

دوره مهندسی معکوس نرم افزار

- Site: OnHexGroup.ir
- Youtube: @onhexgroup
- Telegram: onhex_ir
- Twitter: @onhexgroup
- Github: onhexgroup

نسخه ی ۳۲ و ۶۴ بیتی

پلتفرم: ویندوز

ارائه دهنده : **OnhexGroup**

■ موقع اجرای برنامه در مموری ایجاد میشه.

Youtube: Onhexgroup

پشته

Process Memory

STACK

HEAP

CODE

Global Data

← آدرس بالا

← آدرس پایین

Onhexgroup.ir

کاربردهای پشته

- آدرس بازگشت در توابع
- پارامترهای ورودی توابع
- متغیرهای محلی توابع
- وضعیت رجیسترها (فراخوانی توابع یا تغییر سطح دسترسی)

Twitter:Onhexgroup

پشته

■ پشته یک ساختار داده ای براساس **LIFO** هستش.

■ **LIFO = Last Input First Output**



Github: Onhexgroup

پشته

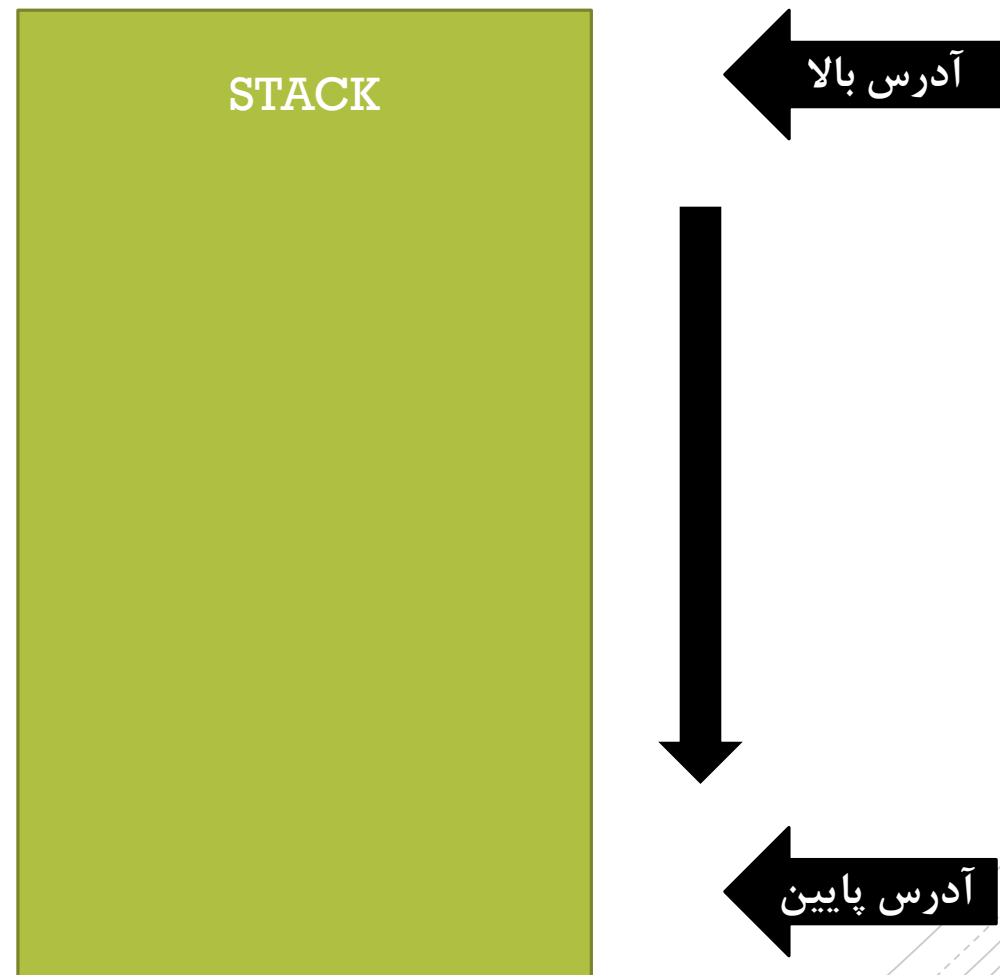
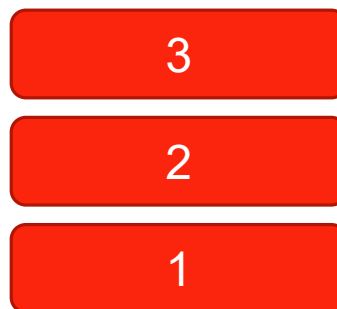
■ فضای آدرس دهی بصورت پیش فرض ۱ مگابایت

■ هر آدرس:

■ در ۳۲ بیتی: سائز آدرس ها ۳۲ بیت یا ۴ بایت است.

