A0B36APO - Computer architectures
Homework 4

LS 2015/2016
Katedra počítačů, FEL, ČVUT v Praze

Homework deadline: your lecture in week starting May 9.
Use only papers given to you, do not add any more papers.

To answer the questions, use a memory program listing on the following sheet(s).

Program was compiled using C compiler for **x86 32-bit** computer and OS GNU/Linux first. Then the same code has been compiled for little-endian variant of MIPS32 architecture.

Program consists of a function called **toplevel_fnc**, which calls a function **subroutine_fnc**. Also there are system calls.

1 System calls (max 1pt)

- Find out which system calls the program uses.

Function code	System call name	Parameter 1	Parameter 2	Parameter 3
1	exit	return code		
3	read	fd	buffer ptr	size

2 Function/Subroutine calls (max. 2pts)

Function **subrouting_fnc** takes one or two parameters. To make things easier, program uses only integer values and pointers. In our code by integer values we mean data types `char` or `int` and before function call the `char` data type is converted to `int`. We also use only pointers to data types `char` and `int`.

- How many input parameters the **subroutine_fnc** function takes:

    2

- Determine a data type of each parameter of **subroutine_fnc** function:

Parameter	Data type (int/pointer)	Parameter description
Parameter 1	int	fd of input file
Parameter 2	int	numeral system base

- Describe meaning of return value of the **subroutine_fnc**:

    the function returns converted decimal value read from imput file

3 Program (max. 3pts)

- Describe a function of the program and determine a rule for the **toplevel_fcn** return value:

    the function returns the same value as subroutine_fnc

- Describe a function of a whole program on algorithm level. E.g. program calculates an arithmetic average of numbers read from standard input.

    the program reads decimal number from the standard inpu

    then creates file of length given by the function filled by "A" character

Estimate time needed to complete the homework: ....

Listing x86:

