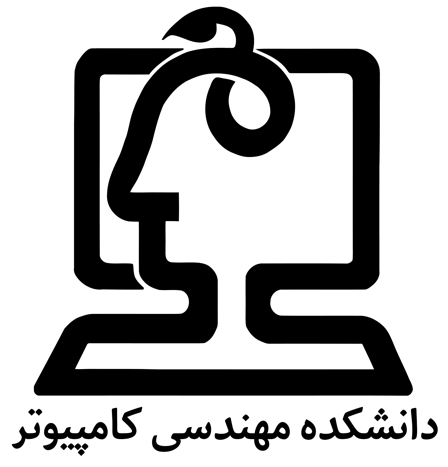
«به نام ایزد یکتا»



گزارش نهم درس آزمایشگاه سیستم عامل

­­

استاد: مهندس قاسمی

تهیه کننده: بردیا اردکانیان

۹۸۳۱۰۷۲

MEMORY ALLOCATION

بخش اول: SEGMENTATION

کد این بخش در فایل segment.c ساختمان داده‌ای برای پیاده سازی قابل مشاهده است. پیاده سازی به این صورت است که پردازه‌ای ساخته می‌شود و با مشخص کردن اندازه سگمنت، از سگمنت‌های قابل استفاده به پردازه اختصاص داده می‌شود.

در این پیاده‌سازی از ساختمان‌داده‌ای استفاده می‌کنیم که پنج مشخصه دارد:

1. **آدرس شروع**
2. **اندازه سگمنت**
3. **علامت**
4. **پر شده ؟ ۰/۱**
5. **سگمنت بعدی**

کارکرد این الگوریتم به این صورت است که با ساختن پردازه جدید و گرفتن id متمایز، از کاربر اندازه Text, data, heap را ورودی می‌گیرد. بعد به سه شکل می‌تواند برای پردازه سگمنتیشن را انجام دهد:

1. First fit
2. best fit
3. worst fit

با توجه به ورودی کاربر یکی از این سه را انجام می‌دهد. مثالی در ذیل قابل مشاهده است:

The initial memory partition:  
Total mem avail=95  
0  8 0  
8  7 0  
15  10 0  
25  4 0  
29  2 0  
31  16 0  
47  11 0  
58  13 0  
71  10 0  
81  14 0  
  
Segment tables  
  
Enter choice:  
1>New process  
2>Deallocate  
3>Exit  
1  
  
Enter text, data and heap size  
10 10 10  
  
Enter 1 for First Fit,2 for Best fit and 3 for worst fit  
1  
Allocating Text for process no. 1  
Allocating Data for process no. 1  
6 units of memory lost due to internal fragmentation  
Allocating Heap for process no. 1  
1 units of memory lost due to internal fragmentation  
Total mem avail=95  
0  8 0  
8  7 0  
15  10 10  
25  4 0  
29  2 0  
31  16 11  
47  11 12  
58  13 0  
71  10 0  
81  14 0  
  
Segment tables  
Process:1 Segment:0 Base:15 Limit:10  
Process:1 Segment:1 Base:31 Limit:16  
Process:1 Segment:2 Base:47 Limit:11  
  
Enter choice:  
1>New process  
2>Deallocate  
3>Exit

بخش دوم: Paging

کد این بخش در فایل page.cpp قابل مشاهده است. در این بخش نیز مانند بخش قبلی از ساختمان‌داده‌ای استفاده می‌کنیم. این ساختمان داده تنها دو بخش دارد:

1. id
2. page number

این الگوریتم کارکرد بسیار ساده‌ای دارد. مانند قبلی پردازه جدیدی می‌سازد و اندازه page را ورودی می‌گیرد. بعد با توجه به صفحه‌های خالی و انتخاب آنها، این صفحه‌ها را به پردازه اختصاص می‌دهد. همچنین می‌توان حافظه را ازاد کرد تا بقیه پردازه‌ها نیز استفاده نماییند. مثالی در ذیل قابل مشاهده است:

|  |
| --- |
| /Users/Bardia/Documents/aut/courses/CE304-OS-Lab/assignment/9/Codes/cmake-build-debug/9 The initial memory: No  ID  page no  0   0   0 1   0   0 2   0   0 3   0   0 4   0   0 5   0   0 6   0   0 7   0   0 8   0   0 9   0   0  Enter choice: 1>New process 2>Deallocate 3>Exit 1  Enter size 100 28 units of memory lost to internal fragmentation No  ID  page no  0   1   1 1   0   0 2   0   0 3   1   2 4   0   0 5   0   0 6   1   3 7   1   4 8   0   0 9   0   0 Page Tables: Page no:1 Frame no:0 for process 1 Page no:2 Frame no:3 for process 1 Page no:3 Frame no:6 for process 1 Page no:4 Frame no:7 for process 1   Enter choice: 1>New process 2>Deallocate 3>Exit 1  Enter size 3 29 units of memory lost to internal fragmentation No  ID  page no  0   1   1 1   0   0 2   0   0 3   1   2 4   0   0 5   2   1 6   1   3 7   1   4 8   0   0 9   0   0 Page Tables: Page no:1 Frame no:0 for process 1 Page no:2 Frame no:3 for process 1 Page no:3 Frame no:6 for process 1 Page no:4 Frame no:7 for process 1 Page no:1 Frame no:5 for process 2   Enter choice: 1>New process 2>Deallocate 3>Exit |