

---

# **ОПЕРАЦИОННЫЕ СИСТЕМЫ**

# Несколько слов об архитектуре ЭВМ

---

# Введение в архитектуру ЭВМ

---

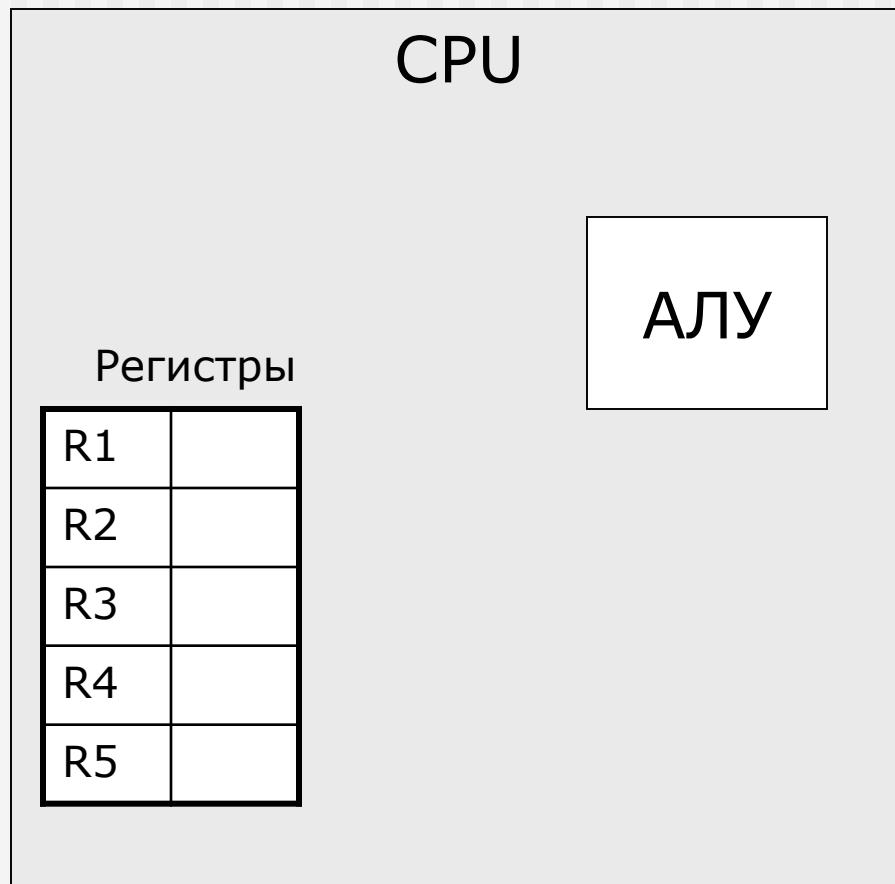
