ЛАБОРАТОРНАЯ РАБОТА №2

по курсу «Архитектура вычислительных систем»

по теме:

**Отладка кода**

Выполнили:

Студент 3-го курса

Петров Д.А.

Рукавишников С.?.

Москва 2020

|  |
| --- |
|  |
| #include <iostream>  using namespace std;  int main() {  int x=10;  int y1, y2, y3;  y1 = x \* 5;  y2 = x \* 7;  if (x < 7) {  y3 = 0;  }  else  {  y3 = x;  }  printf\_s("x=%d\ny1=%d\ny2=%d\ny3=%d\n", x, y1, y2, y3);  system("Pause");  return 0;  } |
|  |

|  |
| --- |
|  |
| #include <iostream>  using namespace std;  int main() {  int x=10;  int y1, y2, y3;  y1 = x \* 5;  y2 = x \* 7;  if (x < 7) {  y3 = 0;  }  else  {  y3 = x;  }  cout<<"x="<<x<<" y1="<<y1<<" y2="<<y2<<" 23="<<y3<<" ";  system("Pause");  return 0;  } |
| ;10 : y1 = x \* 5;  0x40147d mov -0x10(%ebp),%edx **//перенос значения х в регистр данных edx**  0x401480 mov %edx,%eax **// перенос х в аккумулятор**  0x401482 shl $0x2,%eax **// битовый сдвиг х на 2 бита = х\*4**  0x401485 add %edx,%eax **// результат сдвига + х = 4х+х = х\*5**  0x401487 mov %eax,-0x14(%ebp)**// перенос результата в у1**  ;11 : y2 = x \* 7;  0x40148a mov -0x10(%ebp),%edx **//перенос значения х в регистр данных edx**  0x40148d mov %edx,%eax **// перенос х в аккумулятор**  0x40148f shl $0x3,%eax **// битовый сдвиг х на 3 бита = х\*8**  0x401492 sub %edx,%eax **// результат сдвига - х = 8х-х = 7х**  0x401494 mov %eax,-0x18(%ebp) **// перенос результата в у2**  ;12 : if (x < 7) {  0x401497 cmpl $0x6,-0x10(%ebp) **//просиходит сравнение(х-6) и установка флага**  0x40149b jg 0x4014a6 <main()+70**>// если х>(jg –jump greater ) то происходит переход к метке ”0x4014a6”**  ;13 : y3 = 0;  0x40149d movl $0x0,-0xc(%ebp) m **// в у3 кладется 0**  0x4014a4 jmp 0x4014ac <main()+76> **// прыжок к метке «0x4014ac» - выход из if.**  ;14 : }  ;15 : else  ;16 : {  ;17 : y3 = x;  0x4014a6 mov -0x10(%ebp),%eax **// значение х помещается в аккумулятор**  0x4014a9 mov %eax,-0xc(%ebp) **// из аккумулятора х помещается в у3**  ;18 : } |

