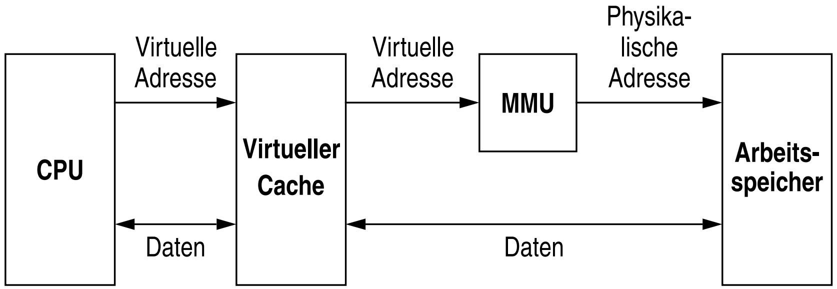
4.3 Virtuelle Caches

4.3.1 Die Arbeitsweise



❏ virtueller Cache

- i860 on-chip cache
- Prüfung der Zugriffsrechte

  write-through

  hit: virtuelle Adresse auch an MMU zur Prüfung

  miss: bei write-allocate mit phys. Adr. aus ASP lesen

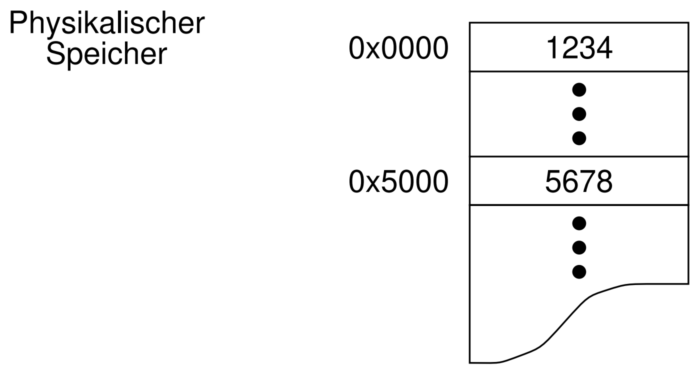
  write-back

  miss: Prüfung verbunden mit laden evtl. Rückschreiben unter Nutzung der MMU

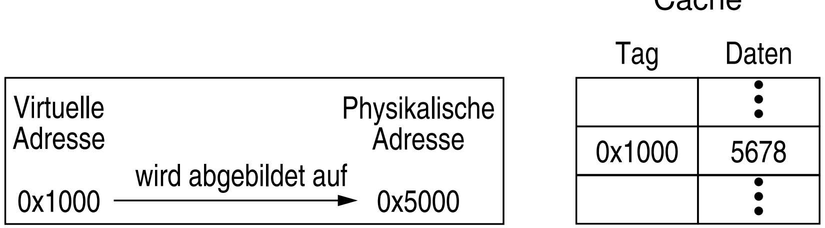
  hit: Falls schon modifiziert, dann erlaubt; falls nicht modifiziert, parallel Prüfung mittels MMU

4.3.2 Probleme

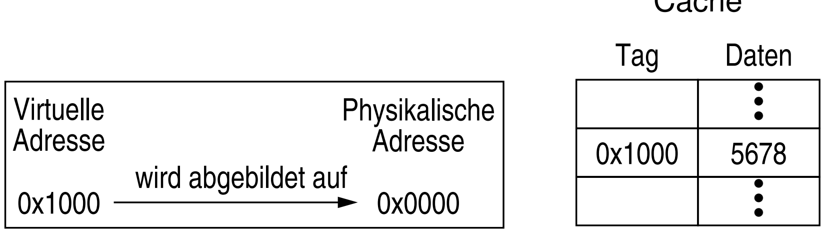
❏ Mehrdeutigkeit (ambiguity)



