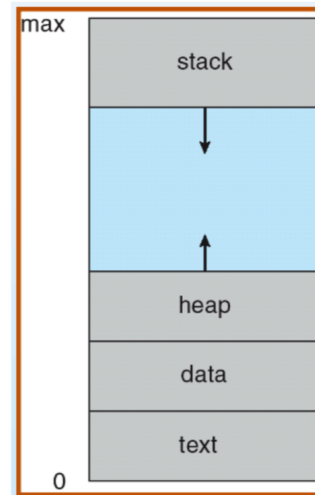


프로세스와 컨텍스트 스위칭

- 프로세스 구조 deep dive

프로세스와 컨텍스트 스위칭

- 프로세스 (process) 는 일반적으로 어떻게 구성되어 있을까?
 - text(CODE): 코드
 - data: 변수/초기화된 데이터
 - stack: 임시 데이터(함수 호출, 로컬 변수등)
 - heap: 코드에서 동적으로 만들어지는 데이터



OS.xlsx -> ProcessStructure

여기서 잠깐 복습!

- 스택 (stack)

| 이 자료구조는 뭐에 쓰이고 왜 강조해서 배웠나?

| datastructure.xlsx -> Stack

프로세스와 컴퓨터 구조 복습

- 프로세스 (process) 는 일반적으로 어떻게 동작할까? - 컴퓨터 구조도 복습
 - text: 코드
 - data: 변수/초기화된 데이터
 - stack: 임시 데이터(함수 호출, 로컬 변수등)
 - heap: 코드에서 동적으로 만들어지는 데이터
- PC(Program Counter) + SP(Stack Pointer)

OS.xlsx -> ProcessWithCS

Heap 이란?

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

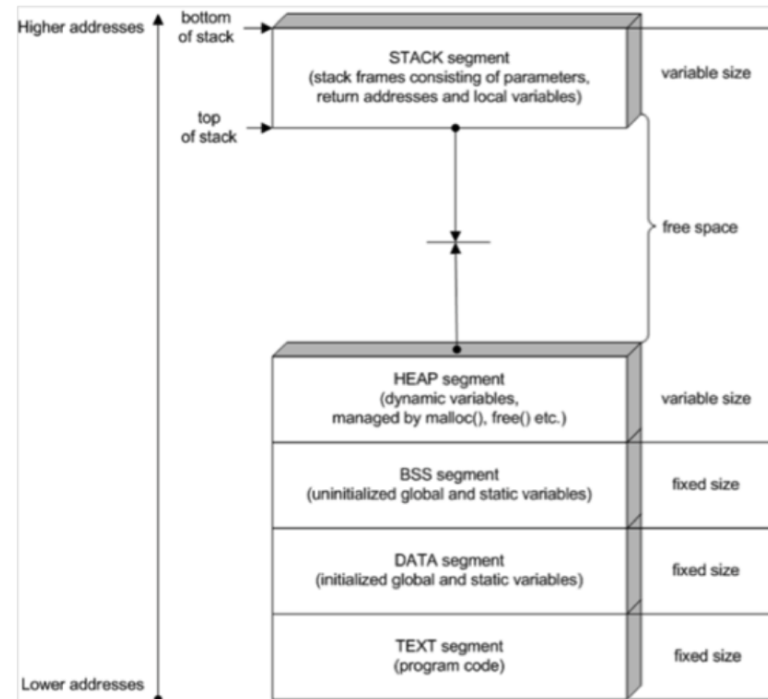
int main()
{
    int *data;
    data = (int *) malloc(sizeof(int));
    *data = 1;
    printf("%d\n", *data);