|  |
| --- |
|  |
| 1. **#define**  |  | | --- | | #define bar() \  int var = 10;\  printf("result=: %d\n", var\*7) | | ;27 : bar();  0x40155d movl $0xa,-0x1c(%ebp)  0x401564 mov -0x1c(%ebp),%edx  0x401567 mov %edx,%eax  0x401569 shl $0x3,%eax  0x40156c sub %edx,%eax  0x40156e mov %eax,0x4(%esp)  0x401572 movl $0x405079,(%esp)  0x401579 call 0x403c08 <printf> | | Вывод: препроцессор заменил «bar();» на код, который описан в define. Ассемблерный код принципиально ничем не отличается от кода внутри main(). | |
| 1. **Глобальные переменные:**  |  | | --- | | #include <iostream>  using namespace std;  int x=10;  int y1=0;  int y2=0;  int y3=0;  int main() {  y1 = x \* 5;  y2 = x \* 7;  if (x < 7) {  y3 = 0;  }  else  {  y3 = x;  }  cout<<"x="<<x<<" y1="<<y1<<" y2="<<y2<<" 23="<<y3<<" ";  system("Pause");  return 0;  } | | ***Участки отличающегося кода с глобальными переменными:***  ;12 : int main() {  0x401460 lea 0x4(%esp),%ecx  0x401464 and $0xfffffff0,%esp  0x401467 pushl -0x4(%ecx)  0x40146a push %ebp  0x40146b mov %esp,%ebp  0x40146d push %edi  0x40146e push %esi  0x40146f push %ebx  0x401470 push %ecx  0x401471 sub $0x28,%esp  0x401474 call 0x401b30 <\_\_main>  ;15 : y1 = x \* 5;  0x401479 mov 0x404004,%edx  0x40147f mov %edx,%eax  0x401481 shl $0x2,%eax  0x401484 add %edx,%eax  0x401486 mov %eax,0x407020  ***Участки отличающегося кода без глобальных переменных:***  ;8 : int main() {  0x401460 lea 0x4(%esp),%ecx  0x401464 and $0xfffffff0,%esp  0x401467 pushl -0x4(%ecx)  0x40146a push %ebp  0x40146b mov %esp,%ebp  0x40146d push %ecx  0x40146e sub $0x24,%esp  0x401471 call 0x401b30 <\_\_main>  ;15 : y1 = x \* 5;  0x401492 mov -0x10(%ebp),%edx  0x401495 mov %edx,%eax  0x401497 shl $0x2,%eax  0x40149a add %edx,%eax  0x40149c mov %eax,-0x14(%ebp) | | ***Выводы***  В блоке main() добавились push в edi ,esi и ebx.  Далее при непосредственном вычислении значений функций Y – используются конкретные адреса(?)(0x404004), а не место в регистре(-0x10(%ebp)). | |
| 1. **Char**  |  | | --- | | #include <iostream>  #include <string.h>  using namespace std;  int main() {  char x[100]="10";  char y1[100]="hello";  char y2[100]="darkness";  char y3[100]="my old friend";  strncat(y1,x,100);  strncat(y2,x,100);  if (strlen(x) < 7) {  strncat(y3,x,100);  }  else  {  strncat(y3,"empty",5);  }  cout<<"x="<<x<<" y1="<<y1<<" y2="<<y2<<" 23="<<y3<<" ";  system("Pause");  return 0;  } | | ;8 : int main() {  0x401460 push %ebp  0x401461 mov %esp,%ebp  0x401463 push %edi  0x401464 and $0xfffffff0,%esp  0x401467 sub $0x1a0,%esp  0x40146d call 0x401c20 <\_\_main>  ;10 : char x[100]="10";  0x401472 movl $0x3031,0x13c(%esp)  0x40147d lea 0x140(%esp),%edx  0x401484 mov $0x0,%eax  0x401489 mov $0x18,%ecx  0x40148e mov %edx,%edi  0x401490 rep stos %eax,%es:(%edi)  ;11 : char y1[100]="hello";  0x401492 movl $0x6c6c6568,0xd8(%esp)  0x40149d movl $0x6f,0xdc(%esp)  0x4014a8 lea 0xe0(%esp),%edx  0x4014af mov $0x0,%eax  0x4014b4 mov $0x17,%ecx  0x4014b9 mov %edx,%edi  0x4014bb rep stos %eax,%es:(%edi)  ;12 : char y2[100]="darkness";  0x4014bd movl $0x6b726164,0x74(%esp)  0x4014c5 movl $0x7373656e,0x78(%esp)  0x4014cd movl $0x0,0x7c(%esp)  0x4014d5 lea 0x80(%esp),%edx  0x4014dc mov $0x0,%eax  0x4014e1 mov $0x16,%ecx  0x4014e6 mov %edx,%edi  0x4014e8 rep stos %eax,%es:(%edi)  ;13 : char y3[100]="my old friend";  0x4014ea movl $0x6f20796d,0x10(%esp)  0x4014f2 movl $0x6620646c,0x14(%esp)  0x4014fa movl $0x6e656972,0x18(%esp)  0x401502 movl $0x64,0x1c(%esp)  0x40150a lea 0x20(%esp),%edx  0x40150e mov $0x0,%eax  0x401513 mov $0x15,%ecx  0x401518 mov %edx,%edi  0x40151a rep stos %eax,%es:(%edi)  ;15 : strncat(y1,x,100);  0x40151c movl $0x64,0x8(%esp)  0x401524 lea 0x13c(%esp),%eax  0x40152b mov %eax,0x4(%esp)  0x40152f lea 0xd8(%esp),%eax  0x401536 mov %eax,(%esp)  0x401539 call 0x403cb0 <strncat>  ;16 : strncat(y2,x,100);  0x40153e movl $0x64,0x8(%esp)  0x401546 lea 0x13c(%esp),%eax  0x40154d mov %eax,0x4(%esp)  0x401551 lea 0x74(%esp),%eax  0x401555 mov %eax,(%esp)  0x401558 call 0x403cb0 <strncat>  ;17 : if (strlen(x) < 7) {  0x40155d lea 0x13c(%esp),%eax  0x401564 mov %eax,(%esp)  0x401567 call 0x403cb8 <strlen>  0x40156c cmp $0x6,%eax  0x40156f ja 0x401592 <main()+306>  ;18 : strncat(y3,x,100);  0x401571 movl $0x64,0x8(%esp)  0x401579 lea 0x13c(%esp),%eax  0x401580 mov %eax,0x4(%esp)  0x401584 lea 0x10(%esp),%eax  0x401588 mov %eax,(%esp)  