CPU

Регистры

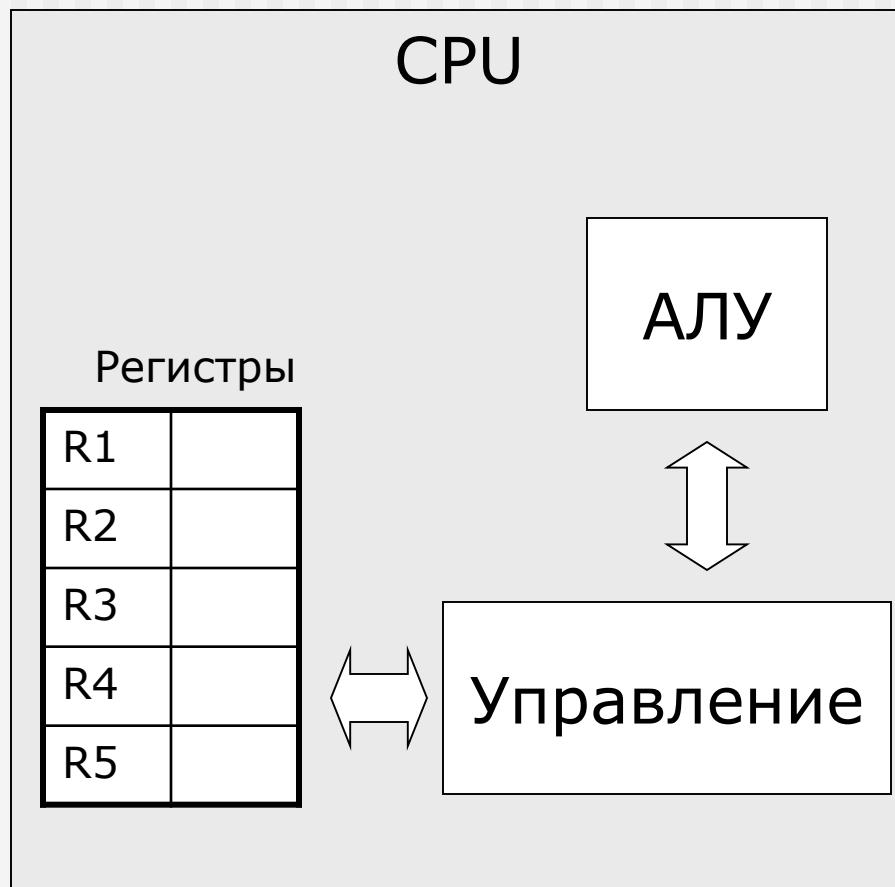
R1	
R2	
R3	
R4	
R5	

# Введение в архитектуру ЭВМ

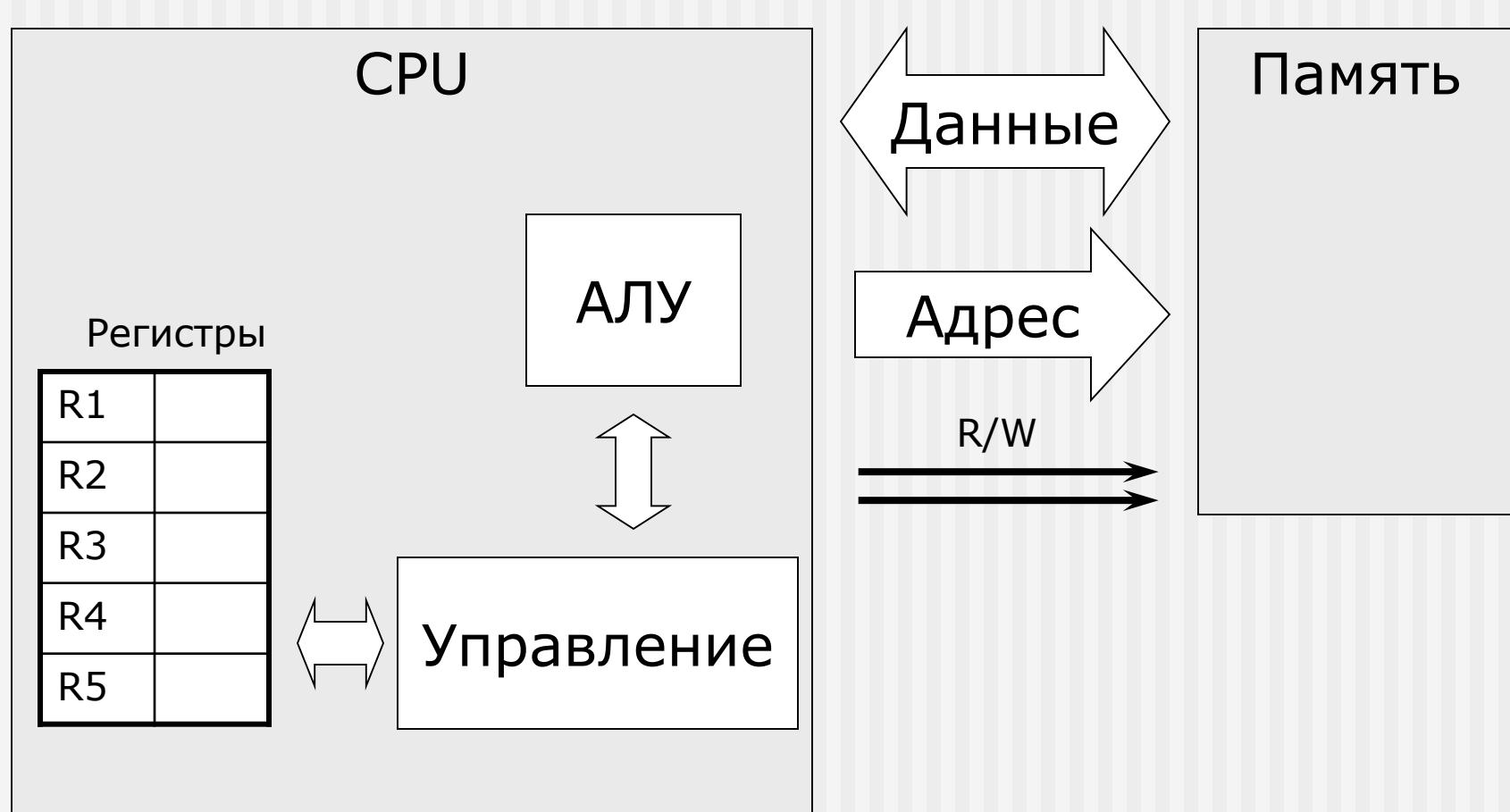