program code

```
build/program-x86:      file format elf32-i386

Disassembly of section my_text:

080a9dfc <subroutine_fnc>:
 80a9dfc: 55                       push   %ebp
 80a9dfd: 89 e5                    mov    %esp,%ebp
 80a9dff: 57                       push   %edi
 80a9e00: 56                       push   %esi
 80a9e01: 53                       push   %ebx
 80a9e02: 83 ec 14                 sub    $0x14,%esp
 80a9e05: c7 45 e0 00 00 00 00     movl   $0x0,-0x20(%ebp)
 80a9e0c: 8d 7d f3                 lea    -0xd(%ebp),%edi
 80a9e0f: 8b 75 0c                 mov    0xc(%ebp),%esi
 80a9e12: 83 c6 30                 add    $0x30,%esi
 80a9e15: ba 01 00 00 00           mov    $0x1,%edx
 80a9e1a: 8b 5d 08                 mov    0x8(%ebp),%ebx
 80a9e1d: 89 f9                    mov    %edi,%ecx
 80a9e1f: b8 03 00 00 00           mov    $0x3,%eax
 80a9e24: cd 80                    int    $0x80
 80a9e26: 83 f8 01                 cmp    $0x1,%eax
 80a9e29: 74 0c                    je     80a9e37 <subroutine_fnc+0x3b>
 80a9e2b: bb 01 00 00 00           mov    $0x1,%ebx
 80a9e30: b8 01 00 00 00           mov    $0x1,%eax
 80a9e35: cd 80                    int    $0x80
 80a9e37: 0f b6 45 f3              movzbl -0xd(%ebp),%eax
 80a9e3b: 3c 2f                    cmp    $0x2f,%al
 80a9e3d: 7e 17                    jle    80a9e56 <subroutine_fnc+0x5a>
 80a9e3f: 0f be d0                 movsbl %al,%edx
 80a9e42: 39 f2                    cmp    %esi,%edx
 80a9e44: 7d 10                    jge    80a9e56 <subroutine_fnc+0x5a>
 80a9e46: 8b 45 0c                 mov    0xc(%ebp),%eax
 80a9e49: 0f af 45 e0              imul   -0x20(%ebp),%eax
 80a9e4d: 8d 54 10 d0              lea    -0x30(%eax,%edx,1),%edx
 80a9e51: 89 55 e0                 mov    %edx,-0x20(%ebp)
 80a9e54: eb bf                    jmp    80a9e15 <subroutine_fnc+0x19>
 80a9e56: 8b 45 e0                 mov    -0x20(%ebp),%eax
 80a9e59: 83 c4 14                 add    $0x14,%esp
 80a9e5c: 5b                       pop    %ebx
 80a9e5d: 5e                       pop    %esi
 80a9e5e: 5f                       pop    %edi
 80a9e5f: 5d                       pop    %ebp
 80a9e60: c3                       ret

080a9e61 <toplevel_fnc>:
 80a9e61: 55                       push   %ebp
 80a9e62: 89 e5                    mov    %esp,%ebp
 80a9e64: 57                       push   %edi
 80a9e65: 56                       push   %esi
 80a9e66: 53                       push   %ebx
 80a9e67: 83 ec 20                 sub    $0x20,%esp
 80a9e6a: c6 45 f3 41              movb   $0x41,-0xd(%ebp)
 80a9e6e: c7 44 24 04 0a 00 00     movl   $0xa,0x4(%esp)
 80a9e75: 00
 80a9e76: c7 04 24 00 00 00 00     movl   $0x0,(%esp)
 80a9e7d: e8 7a ff ff ff           call   80a9dfc <subroutine_fnc>
 80a9e82: 89 c7                    mov    %eax,%edi
 80a9e84: ba 80 01 00 00           mov    $0x180,%edx
 80a9e89: b9 42 02 00 00           mov    $0x242,%ecx
 80a9e8e: be 00 7f 0c 08           mov    $0x80c7f00,%esi
 80a9e93: 89 f3                    mov    %esi,%ebx
```

```
80a9e95: b8 05 00 00 00        mov    $0x5,%eax
80a9e9a: cd 80                 int    $0x80
80a9e9c: 89 45 dc              mov    %eax,-0x24(%ebp)
80a9e9f: 85 c0                 test   %eax,%eax
80a9ea1: 79 0e                 jns    80a9eb1 <toplevel_fnc+0x50>
80a9ea3: b8 01 00 00 00        mov    $0x1,%eax
80a9ea8: 89 c3                 mov    %eax,%ebx
80a9eaa: b8 01 00 00 00        mov    $0x1,%eax
80a9eaf: cd 80                 int    $0x80
80a9eb1: 89 7d e0              mov    %edi,-0x20(%ebp)
80a9eb4: 8d 75 f3              lea    -0xd(%ebp),%esi
80a9eb7: eb 22                 jmp    80a9edb <toplevel_fnc+0x7a>
80a9eb9: 8b 5d dc              mov    -0x24(%ebp),%ebx
80a9ebc: 89 f1                 mov    %esi,%ecx
80a9ebe: ba 01 00 00 00        mov    $0x1,%edx
80a9ec3: b8 04 00 00 00        mov    $0x4,%eax
80a9ec8: cd 80                 int    $0x80
80a9eca: 85 c0                 test   %eax,%eax
80a9ecc: 79 09                 jns    80a9ed7 <toplevel_fnc+0x76>
80a9ece: 89 d3                 mov    %edx,%ebx
80a9ed0: b8 01 00 00 00        mov    $0x1,%eax
80a9ed5: cd 80                 int    $0x80
80a9ed7: 83 6d e0 01          subl   $0x1,-0x20(%ebp)
80a9edb: 83 7d e0 00          cmpl   $0x0,-0x20(%ebp)
80a9edf: 75 d8                 jne    80a9eb9 <toplevel_fnc+0x58>
80a9ee1: 8b 5d dc              mov    -0x24(%ebp),%ebx
80a9ee4: b8 06 00 00 00        mov    $0x6,%eax
80a9ee9: cd 80                 int    $0x80
80a9eeb: 85 c0                 test   %eax,%eax
80a9eed: 79 0e                 jns    80a9efd <toplevel_fnc+0x9c>
80a9eef: b8 01 00 00 00        mov    $0x1,%eax
80a9ef4: 89 c3                 mov    %eax,%ebx
80a9ef6: b8 01 00 00 00        mov    $0x1,%eax
80a9efb: cd 80                 int    $0x80
80a9efd: 89 f8                 mov    %edi,%eax
80a9eff: 83 c4 20              add    $0x20,%esp
80a9f02: 5b                    pop    %ebx
80a9f03: 5e                    pop    %esi
80a9f04: 5f                    pop    %edi
80a9f05: 5d                    pop    %ebp
80a9f06: c3                    ret
```