■ در ۶۴ بیتی: سائز آدرس ها ۶۴ بیت یا ۸ بایت است.

■ پشته رو به آدرس های پایین تر رشد میکنه.



Youtube: Onhexgroup

پشته

2

1

STACK

3

1000

آدرس بالا

آدرس پایین

Onhexgroup.ir

پشته

1

STACK

3

1000

2

999

آدرس بالا

آدرس پایین

Github: Onhexgroup

پشته

STACK

3

1000

2

999

1

998

آدرس بالا

آدرس پایین

Twitter: Onhexgroup

پشته

1

STACK

3

1000

2

999

آدرس بالا

آدرس پایین

Insta: Onhexgroup

پشته

2

1

STACK

3

1000

آدرس بالا

آدرس پایین

Telegram: onhex_ir

پشته

3

2

1

STACK

آدرس بالا

آدرس پایین

■ پشته یک ساختار داده ای از نوع **LIFO**

■ اشاره گر به بالای پشته: **ESP** یا **RSP**

Youtube: Onhexgroup

پشته

2

1

STACK

آدرس بالا

آدرس پایین

Onhexgroup.ir

پشته

RSP

STACK

2

1000

1

آدرس بالا

آدرس پایین

twitter: Onhexgroup

پشته

RSP

STACK

2

1000

1

999

آدرس بالا

آدرس پایین

Github: Onhexgroup

پشته

RSP

STACK

2

1000

999

1

آدرس بالا

آدرس پایین

Insta: Onhexgroup

پشته

RSP

STACK

1000

999

2

1

آدرس بالا

آدرس پایین

■ برای قرار دادن در پشته از **PUSH** استفاده میکنیم.

PUSH R/M/IMM

Youtube: Onhexgroup

دستورات پشته

نسخه	نوع عملوند	سایز پشته
۳۲ بیتی	مقدار صریح	۴ بایت کم میکند
	رجیستر	در حالت بایتی و Dword ۴ بایت و در حالت WORD ۲ بایت کم میکنند.
	مموری	در حالت بایتی و Dword ۴ بایت و در حالت WORD ۲ بایت کم میکنند.
۶۴ بیتی	مقدار صریح	حالت Qword ممنوع. بقیه حالات ۸ بایت کم میکنند.
	رجیستر	حالت بایتی و Dword ممنوع. حالت WORD ۲ بایت حالت QWORD ۸ بایت
	مموری	حالت بایتی و Dword ممنوع. حالت WORD ۲ بایت حالت QWORD ۸ بایت

Youtube: Onhexgroup

دستورات پشته

2

1

STACK

آدرس بالا

آدرس پایین

Onhexgroup.ir

پشته

PUSH 2

1

ESP

STACK

2

1000

آدرس بالا

آدرس پایین

Twitter: Onhexgroup

پشته

PUSH 1

STACK

2

1000

1

996

ESP

آدرس بالا

آدرس پایین

■ برای برداشتن از پشته از **POP** استفاده میکنیم.

POP R/M

Github: Onhexgroup

دستورات پشته

نسخه	نوع عملوند	سایز پشته
۳۲ بیتی	رجیستر	در حالت بایتی و word ۲ بایت و در حالت Dword ۴ بایت اضافه میکنن.
	مموری	در حالت بایتی و word ۲ بایت و در حالت Dword ۴ بایت اضافه میکنن.
۶۴ بیتی	رجیستر	حالت بایتی و Dword ممنوع. حالت WORD ۲ بایت حالت QWORD ۸ بایت
	مموری	حالت بایتی و Dword ممنوع. حالت WORD ۲ بایت حالت QWORD ۸ بایت