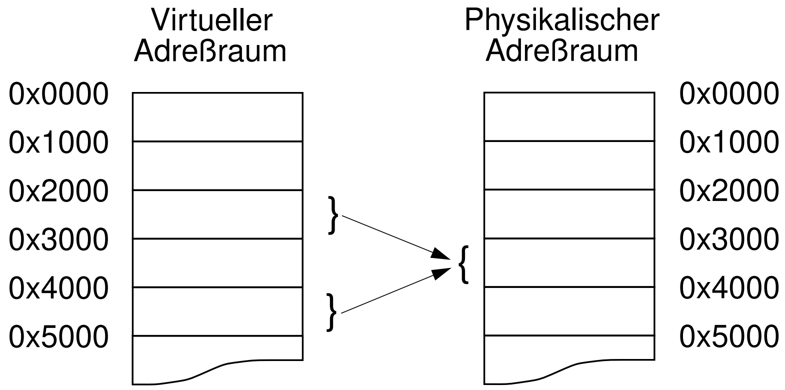
Zeitpunkt 1:



Zeitpunkt 2:



❑ Bedeutungsgleichheit (alias, synonym)



Nach Ansprechen von 0x2000 und 0x4000

Virtueller Cache

Tag	Daten
	⋮
0x2000	1234
	⋮
0x4000	1234
	⋮

Nachfolgende Modifikation von 0x2000

Virtueller Cache

Tag	Daten
	⋮
0x2000	5678
	⋮
0x4000	1234
	⋮

4.3.3 Verwaltung virtueller Caches

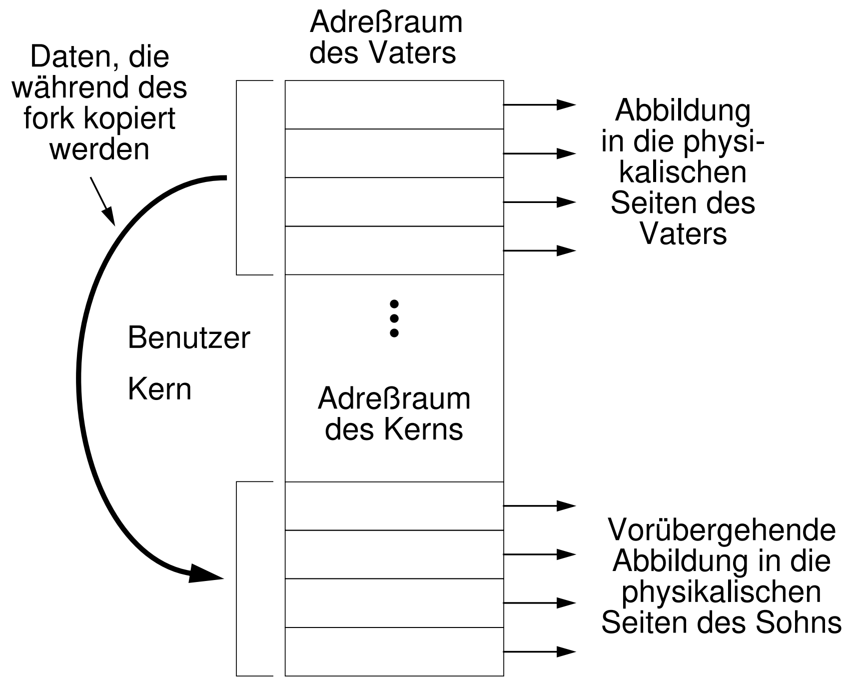
❑ Kontextumschaltung

Da gleiche Adressen in unterschiedlichen Adreßräumen unterschiedliche physikalische Adressen referenzieren können, muß Cache invalidiert werden (Vorsicht bei write-back!)

❑ fork

Bei write-back modifizierte Cachezeilen in ASP transferieren

Wie wird Kopie angelegt, falls kein copy-on-write?



❏ exec

Cache muß invalidiert werden

write-back benötigt kein Rückschreiben, da bisherige Daten aufgegeben werden.