program data

build/program-x86:   file format elf32-i386

Contents of section my_data:
 80c7f00 64617461 00   data.

Listing mips:

program code

build/program-mips: file format elf32-tradlittlemips

Disassembly of section my_text:

```
00404e20 <subroutine_fnc>:
  404e20: 3c1c0005  lui gp,0x5
  404e24: 279c95e0  addiu gp,gp,-27168
  404e28: 0399e021  addu gp,gp,t9
  404e2c: 27bdffd8  addiu sp,sp,-40
  404e30: afb40020  sw s4,32(sp)
```

```
   404e34: afb3001c  sw s3,28(sp)
   404e38: afb20018  sw s2,24(sp)
   404e3c: afb10014  sw s1,20(sp)
   404e40: afb00010  sw s0,16(sp)
   404e44: 0080a021  move s4,a0
   404e48: 00a09821  move s3,a1
   404e4c: 00008021  move s0,zero
   404e50: 27a50008  addiu a1,sp,8
   404e54: 24060001  li a2,1
   404e58: 24120001  li s2,1
   404e5c: 26710030  addiu s1,s3,48
   404e60: 02802021  move a0,s4
   404e64: 24020fa3  li v0,4003
   404e68: 0000000c  syscall
   404e6c: 10e00002  beqz a3,404e78 <__start_my_text+0x58>
   404e70: 00000000  nop
   404e74: 2402ffff  li v0,-1
   404e78: 10520003  beq v0,s2,404e88 <__start_my_text+0x68>
   404e7c: 24040001  li a0,1
   404e80: 24020fa1  li v0,4001
   404e84: 0000000c  syscall
   404e88: 83a30008  lb v1,8(sp)
   404e8c: 00000000  nop
   404e90: 28620030  slti v0,v1,48
   404e94: 14400007  bnez v0,404eb4 <__start_my_text+0x94>
   404e98: 0071102a  slt v0,v1,s1
   404e9c: 10400005  beqz v0,404eb4 <__start_my_text+0x94>
   404ea0: 02130018  mult s0,s3
   404ea4: 00008012  mflo s0
   404ea8: 00701021  addu v0,v1,s0
   404eac: 1000ffec  b 404e60 <__start_my_text+0x40>
   404eb0: 2450ffd0  addiu s0,v0,-48
   404eb4: 02001021  move v0,s0
   404eb8: 8fb40020  lw s4,32(sp)
   404ebc: 8fb3001c  lw s3,28(sp)
   404ec0: 8fb20018  lw s2,24(sp)
   404ec4: 8fb10014  lw s1,20(sp)
   404ec8: 8fb00010  lw s0,16(sp)
   404ecc: 03e00008  jr ra
   404ed0: 27bd0028  addiu sp,sp,40

00404ed4 <toplevel_fnc>:
   404ed4: 3c1c0005  lui gp,0x5
   404ed8: 279c952c  addiu gp,gp,-27348
   404edc: 0399e021  addu gp,gp,t9
   404ee0: 27bdffc8  addiu sp,sp,-56
   404ee4: afbf0030  sw ra,48(sp)
   404ee8: afb3002c  sw s3,44(sp)
   404eec: afb20028  sw s2,40(sp)
   404ef0: afb10024  sw s1,36(sp)
   404ef4: afb00020  sw s0,32(sp)
   404ef8: afbc0010  sw gp,16(sp)
   404efc: 24020041  li v0,65
   404f00: a3a20018  sb v0,24(sp)
   404f04: 00002021  move a0,zero
   404f08: 2405000a  li a1,10
   404f0c: 8f99809c  lw t9,-32612(gp)
   404f10: 00000000  nop
   404f14: 0320f809  jalr t9
   404f18: 00000000  nop
   404f1c: 8fbc0010  lw gp,16(sp)
   404f20: 00409821  move s3,v0
   404f24: 00408021  move s0,v0
   404f28: 8f8481b4  lw a0,-32332(gp)
   404f2c: 24050302  li a1,770
   404f30: 24060180  li a2,384
```

