Lab 2

Tips, DD, Open OH

Admin

- Lab 2 has 2 parts with separate design docs and due dates
  - part 2 design due 2/01 (today, **no late days**)
  - part 1 code due 2/02 (with late days)
  - part 2 code due 2/09 (with late days)
- Pset/ Quiz 2 due next Friday 2/9
  - 11:59pm
  - No late days
  - 30 Questions on Gradescope on Week 4-6 content
  - Not timed

Pipe Impl Hints

- Variant of the bounded buffer problem
  - producer = writer, consumer = reader
- When should a writer wait?
  - A: No room to write **and** still readers left
  - When should reader wait?
- What should happen when all writers are closed
  - What if sleeping readers? What should happen?
  - What new reader comes? What should happen?

exec(program, args): args setup

As a reminder, all programs should have the following main signature:

int main(int argc, char** argv)

argc: The number of elements in argv

argv: An array of strings representing program arguments

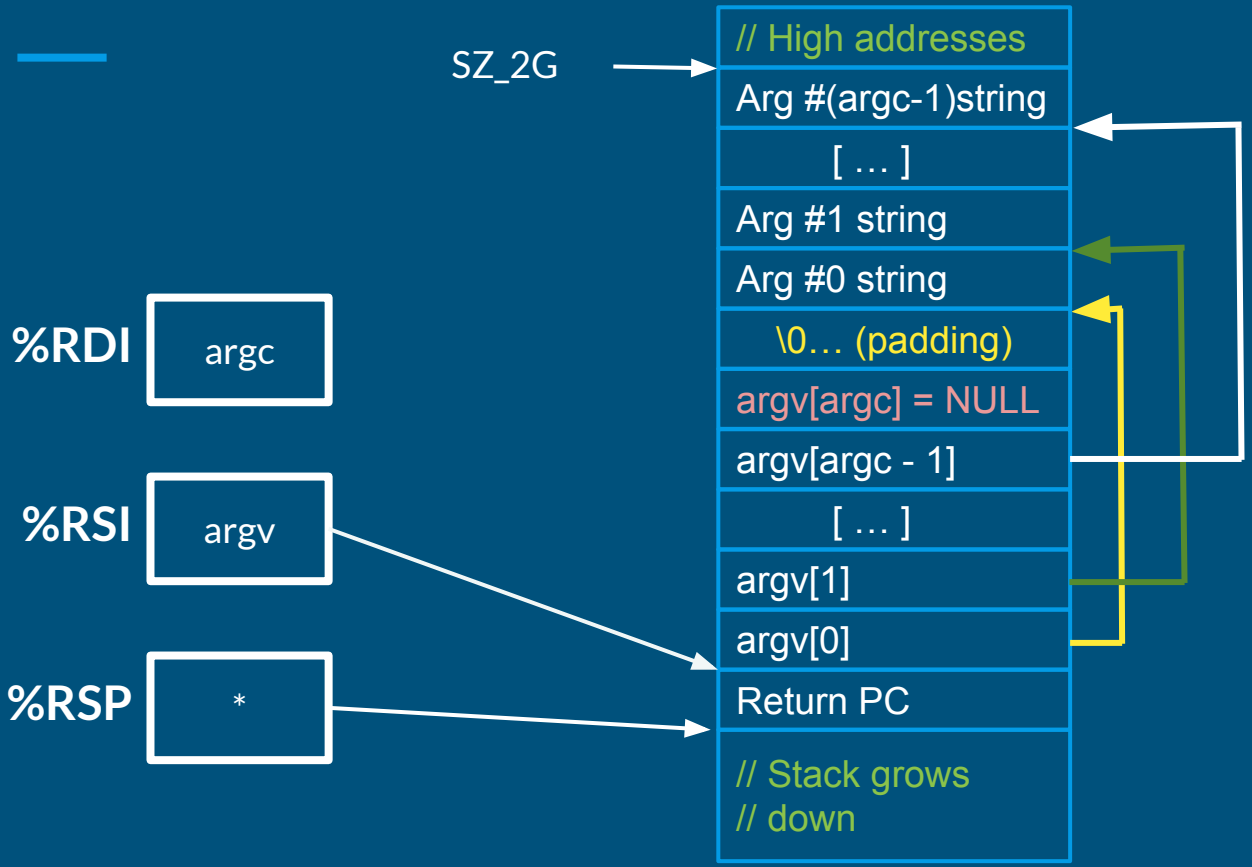
- First is always the name of the program
- Argv[argc] = 0

