

Project 1

- **Description:**

1. The user address space of a Linux thread consists of the following segments.
 - text segment
 - data segment (global variables with initial values)
 - BSS segment (global variables without initial values)
 - heap segment (memory area allocated through function `malloc()`)
 - libraries
 - stack segment
2. In this project, you need to add some new system calls first. The functionality and number of the new system calls are determined by you.
3. Write a multi-thread program with three threads (main thread, thread 1, and thread 2) and the new system calls to check which segments of a thread are shared by which other thread(s).
4. You do not need to calculate the size, star address, and end address of each thread segment. However, if you can obtain the the size, star address, and end address of each thread segment, we will add **30** extra points to your project 1 grade.
5. Write a report to describe the results created by your multi-thread program. Your report should contain a figure like the following one to summarize your results.

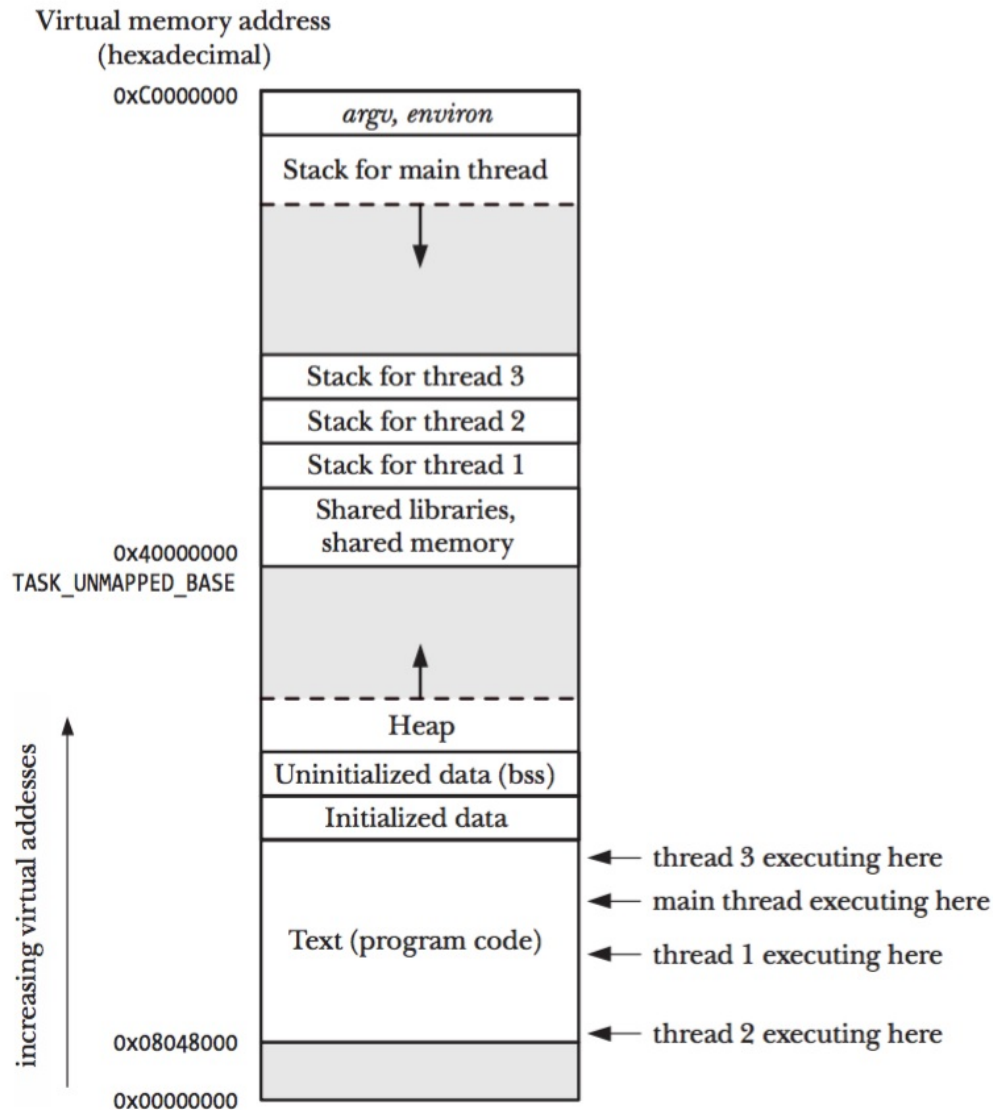


Figure 29-1: Four threads executing in a process (Linux/x86-32)

-- by Jason/cntofu.com

6. Hint:

- Global data pointer variables and global function pointer variables are shared by all threads; hence, they are ideal items to pass information among threads.
- Two threads show a physical memory cell (one byte) if both of them have a virtual address that is translated into the physical address of the memory cell.
- The kernel usually does not allocate physical memories to store all code and data of a process when the process starts execution. Hence, if you want kernel to allocate physical memories to a piece of code, execute the code first. If you want kernel to allocate physical memories to a variable, access the variable first.
- Inside the Linux kernel, you need to use function `copy_from_user()` and function `copy_to_user()` to copy data from/to a user address buffer.
- Check the "Referenced Material" part of the Course web site to see how to add a new system call in Linux.

- **Project Submission:**

- You need to upload an electronic version of your project report through new eeclass before **23:55 11/8**.
 - Do not forget writing the names and student IDs of all members in your team in your report.
 - Your report should contain:

- the source code of your multi-thread application and your kernel code
 - the execution results
 - a report describing and analyzing the results of your experiments.
- Late submission is not allowed.
- Demo time
 - The demos begin on **11/9** and end on **11/11**.
 - Please write your demo time in this [form](#) before **23:55 11/6**.
 - For NCU students, the demos will proceed through on-site meetings. Every member of a team must show up in the team's demo. We will post the location of the demo classroom before the demo.
 - For NTHU students, the demos will proceed through google meetings. Every member of a team must show up in the team's demo. The related link will be posted before the demo.
- When demonstrating your projects, the TAs will ask you some questions regarding to your projects. Part of your project grade is determined by your answers to the questions.
- Recommended Linux Distribution: Fedora or Debian.
- Kernel version: versions around Linux 3.9.
- [關於 Project 的一些注意事項](#) - by 林哲安
- **Reference:**
 - G. T. Wang, [C 語言 pthread 多執行緒平行化程式設計入門教學與範例](#)。
 - Jason/cntofu.com, [深入 Linux 多線程編程](#)。
 - Will, [C pthread create 傳遞參數的用法](#)。
 - Chin-Hung Liu, [Work Note-pthread](#)。
 - MIT, [Thread-Local Storage](#)