---



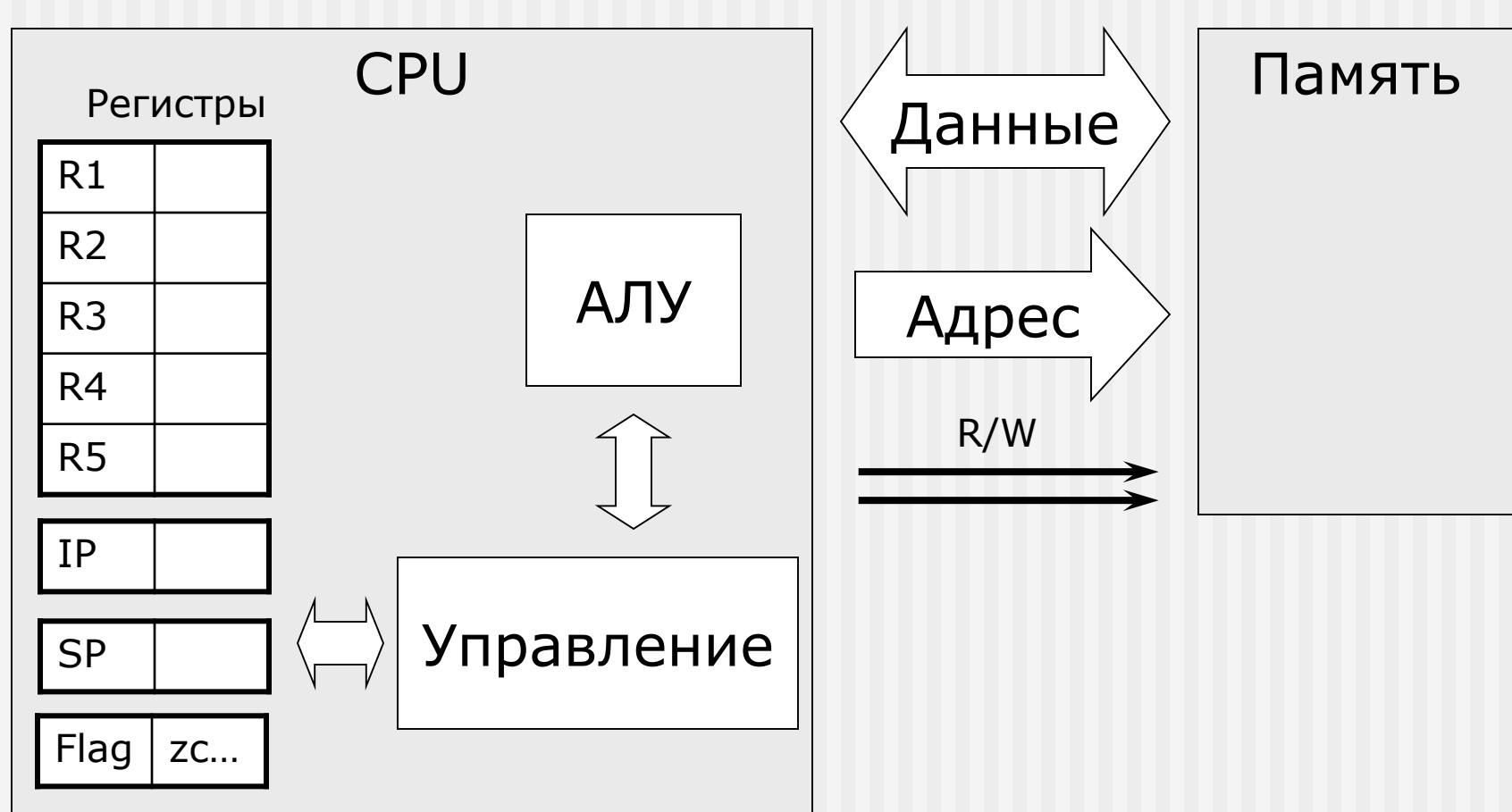
# Введение в архитектуру ЭВМ



# Введение в архитектуру ЭВМ



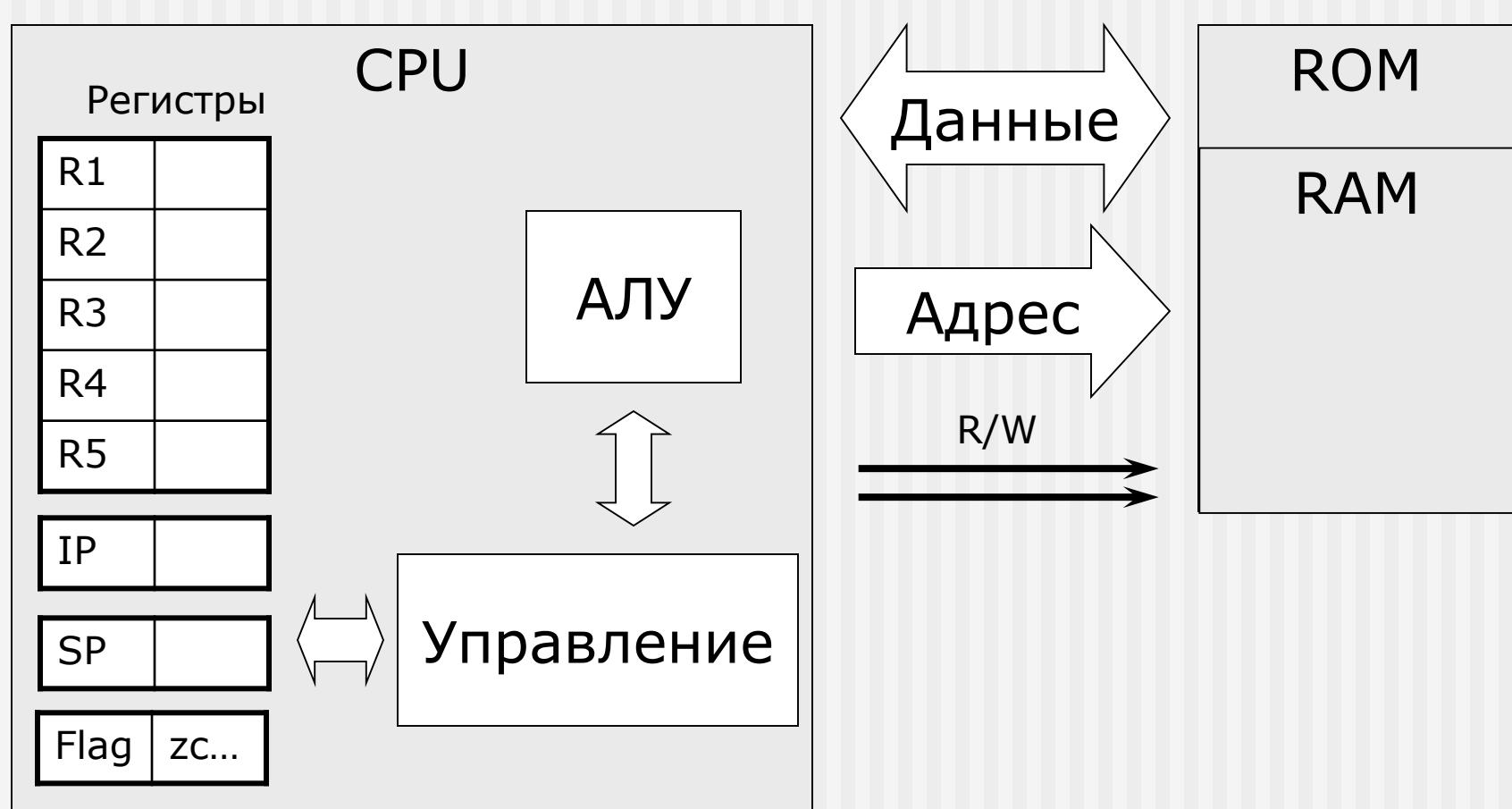