5

```
  404f34: 24020fa5  li v0,4005
  404f38: 0000000c  syscall
  404f3c: 10e00002  beqz a3,404f48 <toplevel_fnc+0x74>
  404f40: 00000000  nop
  404f44: 2402ffff  li v0,-1
  404f48: 04410004  bgez v0,404f5c <toplevel_fnc+0x88>
  404f4c: 00408821  move s1,v0
  404f50: 24040001  li a0,1
  404f54: 24020fa1  li v0,4001
  404f58: 0000000c  syscall
  404f5c: 2610ffff  addiu s0,s0,-1
  404f60: 2402ffff  li v0,-1
  404f64: 12020010  beq s0,v0,404fa8 <toplevel_fnc+0xd4>
  404f68: 02202021  move a0,s1
  404f6c: 27a50018  addiu a1,sp,24
  404f70: 24060001  li a2,1
  404f74: 2412ffff  li s2,-1
  404f78: 24020fa4  li v0,4004
  404f7c: 0000000c  syscall
  404f80: 10e00002  beqz a3,404f8c <toplevel_fnc+0xb8>
  404f84: 00000000  nop
  404f88: 2402ffff  li v0,-1
  404f8c: 04410003  bgez v0,404f9c <toplevel_fnc+0xc8>
  404f90: 24040001  li a0,1
  404f94: 24020fa1  li v0,4001
  404f98: 0000000c  syscall
  404f9c: 2610ffff  addiu s0,s0,-1
  404fa0: 1612fff5  bne s0,s2,404f78 <toplevel_fnc+0xa4>
  404fa4: 02202021  move a0,s1
  404fa8: 24020fa6  li v0,4006
  404fac: 0000000c  syscall
  404fb0: 10e00002  beqz a3,404fbc <toplevel_fnc+0xe8>
  404fb4: 00000000  nop
  404fb8: 2402ffff  li v0,-1
  404fbc: 04410005  bgez v0,404fd4 <toplevel_fnc+0x100>
  404fc0: 02601021  move v0,s3
  404fc4: 24040001  li a0,1
  404fc8: 24020fa1  li v0,4001
  404fcc: 0000000c  syscall
  404fd0: 02601021  move v0,s3
  404fd4: 8fbf0030  lw ra,48(sp)
  404fd8: 8fb3002c  lw s3,44(sp)
  404fdc: 8fb20028  lw s2,40(sp)
  404fe0: 8fb10024  lw s1,36(sp)
  404fe4: 8fb00020  lw s0,32(sp)
  404fe8: 03e00008  jr ra
  404fec: 27bd0038  addiu sp,sp,56

  program data


  build/program-mips:     file format elf32-tradlittlemips

  Contents of section my_data:
   446400 64617461 00000000                    data....
```