X86_64 Calling Conventions

- %rdi: holds the first argument
- %rsi: holds the second argument
  - %rdx, %rcx, %r8, %r9 comes next
  - overflows (arg7, arg8 …) onto the stack
- %rsp: points to the top of the stack (lowest address)

- Local variables are stored on the stack
- If an array is an argument, the array contents are stored on the stack and the register contains a pointer to the array’s beginning

Stack For User Process

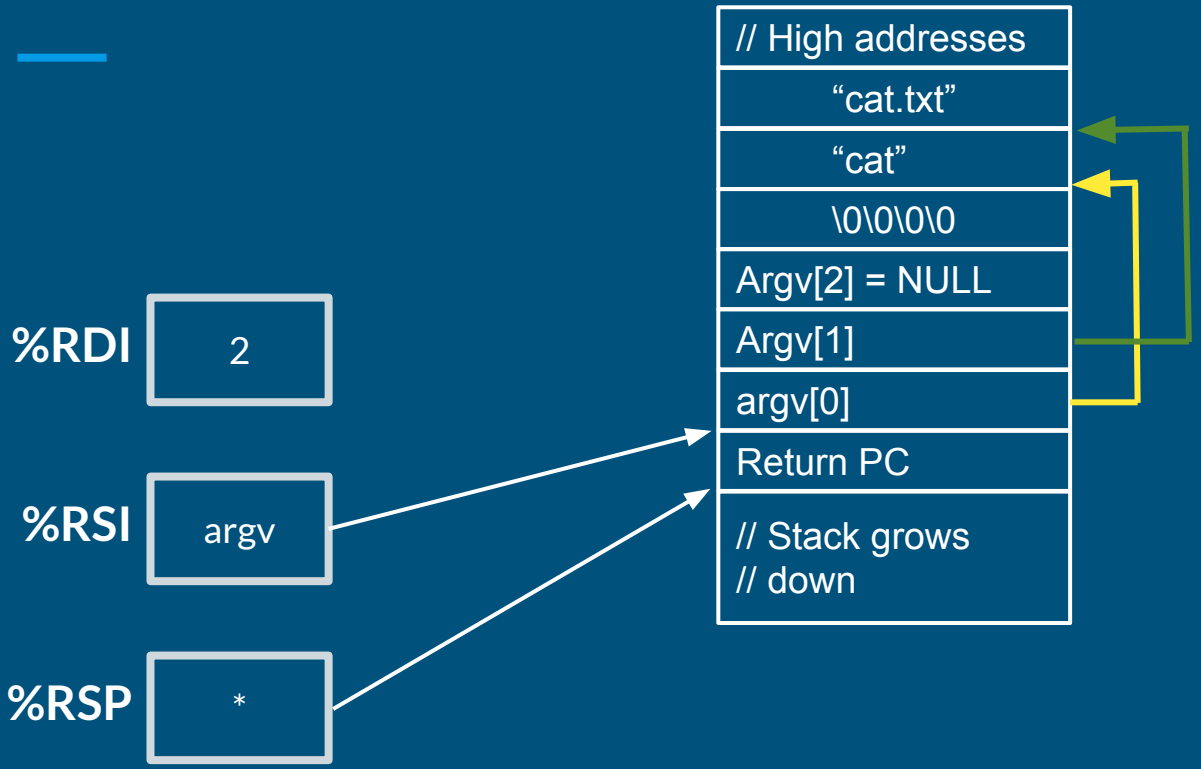


- Since argv is an array of pointers, %RSI points to an array on the stack
- Since each element of argv is a char*, each element points to a string elsewhere on the stack
- Why? Alignment
- Why NULL pointer? Convention

Practice Exercise 1

TODO:
Draw stack layout and determine register values for exec called with “cat cat.txt”

