Hoofdstuk 1: Inleiding operating systems

## Hoe noemt het hardwareafhankelijk en het hardwareonafhankelijk deel van het besturingssysteem?

* Hardwareafhankelijk: besturingssysteem
* Hardwareonafhankelijk: device drivers

## Wat is een process? Leg dit uit aan de hand van een voorbeeld. Wat is het verband met een programma?

Een programma dat in uitvoering is. Wanneer je de applicatie Word opent, ontstaat er een proces voor dit programma, wanneer je Word terug sluit verdwijnt dit proces.

## Uit welke drie delen bestaat een process? Leg deze drie delen beknopt uit. Welke delen worden gelezen en/of geschreven?

* Codesegment
  + Machine-instructies
  + Program counter: heeft adres van de instructie die wordt uitgevoerd door de CPU
  + Enkel gelezen door de CPU, niet geschreven (code mag zichzelf niet aanpassen)
* Datasegment
  + Data of gegevens waarmee de machine-instructies gaan rekenen
  + Gelezen en geschreven door de CPU
* Stacksegment
  + Tijdelijke gestapelde gegevens waar de stack pointer naar wijst
  + Gelezen en geschreven door de CPU

## Wat is een system call? Leg dit uit. Wat is het verband met de instructieset?

Een system call is een aanroep, door een proces, van een speciale kernel routine van het OS om een bepaalde taak gedaan te krijgen. Zowel een instructieset als system calls worden door device drivers gebruikt.

## Leg met eigen woorden uit wat een command line interpreter is. Hoe noemt men dit in UNIX?

Een programma meegeleverd met het OS waarin je alleen kan werken met tekstgebaseerde commando’s. Een CLI interpreteert de commando’s als system calls. Shell.

## Wat is het verschil tussen een single-tasking en multi-tasking OS?

Single-tasking: kan maar één proces tegelijk uitvoeren

Multi-tasking: kan meerdere processen schijnbaar tegelijk uitvoeren

## Wat is een realtime OS?

Een OS dat binnen een bepaalde tijd op een gebeurtenis moet reageren, bijvoorbeeld bij een vliegtuig of medische apparatuur.

## Hoe wordt multi-tasking bekomen op een single core processor?

Door snel te ‘cyclen’ tussen meerdere processen, hierdoor lijkt het alsof alle processen parallel lopen.

## In welke 3 toestanden kan een process zich bevinden?

* Running
* Ready to run
* Waiting

## Welke twee soorten schedulers bestaan er? Leg ze beknopt uit.

* Pre-emptive: alle processen staan in een wachtrij en mogen om de zoveel tijd aan beurt
* Non-pre-emptive: scheduler laat processen runnen tot ze zijn afgelopen

## Wat is een timeslice? Geef het nadeel van een te lange of te korte timeslice.

Een timeslice is de duur waarin een proces loopt voordat overgeschakeld wordt naar een ander proces (round robin principe). Het nadeel bij een te korte timeslice zou zijn dat een context switch (wanneer de timer afloopt) te lang duurt in verhouding met de timeslice (te veel overhead). Wanneer een timeslice te lang duurt, zou het bij een interactief proces lijken alsof het programma even niet reageert.

## Wat is een blocking system call? Leg dit uit a.h.v. een figuur.

Een blocking system call wordt uitgevoerd wanneer een proces wacht op input, in dit geval is het niet nodig om het proces te laten lopen.

\*start timer\*

Timeslice proces 1: - - - - -

\*timer interrupt\*

Timeslice proces 2: - - - - -

\*timer interrupt

Timeslice proces 3: - - - (proces heeft minder tijd gekregen)

\*blocking system call\*

Timeslice proces 4: - - - - -

## Teken een (vereenvoudigd) toestandsdiagram van een proces.



## Wat is het gevaar van een non-preemptive scheduler?

Omdat er altijd één proces tegelijk afgehandeld afgehandeld wordt bestaat het risico dat een proces te lang duurt. Als er een fout in de code van het proces zit, kan het zelfs voorkomen dat het hele systeem vastloopt.

## Wat is relocatable code?

Programmacode die in eender welk geheugenadres geladen kan worden.

## Waarvoor dient een MMU? Waarvan is het de afkorting? Leg beknopt de werking uit.

Zorgt er voor dat een proces op het juiste adres in het werkgeheugen zit.

Memory Management Unit.

Wanneer een nieuw proces opstart zoekt de CPU vrij geheugen. CPU vertelt de MMU waar het logische proces is. Vertaalt CPU-adressen dan naar fysieke geheugenadressen.

## De figuren op blz. 8 kunnen uitleggen (figuur gegeven).

In het geheugen bevinden zich een aantal processen, samen met het OS. Door de vertaalslag van de MMU lijkt het alsof alle processen op de juiste plaats staan wanneer ze actief zijn. Er loopt echter maar eén proces tegelijk. Een proces dat niet actief is staat dus pas op de juiste plaats wanneer deze actief is, dankzij de MMU.

## Leg het verschil tussen user en kernel mode uit. Hoe kan er gekeken worden dat een user process zijn boekje niet te buiten gaat? Geef van beiden een andere benaming.