0x40158b call 0x403cb0 <strncat>  0x401590 jmp 0x4015bf <main()+351>  ;19 : }  ;20 : else  ;21 : {  ;22 : strncat(y3,"empty",5);  0x401592 lea 0x10(%esp),%eax  0x401596 mov $0xffffffff,%ecx  0x40159b mov %eax,%edx  0x40159d mov $0x0,%eax  0x4015a2 mov %edx,%edi  0x4015a4 repnz scas %es:(%edi),%al  0x4015a6 mov %ecx,%eax  0x4015a8 not %eax  0x4015aa lea -0x1(%eax),%edx  0x4015ad lea 0x10(%esp),%eax  0x4015b1 add %edx,%eax  0x4015b3 movl $0x74706d65,(%eax)  0x4015b9 movw $0x79,0x4(%eax)  ;23 : }  ;24 : cout<<"x="<<x<<" y1="<<y1<<" y2="<<y2<<" 23="<<y3<<" ";  **\*\*\*\*\*\*\*\*\*\*\*\*\*ПРОПУЩЕНО МНОГО КОДА ДЛЯ COUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  ;27 : system("Pause");  0x401661 movl $0x405079,(%esp)  0x401668 call 0x403ca8 <system>  ;28 : return 0;  0x40166d mov $0x0,%eax  ;29 : }  0x401672 mov -0x4(%ebp),%edi  0x401675 leave  0x401676 ret | |
| 1. **Short**  |  | | --- | | ;6 : int main() {  0x401460 lea 0x4(%esp),%ecx  0x401464 and $0xfffffff0,%esp  0x401467 pushl -0x4(%ecx)  0x40146a push %ebp  0x40146b mov %esp,%ebp  0x40146d push %edi  0x40146e push %esi  0x40146f push %ebx  0x401470 push %ecx  0x401471 sub $0x38,%esp  0x401474 call 0x401b30 <\_\_main>  ;8 : short x=10;  0x401479 movw $0xa,-0x1c(%ebp) **// mov размером 1 слово – 2 байта (тип short – 2 байта)**  ;9 : short y1=0;  0x40147f movw $0x0,-0x1e(%ebp)  ;10 : short y2=0;  0x401485 movw $0x0,-0x20(%ebp)  ;11 : short y3=0;  0x40148b movw $0x0,-0x1a(%ebp)  ;13 : y1 = x \* 5;  0x401491 movzwl -0x1c(%ebp),%edx **//Перемещение слова (short)с нулевым расширением в long**  0x401495 mov %edx,%eax  0x401497 shl $0x2,%eax  0x40149a add %edx,%eax  0x40149c mov %ax,-0x1e(%ebp) **//перенос из регистра ax(малые разряды eax) в у1**  ;14 : y2 = x \* 7;  0x4014a0 movzwl -0x1c(%ebp),%edx  0x4014a4 mov %edx,%eax  0x4014a6 shl $0x3,%eax  0x4014a9 sub %edx,%eax  0x4014ab mov %ax,-0x20(%ebp)  ;15 : if (x < 7) {  0x4014af cmpw $0x6,-0x1c(%ebp)  0x4014b4 jg 0x4014be <main()+94>  ;16 : y3 = 0;  0x4014b6 movw $0x0,-0x1a(%ebp)  0x4014bc jmp 0x4014c6 <main()+102>  ;17 : }  ;18 : else  ;19 : {  ;20 : y3 = x;  0x4014be movzwl -0x1c(%ebp),%eax  0x4014c2 mov %ax,-0x1a(%ebp)  ;21 : }  ;22 : cout<<"x="<<x<<" y1="<<y1<<" y2="<<y2<<" 23="<<y3<<" ";  **\*\*\*\*\*\*\*\*\*\*\*\*\*ПРОПУЩЕНО МНОГО КОДА ДЛЯ COUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  ;26 : system("Pause");  0x401564 movl $0x405079,(%esp)  0x40156b call 0x403bb8 <system>  ;27 : return 0;  0x401570 mov $0x0,%eax  ;28 : }  0x401575 lea -0x10(%ebp),%esp  0x401578 pop %ecx  0x401579 pop %ebx  0x40157a pop %esi  0x40157b pop %edi  0x40157c pop %ebp  0x40157d lea -0x4(%ecx),%esp  0x401580 ret | | **Вывод:**  Работа с типом short имеет изменения в операциях, некоторые из них имеют суффикс “w”(mov**w**), который свидетельствует о том, что операция производится с операндами длиной в 2 байта. | |
| 1. **Long**  |  | | --- | | ;6 : int main() {  0x401460 lea 0x4(%esp),%ecx  0x401464 and $0xfffffff0,%esp  0x401467 pushl -0x4(%ecx)  0x40146a push %ebp  0x40146b mov %esp,%ebp  0x40146d push %ecx  0x40146e sub $0x24,%esp  0x401471 call 0x401b30 <\_\_main>  ;8 : long x=10;  0x401476 movl $0xa,-0x10(%ebp)  ;9 : long y1=0;  0x40147d movl $0x0,-0x14(%ebp)  ;10 : long y2=0;  0x401484 movl $0x0,-0x18(%ebp)  ;11 : long y3=0;  0x40148b movl $0x0,-0xc(%ebp)  ;13 : y1 = x \* 5;  0x401492 mov -0x10(%ebp),%edx  0x401495 mov %edx,%eax  0x401497 shl $0x2,%eax  0x40149a add %edx,%eax  0x40149c mov %eax,-0x14(%ebp)  ;14 : y2 = x \* 7;  0x40149f mov -0x10(%ebp),%edx  0x4014a2 mov %edx,%eax  0x4014a4 shl $0x3,%eax  0x4014a7 sub %edx,%eax  0x4014a9 mov %eax,-0x18(%ebp)  ;15 : if (x < 7) {  0x4014ac cmpl $0x6,-0x10(%ebp)  0x4014b0 jg 0x4014bb <main()+91>  ;16 : y3 = 0;  0x4014b2 movl $0x0,-0xc(%ebp)  0x4014b9 jmp 0x4014c1 <main()+97>  ;17 : }  ;18 : else  ;19 : {  ;20 : y3 = x;  0x4014bb mov -0x10(%ebp),%eax  0x4014be mov %eax,-0xc(%ebp)  ;21 : }  ;22 : cout<<"x="<<x<<" y1="<<y1<<" y2="<<y2<<" 23="<<y3<<" ";  **\*\*\*\*\*\*\*\*\*\*\*\*\*ПРОПУЩЕНО МНОГО КОДА ДЛЯ COUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  ;26 : system("Pause");  0x40155d movl $0x405079,(%esp)  0x401564 call 0x403bb8 <system>  ;27 : return 0;  0x401569 mov $0x0,%eax  ;28 : }  0x40156e mov -0x4(%ebp),%ecx  0x401571 leave  0x401572 lea -0x4(%ecx),%esp  0x401575 ret | | **Вывод:**  Работа с типом long имеет изменения в операциях, некоторые из них имеют суффикс “l”(mov**l**), который свидетельствует о том, что операция производится с операндами длиной в 4 байта. | |