Practice Exercise 1: Solution

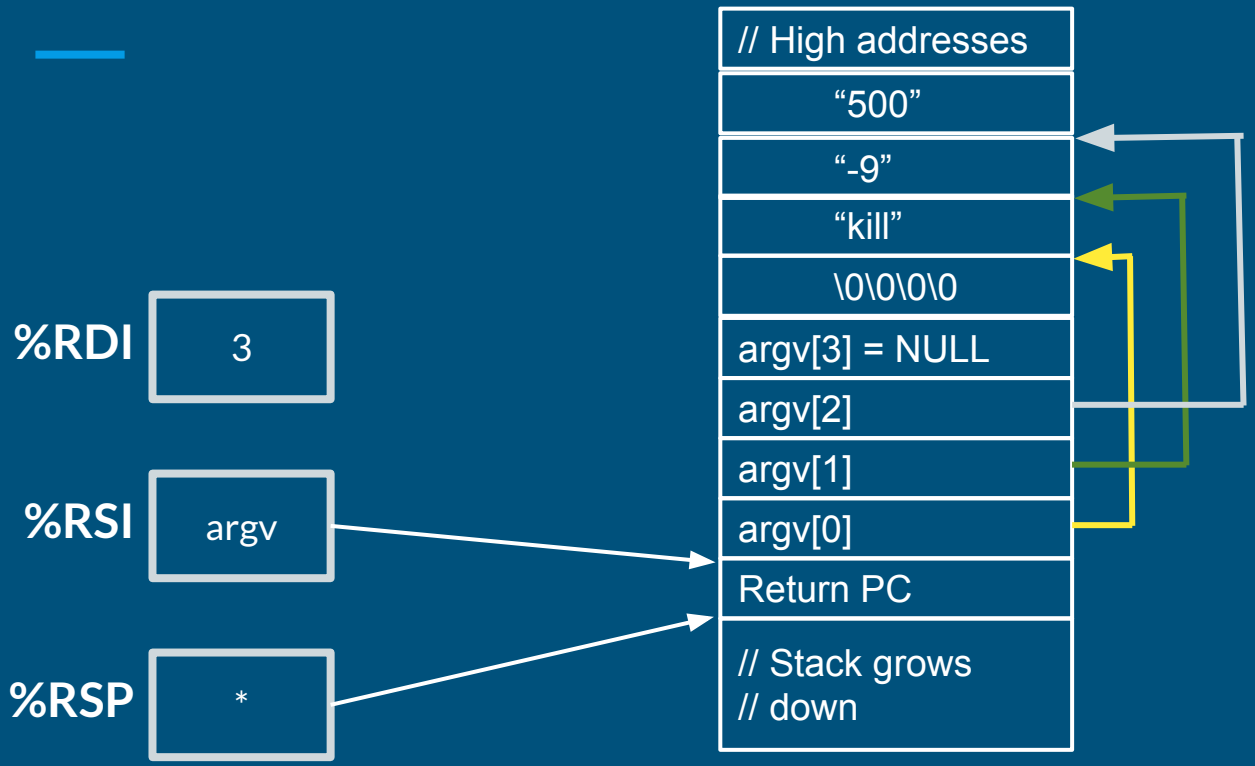


- RDI holds argc, which is 2
- RSI holds argv: the beginning of the argv array
- RSP is properly set to the bottom of the stack.
- The specific value of the return PC doesn’t matter (program exits from main without returning)

Practice Exercise 1

TODO:
Draw stack layout and determine register values for exec called with "kill -9 500"

Practice Exercise 2: Solution



- RDI holds argc, which is 3
- RSI holds argv: the beginning of the argv array
- RSP is properly set to the bottom of the stack.
- The specific value of the return PC doesn't matter (program exits from main without returning)

exec tests

- requires pipe!

Part 2 Design Doc Peer Review (~10 mins)

- Get into groups of 2 and exchange your design docs for peer review
- Did you learn new cases you hadn't thought about?
- Is there anything you can help out for your peers?
- What are some unanswered questions still?

Lab 2 Questions

- In the slides for Lab 2 Part 2, it mentioned that part of the pipe metadata you need is the PID of the waiting writer but we are not sure why that is necessary. What is this useful for?

Lab 2 Open OH