User mode is voor user processen. Kernel mode is waar het OS in draait. Het verschil is dat in user mode bepaalde instructies niet uitgevoerd mogen worden.

Wanneer een user proces een system call doet gaat de CPU over in kernel mode. Kernel mode is dan verantwoordelijk voor de uitvoering. Daardoor kan user mode niet buiten zijn boekje gaan.

Protected mode en supervisory mode.

## Welke drie mogelijkheden zijn er indien een computer te weinig geheugen heeft?

* Geen nieuwe processen meer toelaten totdat er processen klaar zijn
* Swapping (virtueel geheugen)
* Demand paging

## Leg demand paging met eigen woorden uit.

Pages zijn kleine stukjes van processen.

Deze worden bijgehouden op de harde schijf en worden enkel aangesproken door het geheugen wanneer ze nodig zijn.

## Geef vier voordelen en drie nadelen van demand paging.

* Voordelen
  + Pages wegschrijven naar disk ipv hele proces
  + Pages die amper nodig zijn staan niet in RAM geladen
  + Je kan processen verwerken die groter zijn dan beschikbaar RAM
  + Geen probleem om passend aaneensluitend stuk geheugen te vinden voor nieuw proces
* Nadelen
  + MMU moet aan eisen voldoen, moderne CPU’s met MMU hebben hier geen moeite mee
  + CPU moet er hardwarematig op ingericht zijn
  + Snelheid is variabel, maar geldt ook voor swapping

## Wat is virtueel geheugen?

Computer krijgt schijnbaar groter geheugen, groter dan fysiek geheugen. Dit is een plaats op de harde schijf waar delen van een niet-actief proces worden bijgehouden.

## De figuur op blz.11 kunnen uitleggen zo de figuur gegeven is.

De MMU laadt bepaalde pages van een actief proces niet. Omdat deze op dit moment niet nodig zijn voor de uitvoering van het proces.

## Hoe wordt swapping in een OS geïmplementeerd (twee mogelijkheden)?

* Swap space: een stuk van de harde schijf wordt vrijgehouden voor pages of processen
* Swap file: er wordt een bestand gebruikt om de pages of processen op te slaan

## Leg het verschil tussen static en dynamic linking uit.

Linking is het koppelen van machine code van bibliotheekroutines aan de code van een programma.

* Static linking: als proces runt wordt alle code in het geheugen geladen, ook de bibliotheek
* Dynamic linking: linken gebeurt wanneer proces de bibliotheek nodig heeft

## Maak een tekening die het verschil uitlegt tussen een low level en high level device driver.

|  |  |  |
| --- | --- | --- |
| High level device driver | Webcam driver | Printer driver |
| Low level device driver | USB driver | |

## Wat is het verschil tussen het werken met interrupts en polling? Leg dit beknopt uit.

Interrupts worden door de I/O devices gestuurd naar de CPU wanneer ze input willen geven.

Bij polling gaat de CPU vragen aan de I/O devices of er input gegeven moet worden.

## Welke vorm van scheduling wordt er tegenwoordig bij een modern OS toegepast?

Dynamic scheduling

## Er zijn twee mogelijkheden van hoe een timeslice wordt toegekend aan een taak. Geef deze twee en leg ze uit.

* Round robin: alle taken krijgen evenveel tijd toegewezen en komen om de beurten aan bod
* Priority scheduling: taken met een hoge prioriteit komen eerst aan de beurt, wanneer taken dezelfde prioriteit hebben wordt er terug gevallen op round robin

## Wat is een resource? Wat is een shared resource?

Een fysiek of virtueel component van beperkte beschikbaarheid binnenin een computersysteem.

Een component dat gedeeld is tussen meerdere computersystemen.

## Leg de begrippen release time, deadline, hard realtime en soft realtime uit.

* Release time: het tijdstip waarop een job gestart moet worden
* Deadline: het tijdstip waarop een job afgelopen moet zijn
* Hard realtime: systeem met een deadline die gehaald moet worden, bv medische apparatuur, computers in een vliegtuig
* Soft realtime: er zit een marge op de deadline

## Waarvan is QoS de afkorting? Gebruik je dan een soft of hard realtime systeem? Waarom?

Quality of Service. Soft realtime. Omdat dit de garantie geeft dat gegevens met een zekere snelheid verstuurd worden. Maar de deadline mag wel overschreden worden.

## Wat is het verschil tussen static of dynamic scheduling?

* Static: scheduling ligt vast, jobs gebeuren in een vaste volgorde
* Dynamic: scheduler kiest zelf welke jobs eerst moeten gebeuren aan de hand van prioriteiten en deadlines

## Leg het verschil uit tussen online en offline scheduling?

* Online: scheduling gebeurt wanneer het systeem loopt
* Offline: scheduling gebeurt voordat het systeem opgestart wordt

## Leg slack bij een job met eigen woorden uit.

Tijd die een afgebroken proces nog nodig heeft om klaar te geraken.

## Welke vier oplossingen heb je als een scheduler niet haalbaar is?

* Snelle CPU
* Los bottlenecks op
* Verschuif deadlines
* Multiprocessorsysteem gebruiken