Università della Svizzera italiana

Facoltà di scienze informatiche

Operating Systems **Spring - 2024**

Prof. Fernando Pedone, Eliã Rafael L Batista, Lorenzo Martignetti, Nenad Milosevic

Project 4 - User programs PintOS

In this assignment you should implement essential features that are missing in the current PintOS implementation that are related to user programs.

System call handler

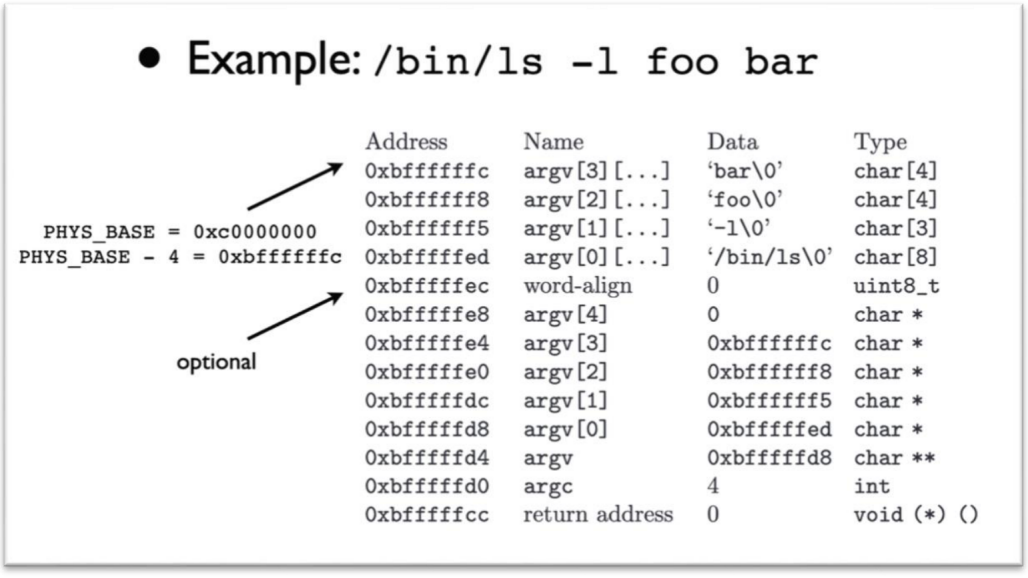
You should implement system call handler (in ***"userprog/syscall.c"***) in order to handle $write$ (you should only consider printing to a $stdout$) and $exit$ system calls.

Hints

- System call numbers are defined in ***"lib/syscall-nr.h"***.
- System call handler has access to registers.
  - Stack pointer is $f \rightarrow esp$.
  - Save the return value to $f \rightarrow eax$.
- Use function $putbuf()$ to print to $stdout$.

Argument passing

In the current implementation kernel is not passing the arguments to the executable. In order to do that you should push arguments to the stack in a proper way and at the proper moment. You can see an example of how you should push arguments on the stack in the picture below.