# Введение в архитектуру ЭВМ



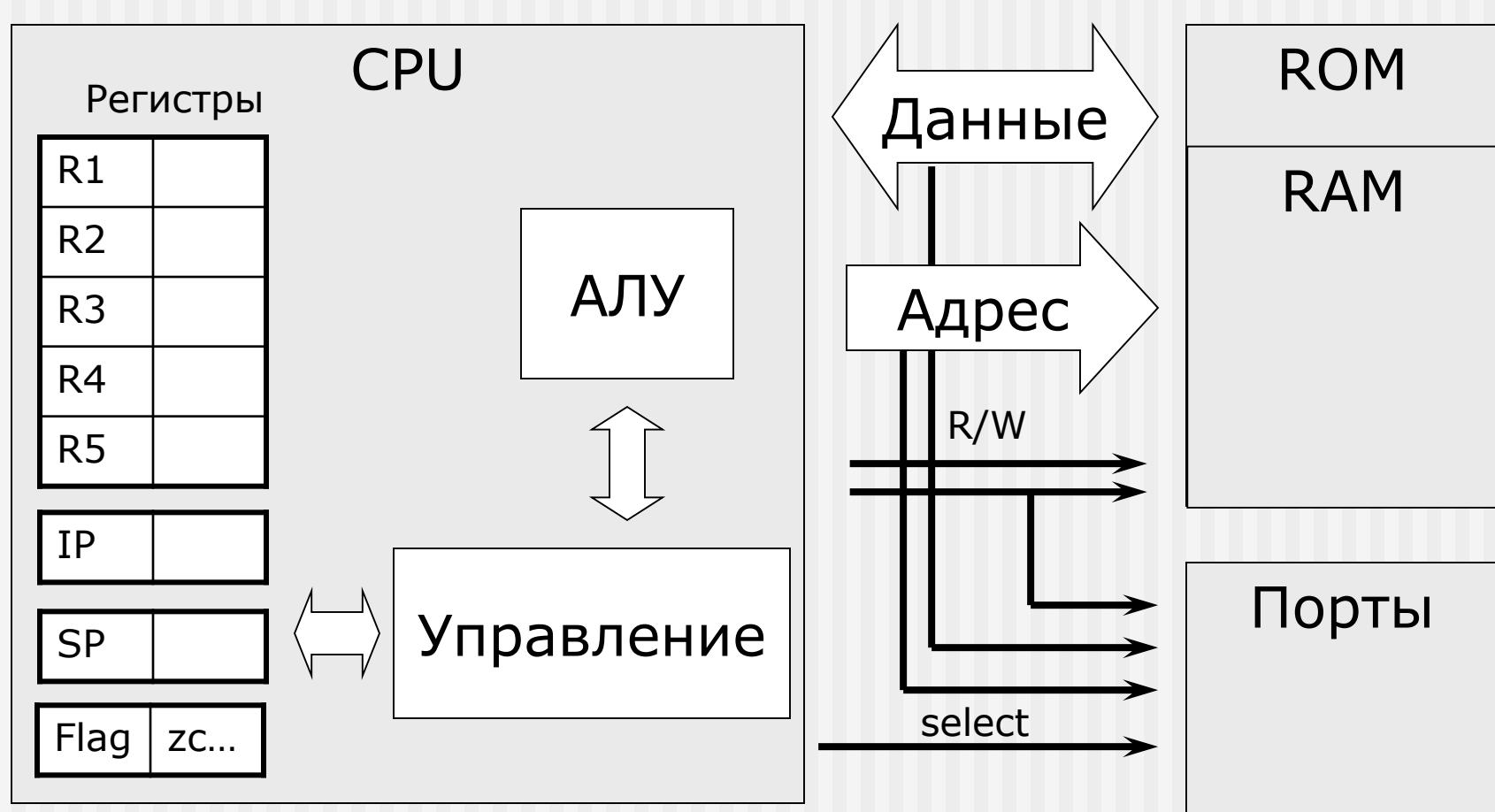
# Регистры современного процессора архитектуры x86