❏ exit

kein Rückschreiben

❏ brk und sbrk

Vergrößerung unproblematisch

Verkleinerung:

❏ gemeinsamer Speicher (shared memory) und speicherabgebildete Dateien (memory mapped files)

führt zum Problem der Bedeutungsgleichheit

Mehrfachzuordnung eines gemeinsamen Segments an einen Prozeß unter verschiedenen virtuellen Adressen:

- verbieten
- nicht in Cache puffern
- Adreßraum direkt abgebildet zusammen mit write_allocate: nur solche virtuelle Anfangsadressen zulassen, die auf die gleiche Cache-Zeile abgebildet werden (Beispiel: Sun 3/200)
- Kachel zu jedem Zeitpunkt nur unter einer virtuellen Anfangsadresse zugänglich machen
- Abkoppeln von gemeinsamen Segmenten wie Verkürzung von BSS

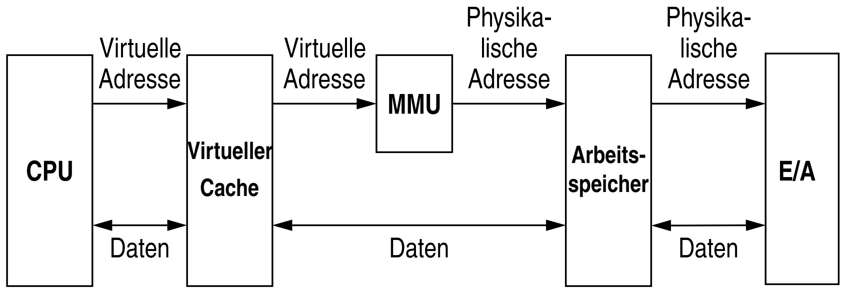
25.06.97 Virtuelle Caches

❏ EA

gepufferte E/A ohne Komplikationen

ungepufferte E/A

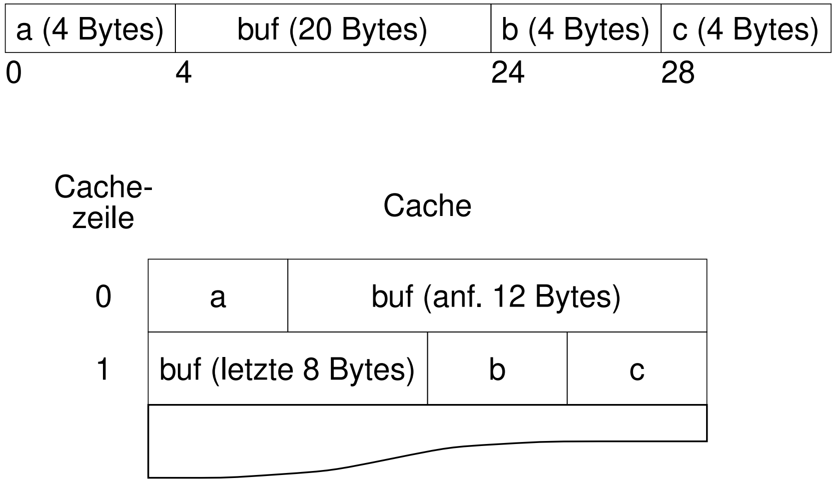
- bei write Informationen evtl. noch im Cache, also Rückschreiben vor E/A-Start
- bei read muß Cache geleert werden



Betriebsprogrammierung II
© 1997, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

Besondere Probleme, wenn E/A-Bereich nicht mit Anfangsadresse einer Cache-Zeile beginnt oder seine Länge kein Vielfaches der Cachezeilengröße ist



- ❏ Mehrdeutigkeiten Benutzer-/Kernadreßraum

  Cache-Invalidierung bei Verlassen des Systemzustands

  write-back: Rückschreiben von Systemdaten muß bei Verlassen des Systemzustands erfolgen

4.3.4 Schlußfolgerung

- ❏ Virtuelle Caches bieten hohe Verarbeitungsgeschwindigkeit, da MMU nur bei Zugriffsfehlern benutzt wird
- ❏ Sie erfordern häufiges Räumen des Cache, was bei großen Caches zeitaufwendig ist
- ❏ Das Betriebssystem muß die Phänomene der Mehrdeutigkeit und Bedeutungsgleichheit berücksichtigen, insbesondere bei asynchroner E/A