

# Hibernation of Private Processes

Rakhi Bhardwaj<sup>1</sup>, Delwin John<sup>2</sup>, Nikhil Shinde<sup>3</sup>, Mehak Daftari<sup>4</sup>, Vrushali Deshpande<sup>5</sup>

<sup>1</sup>Assistant Professor, Department of Computer Engineering, TCOER, SPPU, India

<sup>2,3,4,5</sup> Student, Department of Computer Engineering, TCOER, SPPU, India

**Abstract:** Sometimes when we are not going to be using our computers for few hours, we should put the computer to sleep, also known as the standby mode. If we are not going to be using it for a few days, then it is best to power it off. However, there are few Operating Systems that offer the “hibernate” mode. This hibernate mode crashes after implementation of incognito mode as a service of the Linux OS. In this paper, we try to overcome this restriction that is imposed on the user.

**Keywords:** private processes, hibernation, partial hibernation, complete hibernation, swap space

## 1. Introduction

Hibernation is the process of powering down a computer and continuing to keep a hold on its state. The contents or data is saved on the RAM of computer during hibernation. When exiting the hibernation state, the state of computer gets back to its previous session before getting into hibernation. Thus prevents loss of data with a very low risk. In situations where our data might get lost, due to sudden powering off of machine, hibernation saves the system’s state, to resume when it is turned on again. Power consumption is turned off completely in hibernate mode.

In our previous paper we wrote about whitelisting and blacklisting of private process. In that we mentioned about spawning applications in a private mode, ensuring safety and privacy of data. Due to this, the hibernation mode of the Linux OS crashes completely. In this paper, we mention ways to overcome that side effect. For hibernation mode to work in our project we need to create different method for hibernating the system such that, we can enable the hibernate mode without causing problems to swap partition and securely killing the private process without leaving any traces on our hard disk. It should securely delete all our history, log files, data, running applications, processes. No one should be able to recover the data when the system is resumed again. Also, all other public processes, data, documents, running applications will be saved on hard disk, which can be resumed when system is turned on again.

## 2. Current System

In computing, nowadays software suspends is known as hibernation in Linux operating system. There are basically three methods to suspend -

Suspend to RAM  
Suspend to disk  
Hybrid

In suspend to RAM, the machine status can be reinstated by cutting power off of most machine except RAM, which can prove benefit for power saving, can be used laptop as run on batteries and switch to that mode spontaneously. In suspend to disk, it stores the machine state into swap space and fully powered down the machine. The state of the machine is

resumed as soon as the machine is powered on. And lastly, in hybrid, mode, it stores the machine’s state onto swap space and doesn’t power down completely. As substitute it calls suspend to RAM. So that, if battery is not drained or consumed the system can get back to its previous session from RAM. And if the battery is drained, system can be rolled back from the disk. System rollback from disk is considered to be very slow process as compared to RAM. The advantage it provides is that machine state is not lost.

### 2.1 Hibernation

To use hibernation, one needs to create a file or swap partition. This is unique type of partition where data is saved. With the help of “resume” parameter we can point kernel to swap/file. It is configure by boot loader. One also needs to add resume hook to the config file of intramfs generator mkinitcpio. It notifies the kernel to try rolling back from the mentioned swap which was in userspace.

### 2.2 Swap Partition / File Size

If you have tiny swap partition than RAM, you still can execute hibernation efficiently. The size of the image generated by suspend to disk mechanism is controlled by the command `/sys/power/imagesize`. For the image size non negative integer is used as upper limit. The size of the image doesn’t exceed with the help of suspend to disk method. Even if there are difficulties during suspend it will still execute with even a smallest image. The size of the suspended image will be really small even when “0” is written to the file. One can put minimum value of `/sys/power/image_size` so that the size of suspend image will be small and also maximize the value for speeding the hibernation process. We can use swap file over swap partition that requires an add-on kernel parameter i.e. `resume_offset=swap_file_offset`. When we run the `filefrag -v swap_file` command the value of `swap_file_offset` is taken as an output. Then that output is put into a table format and that value is found in the `physical_offset` column.