Github: Onhexgroup

دستورات پشته

POP AX

AX=1

STACK

ESP

2

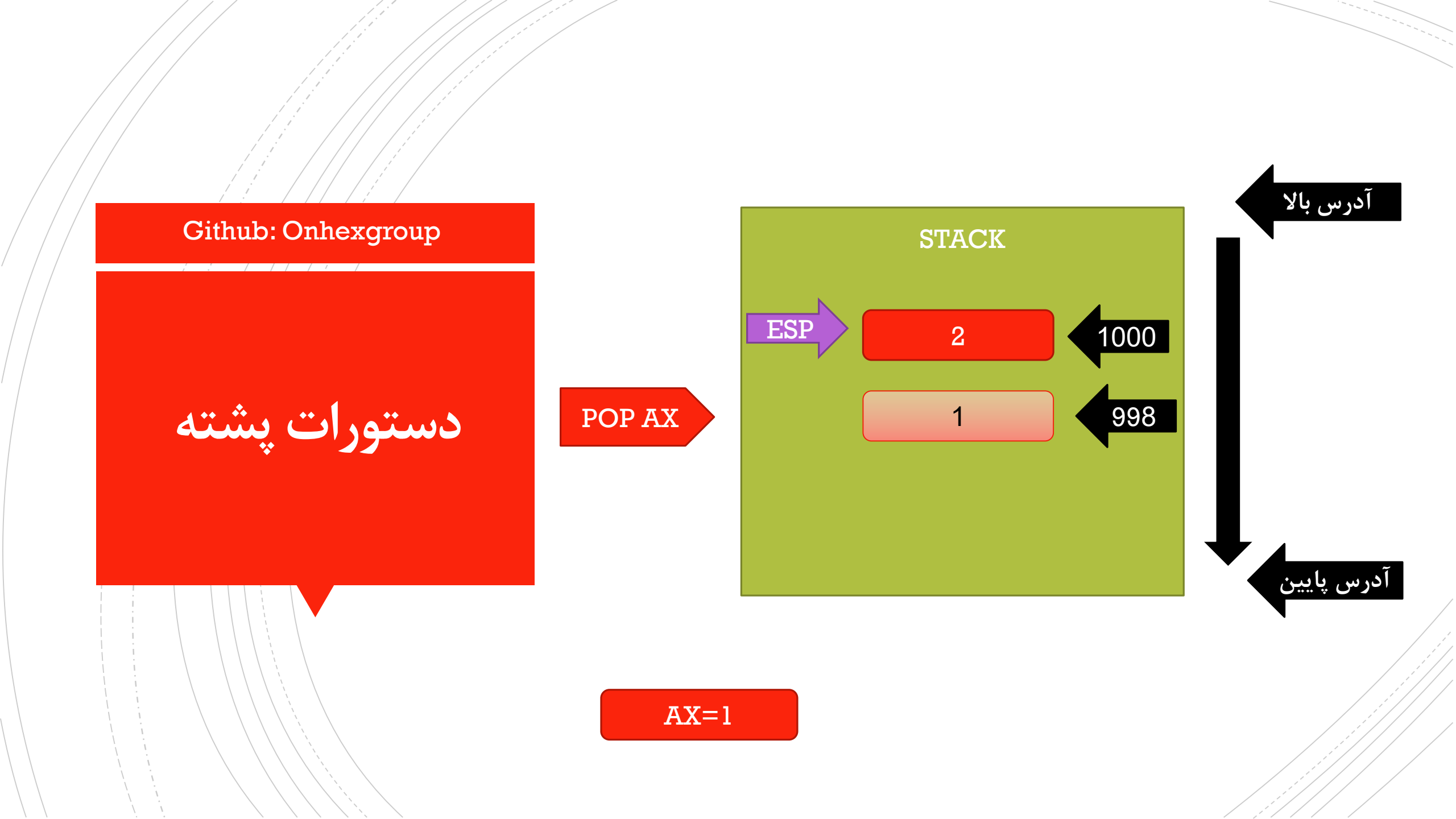
1000

1

998

آدرس بالا

آدرس پایین



Youtube: Onhexgroup

دستورات پشته

POP BX

ESP

STACK

2

1000

1

998

آدرس بالا

آدرس پایین

BX=2

AX=1

■ دستورات ADD و SUB در پشته

Github: Onhexgroup

دستورات پشته

PUSH 1

RSP

STACK_64

2

1000

1

992

$PUSH = RSP - 8$

POP RAX

RSP

STACK_64

2

1000

1

992

$POP = RSP + 8$

■ دستورات **ADD** و **SUB** در پشته

Youtube: Onhexgroup

دستورات پشته

SUB rsp,24

STACK_64

1000

976

PUSH = RSP - 24

ADD rsp,24

STACK_64

1000

976

POP = RSP + 24

Onhexgroup.ir

دستورات پشته

■ دستورات **MOV** و **LEA** در پشته

MOV esp,[eax]

Lea eax,[rsp]

Youtube: Onhexgroup

قاب پشته

- قاب پشته یا فریم پشته یا **Stack frame**، بخشی از پشته که موقع فراخوانی تابع ایجاد میشه.
- این قاب شامل همه ی موارد مورد نیاز برای اجرای یک تابع و برگشت به نقطه قبل است.
- اشاره گر پایه (مرجع فریم): **EBP** یا **RBP**