| 1. **Long long**  |  | | --- | | ;5 : int main() {  0x401460 push %ebp  0x401461 mov %esp,%ebp  0x401463 and $0xfffffff0,%esp  0x401466 sub $0x30,%esp  0x401469 call 0x401af0 <\_\_main>  ;6 : long long x=10;  0x40146e movl $0xa,0x28(%esp)  0x401476 movl $0x0,0x2c(%esp)  ;7 : long long y1=0;  0x40147e movl $0x0,0x20(%esp)  0x401486 movl $0x0,0x24(%esp)  ;8 : long long y2=0;  0x40148e movl $0x0,0x18(%esp)  0x401496 movl $0x0,0x1c(%esp)  ;9 : long long y3=0;  0x40149e movl $0x0,0x10(%esp)  0x4014a6 movl $0x0,0x14(%esp)  ;11 : y1 = x \* 5;  0x4014ae mov 0x2c(%esp),%eax  0x4014b2 imul $0x5,%eax,%edx  0x4014b5 mov 0x28(%esp),%eax  0x4014b9 imul $0x0,%eax,%eax  0x4014bc lea (%edx,%eax,1),%ecx  0x4014bf mov $0x5,%eax  0x4014c4 mull 0x28(%esp)  0x4014c8 add %edx,%ecx  0x4014ca mov %ecx,%edx  0x4014cc mov %eax,0x20(%esp)  0x4014d0 mov %edx,0x24(%esp)  0x4014d4 mov %eax,0x20(%esp)  0x4014d8 mov %edx,0x24(%esp)  ;12 : y2 = x \* 7;  0x4014dc mov 0x2c(%esp),%eax  0x4014e0 imul $0x7,%eax,%edx  0x4014e3 mov 0x28(%esp),%eax  0x4014e7 imul $0x0,%eax,%eax  0x4014ea lea (%edx,%eax,1),%ecx  0x4014ed mov $0x7,%eax  0x4014f2 mull 0x28(%esp)  0x4014f6 add %edx,%ecx  0x4014f8 mov %ecx,%edx  0x4014fa mov %eax,0x18(%esp)  0x4014fe mov %edx,0x1c(%esp)  0x401502 mov %eax,0x18(%esp)  0x401506 mov %edx,0x1c(%esp)  ;13 : if (x < 7) {  0x40150a cmpl $0x0,0x2c(%esp)  0x40150f jg 0x401531 <main()+209>  0x401511 cmpl $0x0,0x2c(%esp)  0x401516 js 0x40151f <main()+191>  0x401518 cmpl $0x6,0x28(%esp)  0x40151d ja 0x401531 <main()+209>  ;14 : y3 = 0;  0x40151f movl $0x0,0x10(%esp)  0x401527 movl $0x0,0x14(%esp)  0x40152f jmp 0x401541 <main()+225>  ;15 : }  ;16 : else  ;17 : {  ;18 : y3 = x;  0x401531 mov 0x28(%esp),%eax  0x401535 mov 0x2c(%esp),%edx  0x401539 mov %eax,0x10(%esp)  0x40153d mov %edx,0x14(%esp)  ;19 : }  ;20 : //cout<<"x="<<x<<" y1="<<y1<<" y2="<<y2<<" 23="<<y3<<" ";  ;22 : system("Pause");  0x401541 movl $0x405065,(%esp)  0x401548 call 0x403b78 <system>  ;23 : return 0;  0x40154d mov $0x0,%eax  ;24 : }  0x401552 leave  0x401553 ret | | **Вывод:**  Работа с типом long long имеет изменения в операциях, некоторые из них имеют суффикс “l”(mov**l**), который свидетельствует о том, что операция производится с операндами длиной в 4 байта. Также, ввиду того, что данный тип имеет размер 8 байт, работа с переменными данного типа получается как работа с двумя переменными по 4 байта.  7 : long long y1=0;  0x40147e movl $0x0,0x20(%esp)  0x401486 movl $0x0,0x24(%esp)  При работе с типом long, было бы достаточно только одной такой строчки, т.к. long – 4 байта.  Таким образом, все операции с одной переменной long long, выглядят, как работа с двумя переменными long. | |
| 1. **Long double**  |  | | --- | | ;5 : int main() {  0x401460 push %ebp  0x401461 mov %esp,%ebp  0x401463 and $0xfffffff0,%esp  0x401466 sub $0x40,%esp  0x401469 call 0x401a80 <\_\_main>  ;6 : long double x=10;  0x40146e fldt 0x405080  0x401474 fstpt 0x34(%esp)  ;7 : long double y1=0;  0x401478 fldz  0x40147a fstpt 0x28(%esp)  ;8 : long double y2=0;  0x40147e fldz  0x401480 fstpt 0x1c(%esp)  ;9 : long double y3=0;  0x401484 fldz  0x401486 fstpt 0x10(%esp)  ;11 : y1 = x \* 5;  0x40148a fldt 0x34(%esp)  0x40148e fldt 0x405090  0x401494 fmulp %st,%st(1)  0x401496 fstpt 0x28(%esp)  ;12 : y2 = x \* 7;  0x40149a fldt 0x34(%esp)  0x40149e fldt 0x4050a0  0x4014a4 fmulp %st,%st(1)  0x4014a6 fstpt 0x1c(%esp)  ;13 : if (x < 7) {  0x4014aa fldt 0x4050a0  0x4014b0 fldt 0x34(%esp)  0x4014b4 fxch %st(1)  0x4014b6 fucompp  0x4014b8 fnstsw %ax  0x4014ba sahf  0x4014bb jbe 0x4014c5 <main()+101>  ;14 : y3 = 0;  0x4014bd fldz  0x4014bf fstpt 0x10(%esp)  0x4014c3 jmp 0x4014cd <main()+109>  ;15 : }  ;16 : else  ;17 : {  ;18 : y3 = x;  0x4014c5 fldt 0x34(%esp)  0x4014c9 fstpt 0x10(%esp)  ;19 : }  ;20 : //cout<<"x="<<x<<" y1="<<y1<<" y2="<<y2<<" 23="<<y3<<" ";  ;22 : system("Pause");  0x4014cd movl $0x405071,(%esp)  0x4014d4 call 0x403b08 <system>  ;23 : return 0;  0x4014d9 mov $0x0,%eax  ;24 : }  0x4014de leave  0x4014df ret | | **Вывод:**  В программе с типом long double используются комманды для работы с вещественными значениями(fldt, fmulp, fstpt).  Принцип работы похож на работу с вещественными переменными, разве что размер переменных – 6 байт. | |