## 3. Proposed System

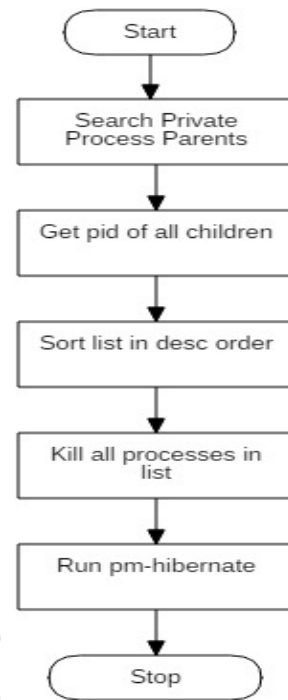
The aim of this project is to modify the kernel such that the storage writes are non-recoverable when application is

executed in incognito mode. Due to modification of the kernel to allow private execution of an application, the hibernation option of the system is not functional. Thus the aim of our project is to enable the hibernation option of Linux OS. The user after spawning an application in private mode should be able to hibernate his/her system if he/she wishes to do so. Since, during hibernation the data is copied to the swap memory, the private data could get copied to the memory and anybody who resumes the machine later on can access the data. Hence, this should be implemented without affecting the basic idea of protecting the user's data. We solve this problem by pausing each private process and moving them completely into swap memory. For this, you walk through each process running on the system and checked is it's a private process. If it's a private process, you fetched the memory map of the process and convert each address into page structure. Then you swap it into memory. We hibernate using the pm-hibernate in VMware Player. There are two techniques to enable hibernation in a private mode –

- Partial hibernation
- Complete hibernation

Partial hibernation, as the name suggests, is when a system goes into locking down the state of the system. But only the public processes are saved. For this, the private processes do not work and need to be killed separately. In partial hibernation, we search for all the private processes first and then look for all their children. We then find the process ids (pid) of these children and maintain them in a list. Once all the pids have been found, we then sort this list in descending order. Then we kill all these processes starting from the top. After killing the processes we run pm-hibernate on the virtual machine. Partial hibernation is more secure and safe in protecting the private session as all the processes are killed down. The only downfall is that the private session is completely lost after hibernation.

Complete hibernation, as the name suggests, is when all data, whether of public process or private, gets copied to the swap memory and can be resumed when powered on again. For complete hibernation the kernel maintains a linked list of all processes running in the system. The linked list has init process as the head. We walk through the entire linked list and check every process's flag. If it is a private process then we walk through the process's virtual memory area to determine the virtual address that the process owns. We then plan on converting the virtual address to page structures. The pages are then paged out. Since they are private processes, the paging out would save the pages in the process's private container. This is written as a kernel module that is loaded and run just before hibernate. It then puts the computer through the normal hibernation process. Complete hibernation is better at saving the state of the system after hibernation as all the private processes are stored in a private container. The downfall of this method is the fact that it does not ensure security of one's data, as anybody can power on the system and the last session will be restored. Thus making our data vulnerable towards theft.



**Figure 1:** Hibernating private processes

#### 4. Mathematical Model

$S = \{s, e, i, o, f, DD, N DD, success, failure\}$

- s: Start State = system waiting to go in hibernation
- e : End State = system ready for normal hibernation
- i : Input = no required input
- o : Output = private programs terminated
- f : Functions = {f1 , f2 , f3 , f4 }
- f1 : search\_private\_process\_parents()
- f2 : generate\_list\_of\_children()
- f3 : sort\_pids()
- f4 : kill\_processes()
- DD: Deterministic Data
- N DD: Non Deterministic Data
- Success: successful termination of all private processes
- f failure: failure in terminating all private processes

The time complexity and space complexity are dependent on the number of private processes that are currently running on the system (m)

The time and space required to generate list of all private process parents is dependent on the number of private processes running

$T1 = O(m)$   
 Space complexity =  $O(m)$

The time and space required to generate list of all children of private processes is dependent on the number of private processes running

$T2 = O(m)$   
 Space complexity =  $O(m)$

The time for sorting the list of pids is proportional to the square of the entire list of pids. Since the number of pids is directly proportional to the number of private processes

$T3 = O(m^2)$   
Space Complexity =  $O(m)$

The time and space required to kill all private processes is dependent on the number of pids which is directly dependent on the number of private processes running

$T4 = O(m)$   
Space Complexity =  $O(m)$   
 $T(n) = O(m + m + m^2 + m) = O(m^2)$   
Space complexity =  $O(m)$

## 5. Expected Result

On the execution of the hibernate mode for private processes the system will be powered off with its state being saved securely and all the private processes will be safely deleted in case of partial hibernation

## 6. Conclusion

Thus with the help of partial and complete hibernation we can provide users with the option of hibernating the system while using incognito mode. This allows the users with ease to use the system whenever they want, without worrying about the safety of their data

## References

- [1] [http://www.freesoftwaremagazine.com/articles/hibernate\\_linux](http://www.freesoftwaremagazine.com/articles/hibernate_linux) (General Internet site)
- [2] [http://wiki.archlinux.org/index.php/Power\\_management/Suspend\\_and\\_hibernate](http://wiki.archlinux.org/index.php/Power_management/Suspend_and_hibernate)

