

## (Operating System) Practice -6-

# exec



# Index

- I. Exec overview
- II. Exec practice – 1
- III. Exec practice – 2

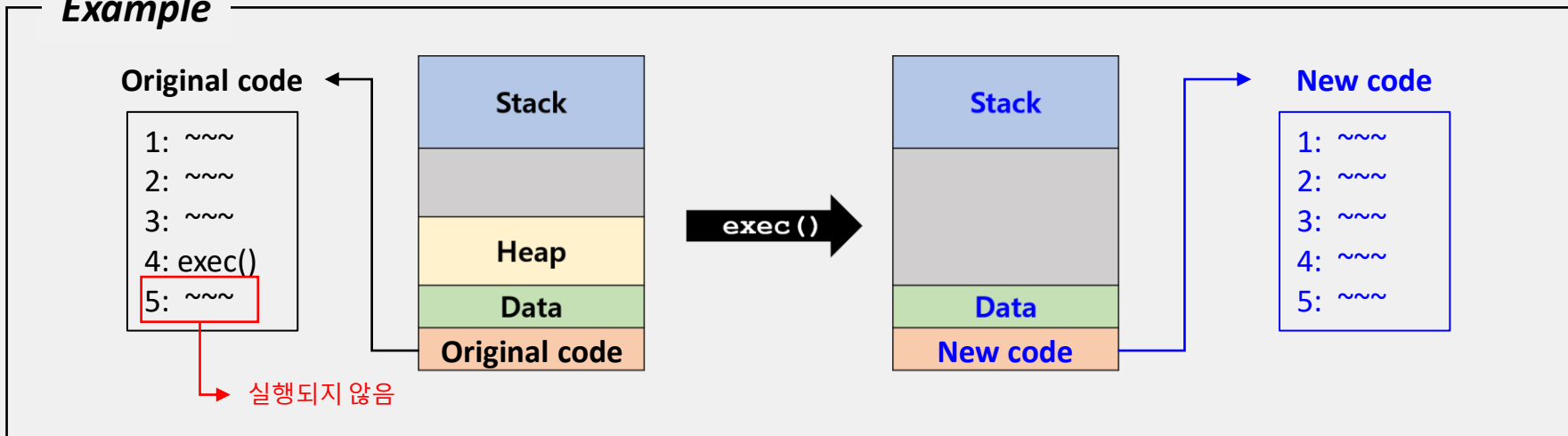


# Exec overview

## • Exec

- 다른 프로세스를 생성하고, 기존 프로세스를 종료함
- 정확히 말하면 덮어쓰기의 개념
  - ✓ 새로운 프로세스는 이전 프로세스가 사용하던 메모리를 그대로 사용함

### Example



# Exec overview

- Exec

- Exec

- ✓ -

- ✓ 형태

- `execl(const char *path, const char &arg, .....)`

→ 나열해서 인자 값 입력

- `execv(const char *path, char *const argv[])`

- ✓ 인수

→ 배열로 정의해서 인자 값 입력

- `const char *path`: 실행시키고자 하는 프로세스의 path

- `const char &arg`: 프로세스 실행을 위한 인수 (명령어, 옵션 등) → 원하는 만큼 입력 가능, 마지막에 Null

- `char *const argv[]`: 배열로 정의한 프로세스 실행을 위한 인수들

- ✓ 반환

- `int`: 비정상 작동시 -1을 반환

- ✓ `<unistd.h>` 헤더에 포함되어 있음

# Exec overview

- Exec

함수 이름
<code>int execl( const char *path, const char *arg, ...)</code>
<code>int execlp( const char *file, const char *arg, ...)</code>
<code>int execl( const char *path, const char *arg ,..., char * const envp[])</code>
<code>int execv( const char *path, char *const argv[])</code>
<code>int execvp( const char *file, char *const argv[])</code>
<code>int execve (const char *filename, char *const argv [], char *const envp[])</code>

함수 이름	프로그램 지정	명령라인 인수	함수 설명
execl	디렉토리와 파일 이름이 합친 전체 이름	인수 리스트	환경 설정 불가
execlp	파일 이름	인수 리스트	환경 설정 불가
execl	디렉토리와 파일 이름이 합친 전체 이름	인수 리스트	환경 설정 가능
execv	디렉토리와 파일 이름이 합친 전체 이름	인수 배열	환경 설정 불가
execvp	파일 이름	인수 배열	환경 설정 불가
excve	전제 경로 명	인수 배열	환경 설정 가능

# Exec practice – 1

- Exec practice – 1

- 기본 예제

```
#include <stdio.h>
#include <unistd.h>

void main(){
    printf("exec practice.\n");
    execl("/bin/ls", "ls", NULL);
    printf("Will this line appear?\n");
}
```

→ 이 라인이 출력될까요?

# Exec practice – 2

- Exec practice – 2
  - 선택해서 프로세스 실행

subfunc1.c

```
1  #include <stdio.h>
2
3  void main(){
4      printf("Strawberry\n");
5  }
```

subfunc2.c

```
1  #include <stdio.h>
2
3  void main(){
4      printf("Banana.\n");
5  }
```

exec\_practice.c

```
1  #include <stdio.h>
2  #include <unistd.h>
3
4  void main(){
5      int num;
6      printf("Which sub-function will you execute?: ");
7      scanf("%d", &num);
8      if(num==1){
9          execl("/home/sam/practice6/subfunc1", "./subfunc1", NULL);
10     }
11     else if(num==2){
12         char *arg[] = {"./subfunc2", NULL};
13         execv("/home/sam/practice6/subfunc2", arg);
14     }
15     else{
16         printf("There is no function%d.\n", num);
17     }
18 }
```