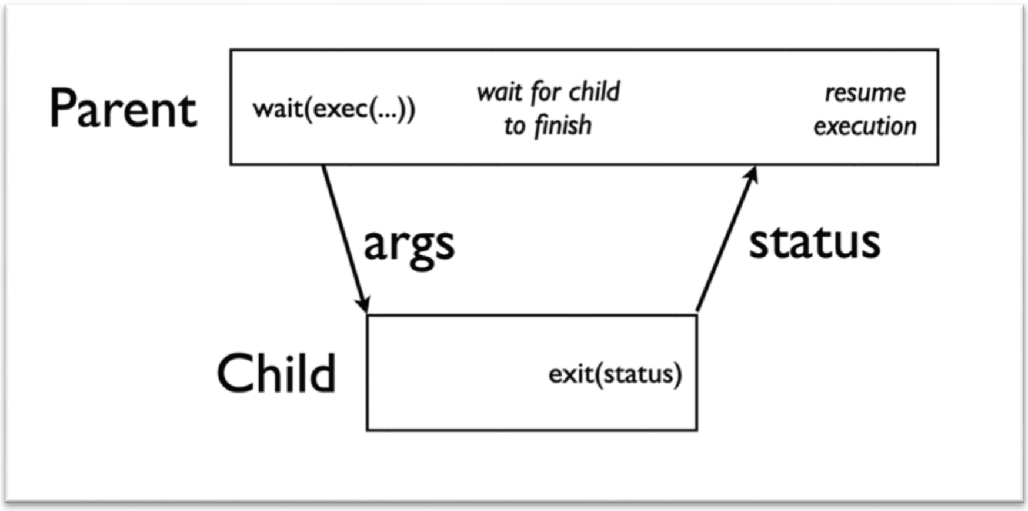


Hints

- Look at functions $process\_execute()$ and $start\_process()$ in ***"userprog/process.c"***.
- Use function $strtok\_r()$ to tokenize the command line.
- Remember that the stack grows downwards.

Process wait

You have to implement function $int\ process\_wait(tid\_t\ child)$ in ***"userprog/process.c"***. Calling process/thread should be blocked until child exits. This is shown on the diagram below. This function is used in PintOS when starting the program and that is the reason why we need to implement it.



Hints

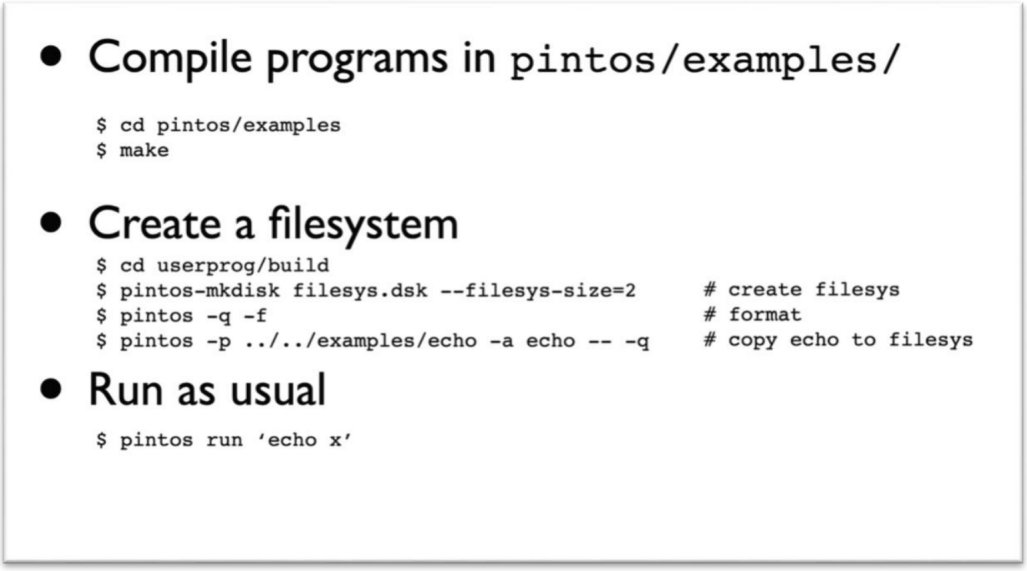
- Child keeps track of parent thread.
- Use $thread\_block()$ to block the parent when $process\_wait()$ is called.
- When child exits, $thread\_unblock()$ parent thread.

Tests

Your implementation should pass first five tests:

- $args - none$
- $args - single$
- $args - multiple$
- $args - many$
- $args - dbl - space$

In addition, when you finish this project you should be able to run ***"examples/echo"*** program. Procedure that shows how you can run this program is shown on the picture below.



*If command **"*pintos -q -f*"** is not working you need to replace **"*CFLAGS = -g -msoft-float -O*"** with **"*CFLAGS = -g -msoft-float -O0*"** in your **"*Make.config*"**.

Readings

- PintOS documentation
  - Chapter 3 (You can skip sections 3.1.5 and 3.3.5.)