em uemacs /opt/em-legacy/em2 1
alternatives --install /usr/bin/em uemacs /usr/local/bin/nem 99
```

이후, alternatives –config uemacs 명령어를 통해 기본 프로그램을 선택할 수 있습니다:

```
alternatives --config uemacs
```

명령어를 실행하면, 사용 가능한 프로그램 목록이 표시되며, 원하는 프로그램의 번호를 입력하여 선택할 수 있습니다.

주의사항

- alternatives 명령어는 시스템 전역 설정을 변경하므로, 관리자 권한이 필요합니다.
- 환경 변수(예: EDITOR, CC)의 설정은 /etc 또는 사용자의 홈 디렉토리($HOME/.profile)에서 직접 관리하는 것이 좋습니다.
- alternatives 명령어를 활용하면, 시스템에 설치된 여러 버전의 프로그램을 효율적으로 관리하고, 사용자나 스크립트의 요구에 맞게 기본 프로그램을 손쉽게 변경할 수 있습니다.

Master - Slave 관계 설정

Master-Slave 관계란? `alternatives`에서는 하나의 **마스터** 설정(예: `/usr/bin/java`)이 선택되면, 그와 연결된 **슬레이브** 설정들(예: `javac`, `javadoc`, `javaws` 등)이 함께 변경되도록 설정할 수 있습니다. 이를 통해 자바와 관련된 모든 바이너리 및 설정 파일이 동시에 변경됩니다.

Master-Slave 관계 설정 예제

아래는 `alternatives`를 사용해 자바 관련 설정에서 `master-slave` 관계를 구현하는 방법입니다.

1. 기본 자바 설치 경로

자바가 설치된 경로를 예로 들면:

- `/usr/lib/jvm/java-8-openjdk/bin/java`
- `/usr/lib/jvm/java-11-openjdk/bin/java`

2. Master-Slave 등록

각 버전의 자바에 대해 마스터와 슬레이브 관계를 등록합니다. 예를 들어:

```
alternatives --install /usr/bin/java java /usr/lib/jvm/java-8-
openjdk/bin/java 1 \
    --slave /usr/bin/javac javac /usr/lib/jvm/java-8-openjdk/bin/javac \
    --slave /usr/bin/javadoc javadoc /usr/lib/jvm/java-8-openjdk/bin/javadoc
\
    --slave /usr/bin/javaws javaws /usr/lib/jvm/java-8-openjdk/bin/javaws

alternatives --install /usr/bin/java java /usr/lib/jvm/java-11-
openjdk/bin/java 2 \
    --slave /usr/bin/javac javac /usr/lib/jvm/java-11-openjdk/bin/javac \
    --slave /usr/bin/javadoc javadoc /usr/lib/jvm/java-11-
openjdk/bin/javadoc \
    --slave /usr/bin/javaws javaws /usr/lib/jvm/java-11-openjdk/bin/javaws
```

3. Master-Slave 작동 방식

1. `/usr/bin/java`가 **마스터**로 설정됩니다.
2. `javac`, `javadoc`, `javaws`는 **슬레이브**로 등록되어 있습니다.
3. 사용자가 `alternatives –config java` 명령어로 자바 버전을 변경하면, 슬레이브들(`javac`, `javadoc`, `javaws`)도 동시에 변경됩니다.

Master-Slave 등록 상태 확인

다음 명령어를 사용하여 마스터와 슬레이브 관계를 확인할 수 있습니다.

```
alternatives --display java
```

출력 예:

```
java - status is auto.
 link currently points to /usr/lib/jvm/java-11-openjdk/bin/java
/usr/lib/jvm/java-8-openjdk/bin/java - priority 1
 slave javac: /usr/lib/jvm/java-8-openjdk/bin/javac
 slave javadoc: /usr/lib/jvm/java-8-openjdk/bin/javadoc
 slave javaws: /usr/lib/jvm/java-8-openjdk/bin/javaws
/usr/lib/jvm/java-11-openjdk/bin/java - priority 2
 slave javac: /usr/lib/jvm/java-11-openjdk/bin/javac
 slave javadoc: /usr/lib/jvm/java-11-openjdk/bin/javadoc
 slave javaws: /usr/lib/jvm/java-11-openjdk/bin/javaws
```

결론

1. alternatives의 **Master-Slave 기능**을 통해, 마스터 바이너리를 변경하면 관련된 슬레이브 바이너리들도 자동으로 변경됩니다.
2. 이는 자바 환경 설정의 일관성을 유지하기 위해 유용하게 사용됩니다.
3. 다른 프로그램(예: python)에서도 동일한 방식으로 적용할 수 있습니다.

alternatives 명령어로 python 버전 관리하는 방법

python 명령어에 심볼링 링크 설정 방법

```
alternatives --install /usr/bin/python python /usr/bin/python3.9 1
alternatives --install /usr/bin/python python /usr/bin/python3.12 2

alternatives --config python
```

python3 명령어에 심볼링 링크 설정 방법

```
alternatives --install /usr/bin/python3 python3 /usr/bin/python3.9 1
alternatives --install /usr/bin/python3 python3 /usr/bin/python3.12 2

alternatives --config python3
```

- master-slave로 구현

```
alternatives --install /usr/bin/python python /usr/bin/python3.9 1 \
    --slave /usr/bin/python3 python3 /usr/bin/python3.9

alternatives --install /usr/bin/python python /usr/bin/python3.12 2 \
    --slave /usr/bin/python3 python3 /usr/bin/python3.12

alternatives --config python
```

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/alternative_%EB%A1%9C_%EC%8B%A4%ED%96%89_%EB%AA%85%EB%A0%B9%EC%96%B4_%EB%B2%84%EC%A0%84_%EA%B4%80%EB%A6%AC

Last update: **2024/12/20 11:06**