|  |
| --- |
|  |
| #include <iostream>  using namespace std;  int func(int x){  int y1=0;  y1 = x \* 5;  return y1;  }  int main() {  int x=10;  int y1=0;  y1=func(x);  system("Pause");  return 0;  } |
| ;12 : int main() {  0x40147f push %ebp  0x401480 mov %esp,%ebp  0x401482 and $0xfffffff0,%esp  0x401485 sub $0x20,%esp  0x401488 call 0x401a60 <\_\_main>  ;14 : int x=10;  0x40148d movl $0xa,0x1c(%esp)  ;15 : int y1=0;  0x401495 movl $0x0,0x18(%esp)  ;30 : y1=func(x);  0x40149d mov 0x1c(%esp),%eax **//значение х перемещается в регистор(eax(«аккумулятор»))**  0x4014a1 mov %eax,(%esp) **// значение передается в регистр(esp)**  0x4014a4 call 0x401460 <func(int)> **//вызов функции**  0x4014a9 mov %eax,0x18(%esp) **//перемещение результата функции из регистра eax в переменную у1.**  ;32 : system("Pause");  0x4014ad movl $0x405065,(%esp)  0x4014b4 call 0x403ae8 <system>  ;33 : return 0;  0x4014b9 mov $0x0,%eax  ;34 : }  0x4014be leave  0x4014bf ret |
| Код функции:  ;5 : int func(int x){  0x401460 push %ebp  0x401461 mov %esp,%ebp  0x401463 sub $0x10,%esp  ;6 : int y1=1;  0x401466 movl $0x1,-0x4(%ebp)  ;7 : y1+=x;  0x40146d mov 0x8(%ebp),%eax  0x401470 add %eax,-0x4(%ebp) **// у1=у1+х**  ;8 : return y1;  0x401473 mov -0x4(%ebp),%eax **// вывод результата в регистр(eax «аккумулятор»)**  ;10 : }  0x401476 leave  0x401477 ret |