    return 0;
}
```

OS.xlsx -> ProcessWithHeap

프로세스 구조: Stack, HEAP, DATA(BSS, DATA), TEXT(CODE)

DATA를 BSS와 DATA로 분리

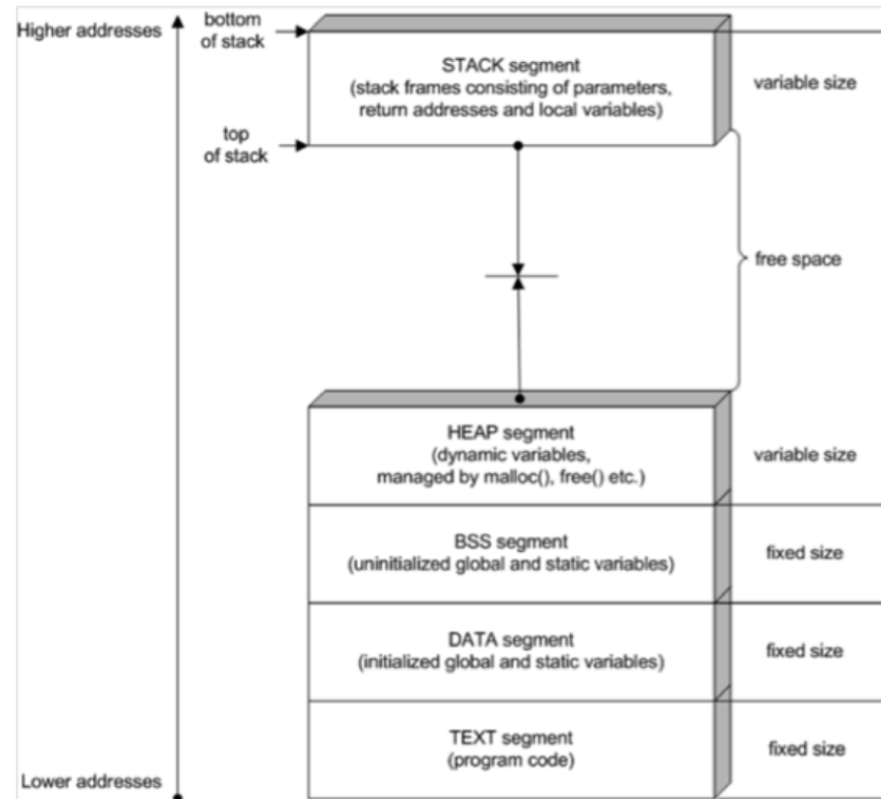


출처: <http://www.drdobbs.com/security/anatomy-of-a-stack-smashing-attack-and-h/240001832>

OS.xlsx -> ProcessAll

가볍게 듣기

- 스택 오버플로우: 주로 해커들의 공격에 활용되었음



출처: <http://www.drdoobs.com/security/anatomy-of-a-stack-smashing-attack-and-h/240001832>

프로세스와 컨텍스트 스위칭

| 다시 PC, SP에 주목하자.

- PC(Program Counter) + SP(Stack Pointer)

| Stack, HEAP, DATA(BSS, DATA), TEXT(CODE)

| OS.xlsx -> ProcessAllWithCS

PCB

| PC, SP는 어디에 저장하나?

Process Control Block (PCB) 에 저장!

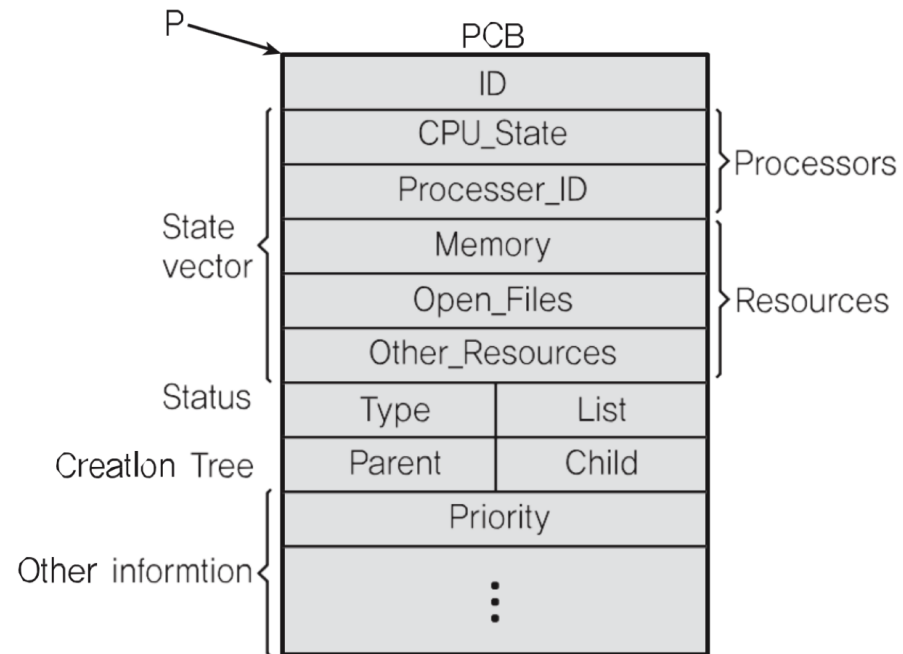
| Process Context Block 이라고도 함

1. Porcess ID
2. Register 값 (PC, SP 등)
3. Scheduling Info (Porcess State)
4. Memory Info (메모리 사이즈 limit)
- ...

| PCB: 프로세스가 실행중인 상태를 캡처/구조화해서 저장

PCB: 리눅스 예

1. Porcess ID
2. Register (PC, SP 등)
3. Scheduling Info (Porcess State)
4. Memory Info (메모리 사이즈 limit)



정리

- 프로세스 구조
 - Stack, HEAP, DATA(BSS, DATA), TEXT(CODE)
- PCB
 - 프로세스 상태 정보 - PC, SP, 메모리, 스케줄링 정보등