Rocky Arkan Adnan Ahmad

1806186566

SysProg-A

Hal yang sudah dipelajari dan dipahami:

1. Memori segment dari proses
    a. **Environment variable and command line args**: Environment variable dan args dari command line disimpan di proses memory layout paling atas.
    b. **Stack segment:** Stack segment digunakan untuk menyimpan local variable dari sebuah fungsi dan juga informasi lainnya yang disimpan ketika sebuah fungsi dipanggil. Informasi tersebut bisa berupa return address dari fungsi.
    c. **Heap segment:** Heap segment digunakan untuk dynamic memory allocation.
    d. **Block Started by Symbol (BSS) segment**: Semua global variable yang tidak diinialisasi oleh program disimpan kedalam BSS segment.
    e. **Data segment**: Sedangkan semua global variable yang diinialisasi oleh program disimpan di data segment.
    f. **Text Segment**: Kumpulan instruksi mesin yang akan dieksekusi oleh CPU disimpan di text segment.

2. Virtual Memory
   Virtual memory masih dibutuhkan hingga saat ini karena physical memory kita terbatas. Kita tidak bisa menjamin kalau program yang berjalan akan pasti bisa berjalan tanpa kehabisan memori. Dengan virtual memory, kita bisa menjalankan suatu program walaupun memory physical yang dibutuhkan program jauh lebih besar dari memory physical yang kita punya.

3. Spatial dan Temporal locality di dalam memori
   Memori yang kita gunakan umumnya bersifat Spatial dan temporal locality.

   Special locality: adalah kecenderungan dimana data digunakan dalam tempo waktu yang dekat. Misalnya, jika kita menggunakan suatu data, maka kemungkinan besar kita akan menggunakannya lagi dalam jarak waktu yang dekat.

   Temporal locality: adalah kecenderungan kumpulan data yang berhubungan berada dalam lokasi yang berdekatan dalam memori. Misalnya, sebuah array akan disimpan dalam lokasi yang bersebelahan.

4. Memory Leak

   Memory leak adalah suatu kejadian dimana virtual memory yang sudah tidak digunakan masih tersimpan di memori atau tidak dibebaskan dari memori, sehingga memakan space dari memori.

   Memory leak terjadi ketika program mengalokasikan memori di heap menggunakan malloc(), namun tidak membebaskannya dengan free() sebelum program berakhir, sehingga memori yang dialokasikan tersebut tidak bisa digunakan lagi, sehingga terjadi memory leak.

5. Malloc, free, dan sbrk()

   Malloc akan mengalokasikan memori di heap dari process. Malloc dapat dilakukan jika kita membutuhkan ekstra memori dari program kita. Lalu, kalau kehabisan space di heap, maka akan memanggil sbrk() untuk melebarkan program break (batas dari data segment) sampai batas maksimal heap. Jika sudah maksimal maka tidak bisa dilebarkan lagi. Untuk membebeaskan alokasi memori yang sudah tidak dipakai, maka bisa menggunakan free(). Jika free tidak dipanggil sesudah malloc, maka akan terjadi memory leak karena memori yang dialokasikan tidak dipakai lagi, namun tidak bisa digunakan kembali lagi oleh proses lain karena belum difree().

   . malloc memanage free memory dengan menggunakan free block list. Free block list adalah suatu stuktur data yang akan mengisi heap dengan blok-blok yang berbentuk linked-list. Blok-blok tersebut berisi pointer yang menuju ke blok selanjutnya, dengan pointer blok terakhir akan berisi null karena sudah akhir dari linked-list. Malloc mempunyai pointer yang merujuk ke pada bagian awal / head dari linked list. Ketika mengalokasi blok, blok akan dilokasikan dari head sampai blok yang ingin dialokasikan. Lalu pointer dari blok terakhir yang dialokasikan akan merubahnya menjadi null agar tidak tersambung ke blok yang belum diissi. Lalu, malloc akan mereturn pointer yang disimpan, yaitu head dari linked list ke dalam pemanggil fungsi di program. Lalu malloc mengubah pointer yang disimpan menjadi pointer menuju blok yang dituju oleh blok terakhir yang dialokasi, sehingga head dari linked list sekarang berubah menjadi blok berikutnya dari blok terakhir yang dialokasi. Untuk free, maka blok terakhir linked-list akan merubah pointer nullnya menjadi pointer yang merujuk ke blok yang difree, sehingga blok yang difree sekarang berada di akhir dari linked-list. Karena blok yang difree ketika dialokasi blok terakhirnya sudah berubah menjadi null, maka tidak perlu mengubah pointernya menjadi null.

6. Longjmp, setjmp

Setjmp akan menyimpan stackpointer dari program counter setjmp dan menyimpannya ke dalam sebuah buffer. Ketika dipanggil longjmp, maka akan mengepop stack pointer dan akan goto ke PC dari setjmp. Setjmp akan mereturn 0 ketika dipanggil langsung, sedangkan akan mereturn value sesuai dengan argument yang diberi longjmp ketika dipanggil dari longjmp.

Hal yang dipelajari namun belum dipahami:

Untuk minggu ini belum ada materi yang belum saya pahami.