ZMM0	YMM0	XMM0	ZMM1	YMM1	XMM1	ST(0) MM0	ST(1) MM1	AX	EAX	RAX	R8	R8W	R8D	R12	R12W	R12D	CR0	CR4		
ZMM2	YMM2	XMM2	ZMM3	YMM3	XMM3	ST(2) MM2	ST(3) MM3	BX	EBX	RBX	R9	R9W	R9D	R13	R13W	R13D	CR1	CR5		
ZMM4	YMM4	XMM4	ZMM5	YMM5	XMM5	ST(4) MM4	ST(5) MM5	CX	ECX	RCX	R10	R10W	R10D	R14	R14W	R14D	CR2	CR6		
ZMM6	YMM6	XMM6	ZMM7	YMM7	XMM7	ST(6) MM6	ST(7) MM7	DX	EDX	RDX	R11	R11W	R11D	R15	R15W	R15D	CR3	CR7		
ZMM8	YMM8	XMM8	ZMM9	YMM9	XMM9			BP	EBP	RBP	DI	EDI	RDI	SP	ESP	RIP	CR3	CR8		
ZMM10	YMM10	XMM10	ZMM11	YMM11	XMM11	CW	FP_IP	FP_DP	FP_CS	SI	ESI	RSI	SP	ESP	RSP	MSW	CR9			
ZMM12	YMM12	XMM12	ZMM13	YMM13	XMM13	SW											CR10			
ZMM14	YMM14	XMM14	ZMM15	YMM15	XMM15	TW											CR11			
ZMM16	ZMM17	ZMM18	ZMM19	ZMM20	ZMM21	ZMM22	ZMM23	FP_DS									CR12			
ZMM24	ZMM25	ZMM26	ZMM27	ZMM28	ZMM29	ZMM30	ZMM31	FP_OPC	FP_DP	FP_IP	CS	SS	DS	GDTR	IDTR	DR0	DR6	CR13		
											ES	FS	GS	TR	LDTR	DR1	DR7	CR14		
														RFLAGS	EFLAGS	FLAGS	DR2	DR8	CR15	
																	DR3	DR9	MXCSR	
																	DR4	DR10	DR12	DR14
																	DR5	DR11	DR13	DR15

