BAHIRDAR UNIVERSITY

INSTITUTION OF TECHNLOGY FACULITY OF COMPUTING DEPARTMENT OF SOFTWARE ENGINEERING

Operating system and system programming

Individual assignment

Title - system call (get\_mempolicy)

Name- Nuhamin Zelalem

ID NO -1313860

Submitted to: wedimu baye

Submission date: July, 24 2022

**What is get\_mempolicy** ()

First system call is the programmatic way in which a [computer program](https://en.wikipedia.org/wiki/Computer_program) requests a service from the [kernel](https://en.wikipedia.org/wiki/Kernel_(operating_system)) of the [operating system](https://en.wikipedia.org/wiki/Operating_system) on which it is executed.

* Thesystemcall **get\_mempolicy** () retrieves the NUMA (Non Uniform Memory Access) policy of the calling thread or of a memory address, depending on the setting of *flags.* Which consists of a policy mode and zero or more nodes, to the values specified by the mode, nodemask, and maxnode arguments.
* The **get\_mempolicy** () Linux system call was added to the Linux kernel in version 2.6.7
* We can check the call if the **get\_mempolicy** () returns **0** it means a success and on error, -**1** is returned and [**errno**](https://man7.org/linux/man-pages/man3/errno.3.html) is set to indicate the error.
* The mempolicy mechanism gives control over how a process’s memory is allocated across the node of a numa system. From which node memory is allocated for the thread.
* Numa nodes are obtained from different scenes according to the incoming parameters, which will not be analyzed in detail.
* A NUMA machine has different memory controllers with different distances to specific CPUs

**List of parameter and flags**

**Parameters:**

* **unsigned** **long flag:** if the flag is 0, it indicates that the mempolicy of the current thread is obtained, and the addr parameter is NULL. If it is flags, it is MPOL\_ F\_ MEMS\_ Allow, the mode parameter will be ignored, and nodemask will return the numa node supported by the configuration. If the flag parameter is MPOL\_F\_ADDR, the result will return the mempolicy numa node corresponding to the specified addr address.
* **int mode**: if the mode parameter is not 0, the result returns the mempolicy node of the specified mode.
* **Unsigned long** \*nodemask: an array similar to unsigned long, with the size of maxnode/sizeof(unsigned long).
* **Unsigned long maxnode**: the maximum number of nume nodes supported.

**Flags**

* mode – NUMA mode
* nodemask – pointer to mask defining node that mode applies to
* maxnode – max number of bits for nodemask
* addr – pointer to memory region
* flags – defines behavior of call

**MPOL\_F\_NODE** or 0 – get information about calling thread’s default policy and store in nodemask buffer.

**MPOL\_INTERLEAVE** this mode specify that page allocation be interleaved on page aggregating in the node specified in the policy.

This mode interleaves page allocations across the nodes specified in *node mask* in numeric node ID order.

This optimizes for bandwidth instead of latency by spreading out pages and memory accesses to those pages across multiple nodes. However, accesses to a single page will still be limited to the memory bandwidth of a single node**.**

**MPOL\_F\_MEMS\_ALLOWED** mode argument ignored and subsequent call return set of nodes thread is allowed to specify is returned in nodemask.

**MPOL\_F\_ADDR** get information about policy for addr, the information is returned about the policy governing the memory address given in *addr.*

**MAP\_PRIVATE**

**MPOL\_ F\_ MEMS\_**

**MAP\_SHARED**

**Code implementation**

**int get\_numa\_node\_id(void\* ptr)**

**{**

**int status = -1;**

**get\_mempolicy(&status, NULL, 0, ptr, MPOL\_F\_NODE | MPOL\_F\_ADDR);**

**return status;**

* This code will returen 0 for succes and -1 for error.