Twitter: Onhexgroup

EBP/RBP

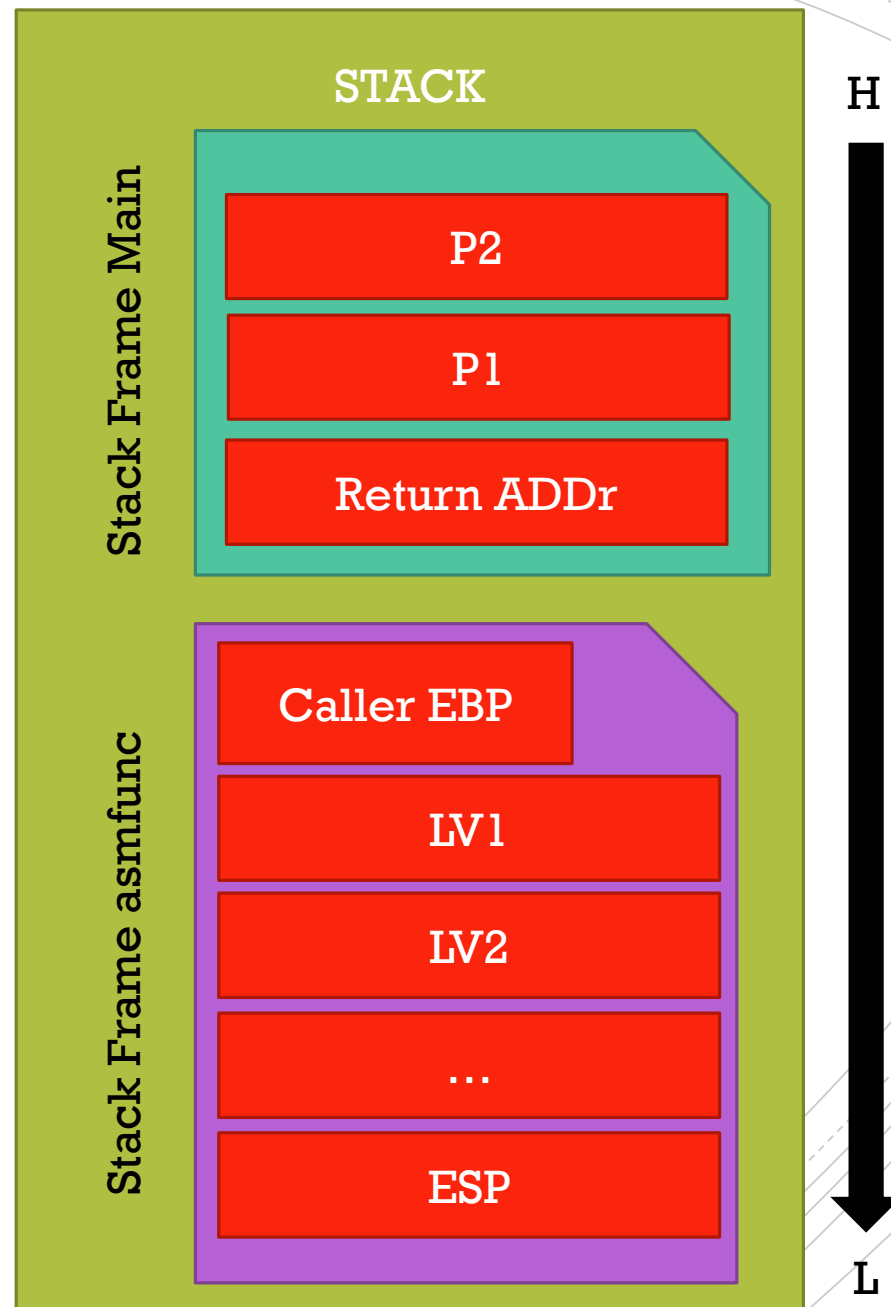
```
Void main(){  
    asmfunc(P1,P2)  
}  
Void asmfunc(P1,P2){  
    LV1;  
    LV2;  
}
```

EBP+N

EBP

EBP-N

ESP



Telegram: onhex_ir

درصد

ONHEXGROUP

87%

NOP 10%

PUSH 15%

CALL 8%

LEA 5%

MOV 27%

INT3 5%

ADD 3%

JNZ 2%

POP 3%

JMP 2%

XOR 2%

XADD 1%

CMP 3%

JG 1%

DEC 1%

JZ 2%

TEST 3%

RET 2%

SUB 2%

OTHRES
5%