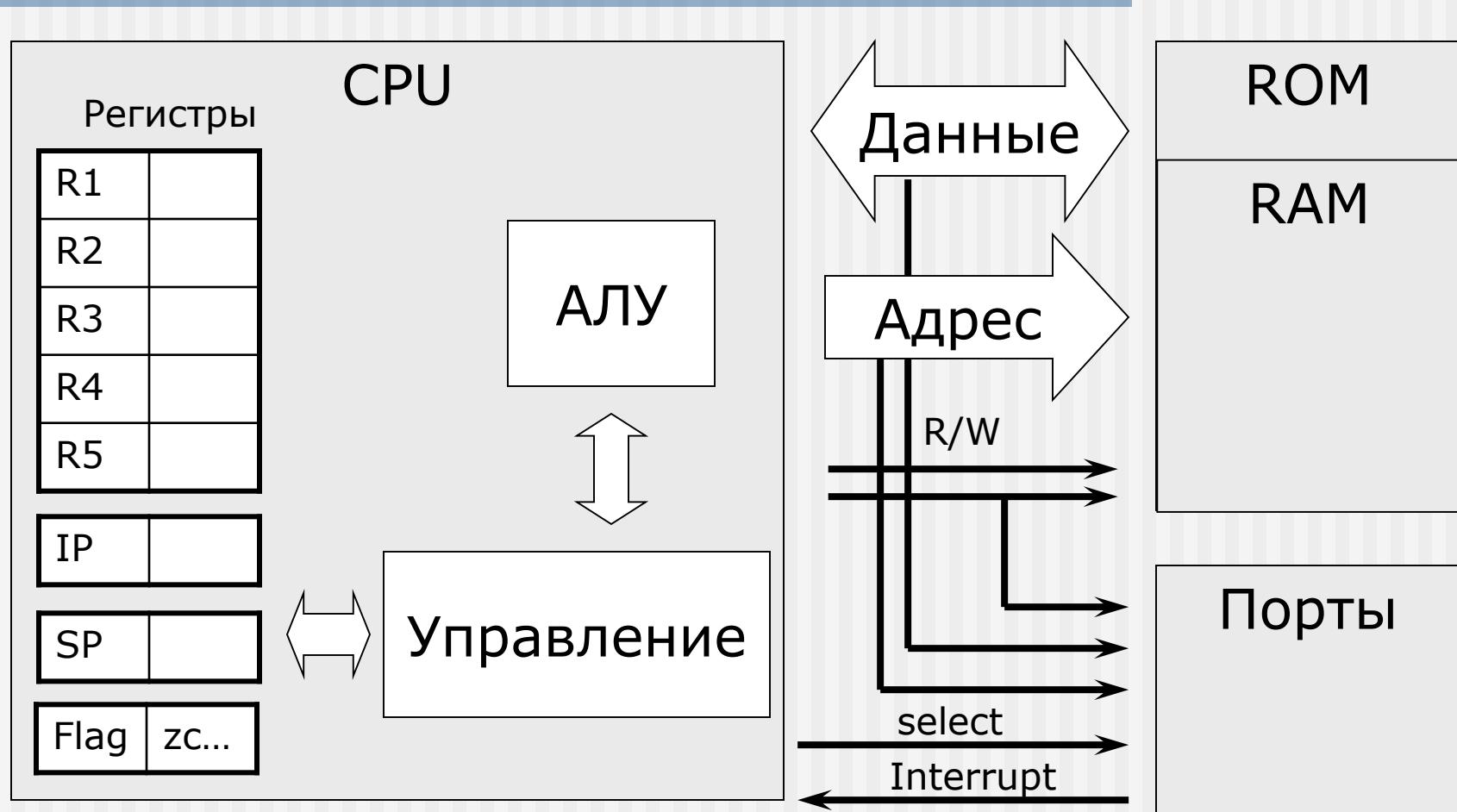
# Введение в архитектуру ЭВМ



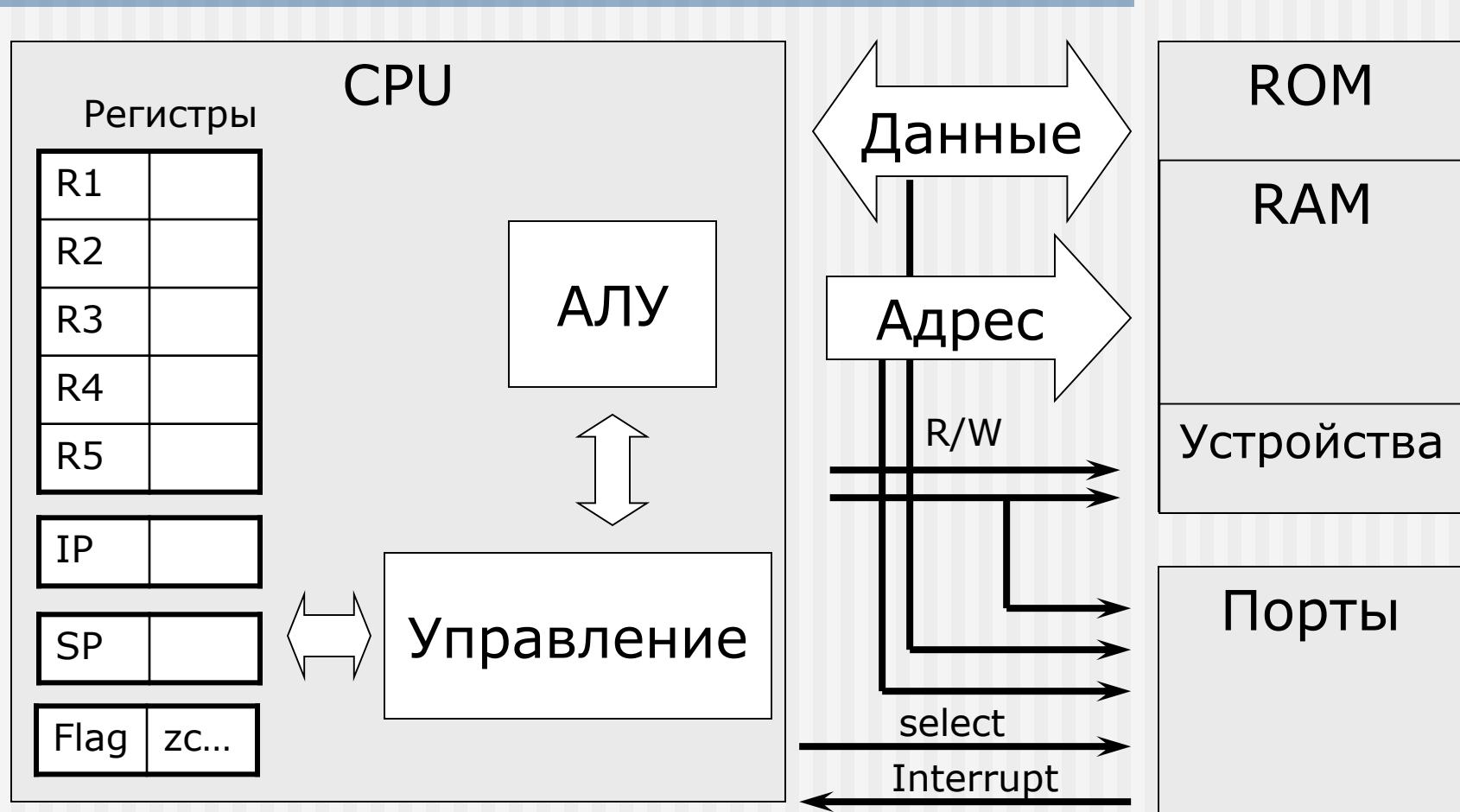
# Введение в архитектуру ЭВМ



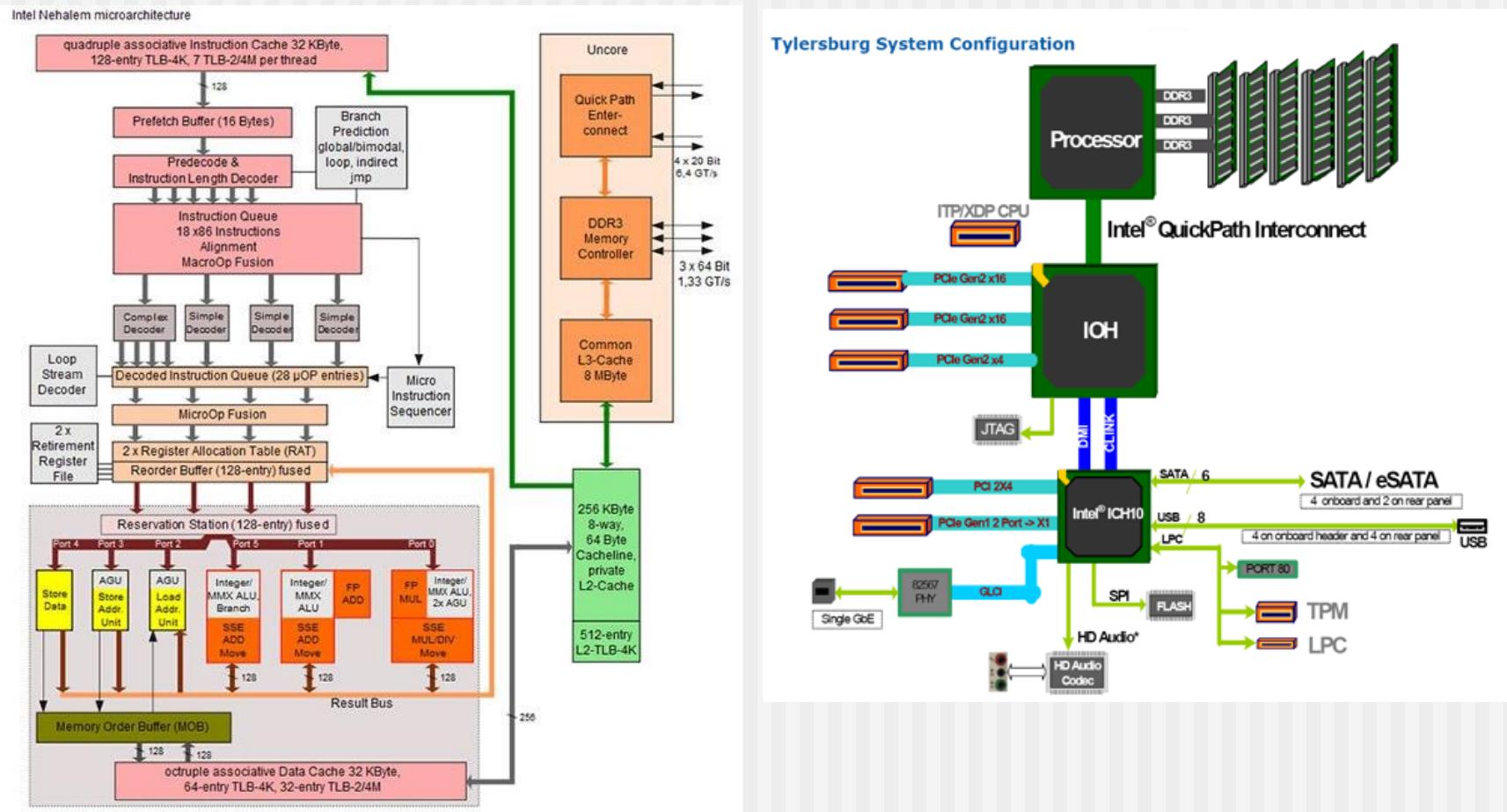
# Введение в архитектуру ЭВМ



# Введение в архитектуру ЭВМ



# Введение в архитектуру ЭВМ



GT/s: gigatransfers per second

---

**СТЕК**

# Пример использования стека при вызове функций

---

```
int factorial(int x)
{
    int a=0;
    if(x<=1)
        return x;
    a=x-1;
    return x*factorial(a);
}
void main(void)
{
    int f=0;
    f=factorial(3);
    print(f);
...
}
```

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	↙
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	↙
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	0
IP	200
SP	672

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
	664	
	666	
	668	
	670	
→	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	↙
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	0
IP	202
SP	670

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
	664	
	666	
	668	
→	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	↙
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	0
IP	204
SP	668

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
	664	
	666	
→	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	↙
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	0
IP	204
SP	666

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	
→	666	ret: 206
	668	x=3
	670	f=0
	672	

# Регистры

<b>R1</b>	0
<b>IP</b>	100
<b>SP</b>	668

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	←←
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	←
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	
→	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	←
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp-3]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	0
IP	100
SP	666

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	
→	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	↙
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	0
IP	102
SP	664

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Регистры

<b>R1</b>	0
<b>IP</b>	102
<b>SP</b>	664

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	↙
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

Кадр  
функции  
factorial()

Кадр  
функции  
main()

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	←
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	3
IP	104
SP	664

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	↙
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	3
IP	106
SP	664

Flag z=0 c=0

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	↙
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	3
IP	112
SP	664

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	↙
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
IP	114
SP	664

