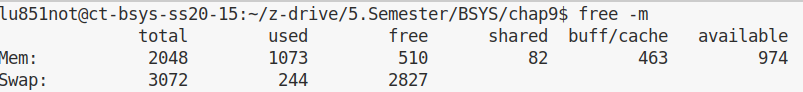
Die Aufgaben wurden im HTWG Container erledigt

**13.2**



Das System des HTWG-Containers hat 2GB Arbeitsspeicher. Davon sind 510MB frei. Diese Darstellung passt auch zu meiner Intuition, da handelsübliche Arbeitsspeicher mit 2 GB auch 2048MB haben. Das 510MB lediglich frei sind, ist auch nicht außergewöhnlich, da 2 GB nicht sonderlich viel sind. Als wir am Anfang ohne free gearbeitet haben, wurde der belegte Speicher zwar nach der Ausführung befreit, aber mit der Zeit wurde der benutzte Arbeitsspeicher immer mehr.

Die Flag -m gibt das ganze in mb an, -g würde es in gb angeben.

**13.4**

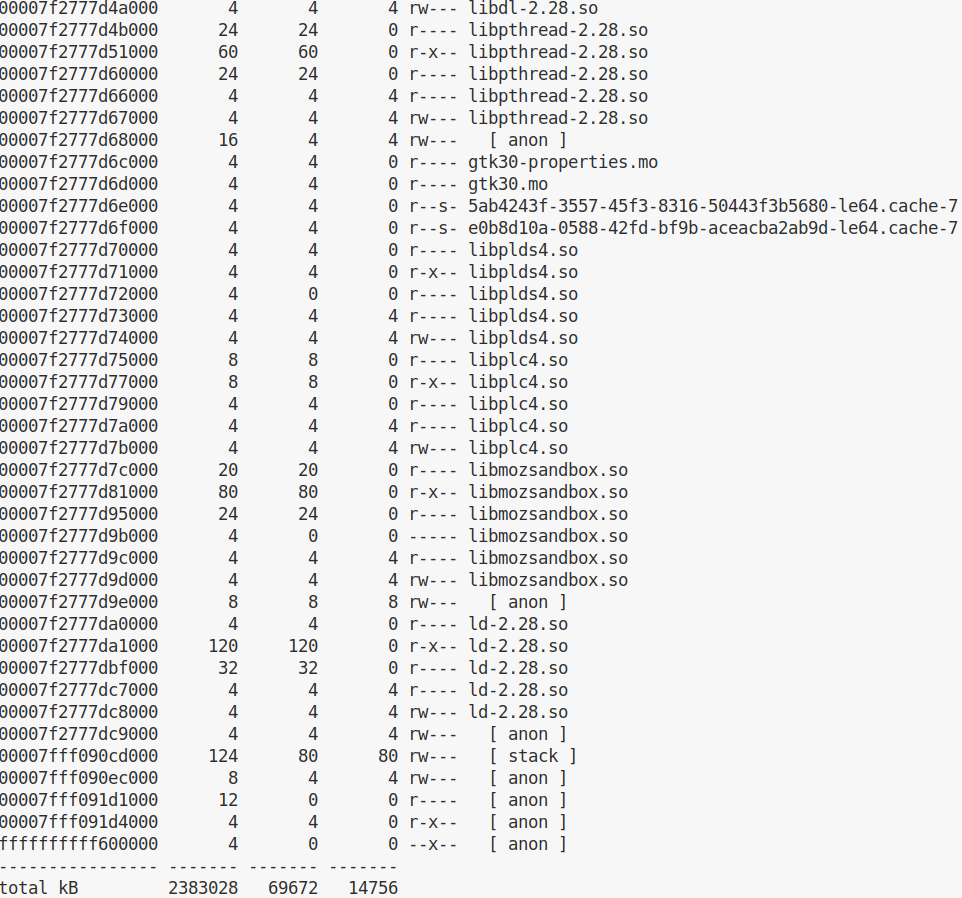
Der verwendete Arbeitsspeicher steigt an, während der freie Arbeitsspeicher sinkt, solange das Programm läuft, da logischerweise Arbeitsspeicher mit malloc reserviert und beschrieben wird. Wenn das Programm endet, wird der allokierte Speicher wieder freigegeben.

Wenn der Prozess memory-user gekillt wird mit z.B. ^C(Strg + c), dann wird der benutzte Arbeitsspeicher wieder freigegeben und der Prozess nicht weiter ausgeführt.