|  |
| --- |
|  |
| #include <iostream>  using namespace std;  float func(float x){  float y1=0;  y1 = x \* 5;  return y1;  }  int main() {  float x=10;  float y1=0;  y1=func(x);  system("Pause");  return 0;  } |
| ;12 : int main() {  0x40147e push %ebp  0x40147f mov %esp,%ebp  0x401481 and $0xfffffff0,%esp  0x401484 sub $0x30,%esp  0x401487 call 0x401a70 <\_\_main>  ;14 : float x=10;  0x40148c flds 0x405070 **//запись в стек сопроцессора значения, заданного адресом памяти**  0x401492 fstps 0x2c(%esp) **//сохранение вещ знач с извлечением из стека**  ;15 : float y1=0;  0x401496 fldz **// запись в стек константы +0.0**  0x401498 fstps 0x28(%esp) **//сохранение константы с извлечением из стека**  ;30 : y1=func(x);  0x40149c flds 0x2c(%esp) **//запись в стек сопроцессора значения х**  0x4014a0 fstps (%esp) **//сохранение вещ знач с извлечением из стека**  0x4014a3 call 0x401460 <func(float)> // вызов функции  0x4014a8 fstps 0x1c(%esp) **//сохранение результата**  0x4014ac mov 0x1c(%esp),%eax **//перенос результата из стека в «аккумулятор»**  0x4014b0 mov %eax,0x28(%esp) **//перенос результата в у1**  ;32 : system("Pause");  0x4014b4 movl $0x405065,(%esp)  0x4014bb call 0x403af8 <system>  ;33 : return 0;  0x4014c0 mov $0x0,%eax  ;34 : }  0x4014c5 leave  0x4014c6 ret |