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	↙
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
IP	116
SP	664

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	↙
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
IP	118
SP	662

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
→	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	←
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
IP	100
SP	660

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
→	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

## Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	↙
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

## Регистры

R1	2
IP	102
SP	658

## Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	↙
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
IP	102
SP	658

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Кадр функции factorial()

Кадр функции factorial()

Кадр функции main()

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	←
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
IP	104
SP	658

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	↙
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
IP	106
SP	658

Flag z=0 c=0

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	↙
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
IP	112
SP	658

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	↙
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	114
SP	658

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	↙
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	116
SP	658

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	↙
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	118
SP	656

# Стек

SP	Адрес	Данные
	650	
	652	
	654	
→	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	➡
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	100
SP	654

# Стек

SP	Адрес	Данные
	650	
	652	
➡	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	↙
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	102
SP	652

# Стек

SP	Адрес	Данные
	650	
→	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	←
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	104
SP	652

# Стек

SP	Адрес	Данные
	650	
→	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	↙
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	106
SP	652

Flag z=1 c=0

# Стек

SP	Адрес	Данные
	650	
→	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	←
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	108
SP	652

# Стек

SP	Адрес	Данные
	650	
→	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	↙
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	110
SP	654

# Стек

SP	Адрес	Данные
	650	
	652	a=0
→	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	←
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	←
	122: r2←[sp+4]	←