Ich interpretiere, dass 500 MB viel Arbeitsspeicher sind. Wenn viel Memory im Programm benutzt wird (z.B. 500MB), dann wird stetig Speicher reserviert, bis das Programm fertig ist und danach wieder freigegeben. Selbst wenn mehr Arbeitsspeicher reserviert werden soll als es im System gibt, ist dies möglich. Dies sind man anhand der Dirty Pages.

**13.7**

Pmap -x 22100 (Firefox)

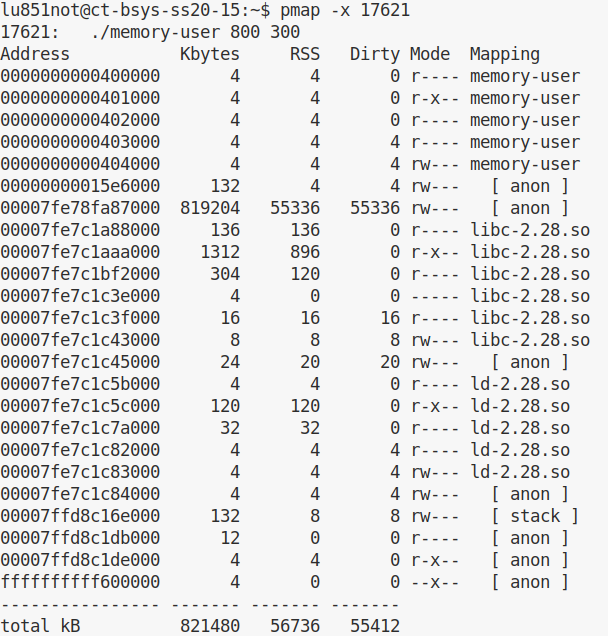
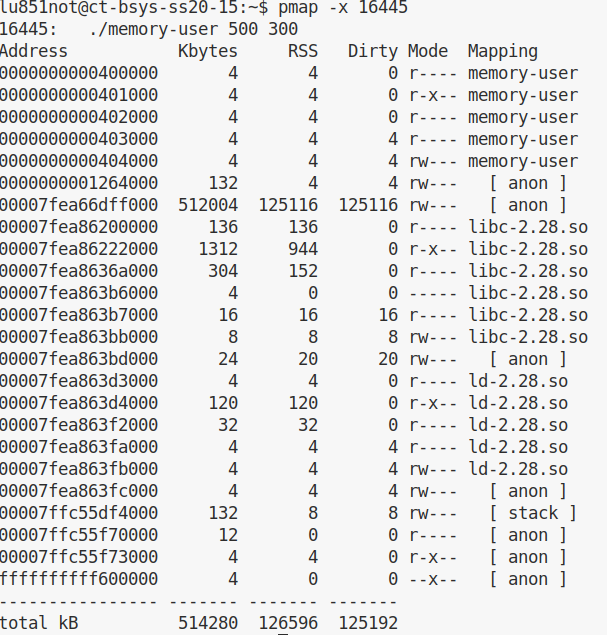


Mit der Flag -x für Pmap bekommt man wie im obigen Bild folgende Prozessdetails (von links nach rechts):

* Die Adresse
* Die Größe der map in Kilobytes
* Residentset größe in Kilobytes
* dirty pages in Kilobytes
* Permissions
* filename (anon für reservierten Speicher, stack für den Programm-stack).

Der Adressraum ist aus deutlich mehr Teilen aufgebaut als unsere einfache Konzeption von Code / Stack / Heap.

**13.8**



Man sieht das mehr Speicher reserviert wird als übergeben. Dies ist nicht verwunderlich, da nur der Array den übergebenen Speicher nimmt. Dazu muss man aber natürlich noch den Rest des Programms oben drauf zählen. An den Screenshots sieht man, wenn man 500mb oder 800mb dem Array zuweist, kommt natürlich am Ende mehr belegter Speicher raus.

Dirty pages werden auf die Festplatte geschrieben, wenn der Arbeitsspeicher zu voll wird. Im ersten Beispiel sind es sogar 125MB, im zweiten lediglich 55MB obwohl mehr Speicher allokiert wurde.

RSS ist die Resident set size und ist die Größe in KB, die tatsächlich auf dem Arbeitsspeicher liegt. Bei der linken Ausführung sind es 126MB und bei der rechten knapp 57MB.