|  |
| --- |
|  |
| #include <iostream>  using namespace std;  int func(int x){  static int y1=1;  y1+=x;  return y1;  }  int main() {  int x=10;  int y1=0;  y1=func(x);  y1=func(x);  cout<<y1;  system("Pause");  return 0;  } |
| Код функции:  ;5 : int func(int x){  0x401460 push %ebp  0x401461 mov %esp,%ebp  ;6 : static int y1=1;  ;7 : y1+=x;  0x401463 mov 0x404004,%edx **//y1 имеет конкретный адрес в памяти**  0x401469 mov 0x8(%ebp),%eax  0x40146c add %edx,%eax **// у1+х**  0x40146e mov %eax,0x404004 **//перенос результата обратно в у1 по адресу**  ;8 : return y1;  0x401473 mov 0x404004,%eax **//возвращение результата у1 через регистр eax**  ;10 : }  0x401478 pop %ebp  0x401479 ret |
| Код программы:  ;12 : int main() {  0x40147a lea 0x4(%esp),%ecx  0x40147e and $0xfffffff0,%esp  0x401481 pushl -0x4(%ecx)  0x401484 push %ebp  0x401485 mov %esp,%ebp  0x401487 push %ecx  0x401488 sub $0x24,%esp  0x40148b call 0x401a90 <\_\_main>  ;14 : int x=10;  0x401490 movl $0xa,-0xc(%ebp)  ;15 : int y1=0;  0x401497 movl $0x0,-0x10(%ebp)  ;30 : y1=func(x);  0x40149e mov -0xc(%ebp),%eax  0x4014a1 mov %eax,(%esp)  0x4014a4 call 0x401460 <func(int)>  0x4014a9 mov %eax,-0x10(%ebp)  ;31 : y1=func(x);  0x4014ac mov -0xc(%ebp),%eax  0x4014af mov %eax,(%esp)  0x4014b2 call 0x401460 <func(int)>  0x4014b7 mov %eax,-0x10(%ebp)  ;32 : cout<<y1;  0x4014ba mov -0x10(%ebp),%eax  0x4014bd mov %eax,(%esp)  0x4014c0 mov $0x6ff01a40,%ecx  0x4014c5 call 0x401554 <std::ostream::operator<<(int)>  0x4014ca sub $0x4,%esp  ;35 : system(“Pause”);  0x4014cd movl $0x405065,(%esp)  0x4014d4 call 0x403b18 <system>  ;36 : return 0;  0x4014d9 mov $0x0,%eax  ;37 : }  0x4014de mov -0x4(%ebp),%ecx  0x4014e1 leave  0x4014e2 lea -0x4(%ecx),%esp  0x4014e5 ret |
| **Вывод:**  Отличия в коде наблюдаются только на уровне кода самой функции, при работе с у1 происходит обращение не к регистру, а к адресу памяти. |

|  |
| --- |
|  |
|  |
|  |