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	120
SP	654

# Стек

SP	Адрес	Данные
	650	
	652	a=0
→	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	←
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	120
SP	656

# Стек

SP	Адрес	Данные
	650	
	652	a=0
→	654	ret:120
→	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	←
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
IP	122
SP	658

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
→	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	↙
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	1
R2	2
IP	124
SP	658

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
→	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	←
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
R2	2
IP	126
SP	658

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
→	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	↙
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
R2	2
IP	128
SP	660

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
→	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	↙
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
R2	2
IP	120
SP	662

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
➡	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	←
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
R2	2
IP	122
SP	664

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
→	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	↙
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	2
R2	3
IP	124
SP	664

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
→	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	←
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	6
R2	3
IP	126
SP	664

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
→	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	↙
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	6
R2	3
IP	128
SP	666

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
➡	664	a=2
➡	666	ret: 206
	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	↙
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	6
R2	3
IP	206
SP	668

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
➡	668	x=3
	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	↙
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

# Регистры

R1	6
R2	3
IP	208
SP	670

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
➡	670	f=0
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	◀
	212: CALL print	
...	214: POP	

# Регистры

R1	6
R2	3
IP	210
SP	670

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
➡	670	f=6
	672	

# Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	↙
...	214: POP	

# Регистры

R1	6
R2	3
IP	212
SP	668

# Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
➡	664	a=2
	666	ret: 206
	668	val=6
	670	f=6
	672	

# Распределение оперативной памяти

