Systemy operacyjne :: Lista 1

Zadanie 2

```
debian-binary
control.tar.gz
data.tar.gz
```

debian-binary

Wersja formatu deb; treść pliku to prawie zawsze `2.0\n`

data.tar.gz

Pliki pakietu, najczęściej utworzone w środowisku fakeroot'a. Posiadają już potrzebne atrybuty więc wystarczy po prostu je skopiować relatywnie do `/`

control.tar.gz

Dodatkowe pliki wymagane do instalacji pakietu:

control - zawiera główne informacje o pakiecie, m.in jego nazwe, wersje, architekture czy opis. Dodatkowo zawiera informacje dla managera pakietów o tym, jakie jeszcze pakiety są potrzebne do zainstalowania aby pakiet docelowy w ogóle działał, czy też konflikty jakie dany pakiet może wygenerować (np. jakbyśmy chcieli dwie wersje tej samej biblioteki, która jest instalowana w tym samym miejscu)
md5sums - sumy md5 plików z `data.tar.gz`; służy do sprawdzenia poprawności pobranych danych
prerm - skrypt, który wykonuje się przed usunięciem pakietu
postinst - skrypt, który wykonuje sie po zainstalowaniu pakietu

Zadanie 3

> The separation of mechanism and policy[1] is a design principle in computer science. It states that mechanisms (those parts of a system implementation that control the authorization of operations and the allocation of resources) should not dictate (or overly restrict) the policies according to which decisions are made about which operations to authorize, and which resources to allocate.
>
> Policies are ways to choose which activities to perform. Mechanisms are the implementations that enforce policies, and often depend to some extent on the hardware on which the operating system runs. For instance, a processes may be granted resources using the first come, first serve policy. This policy may be implemented using a queue of requests.

Polityka obejmuje:

- podział procesów na klasy,
- określenie sposobu szeregowania dla każdej klasy,
- określenie zasad przydziału procesora pomiędzy klasami,
- zarządznie priorytetami statycznymi i dynamicznymi.

Mechanizmy to środki umożliwiające realizacje polityki:

- procedury przełączające kontekst,
- przerwanie zegarowe i inne narzędzia do odmierzania czasu,
- kolejki i inne struktury danych opisujące stan procesów.

Mechanizmy określają, jak czegoś dokonać, a polityka decyduje o tym co ma być wykonane

Zadanie 4

> A batch job is a computer program or set of programs processed in batch mode. This means that a sequence of commands to be executed by the operating system is listed in a file (often called a batch file, command file, or shell script) and submitted for execution as a single unit. The opposite of a batch job is interactive processing, in which a user enters individual commands to be processed immediately.
>
> The monitor is system software that is responsible for interpreting and carrying out the instructions in the batch jobs. When the monitor started a job, it handed over control of the entire computer to the job, which then controlled the computer until it finished.

Z/OS JCL

JOB - służy do zidentyfikowania bądź nazwania kolekcji zadań, które chcemy aby zostały wykonane
EXEC - jeden bądź wiecej; definiuje kroki danej pracy
DD - służy do wprowadzenia wejścia i wyjścia danych

Zadanie 7

> The process scheduler is a part of the operating system that decides which process runs at a certain point in time. It usually has the ability to pause a running process, move it to the back of the running queue and start a new process; such a scheduler is known as preemptive scheduler, otherwise it is a cooperative scheduler.

Są trzy rodzaje planisty:

- Long-term
- Medium-term
- Short-term (CPU scheduler - zwany także dyspozytorem)

  The short-term scheduler (also known as the CPU scheduler) decides which of the ready, in-memory processes is to be executed (allocated a CPU) after a clock interrupt, an I/O interrupt, an operating system call or another form of signal. Thus the short-term scheduler makes scheduling decisions much more frequently than the long-term or mid-term schedulers – a scheduling decision will at a minimum have to be made after every time slice, and these are very short.

  The dispatcher is the module that gives control of the CPU to the process selected by the short-time scheduler(selects from among the processes that are ready to execute). The function involves :

  Swithching context

  Switching to user mode

  Jumping to the proper location in the user program to restart that program.

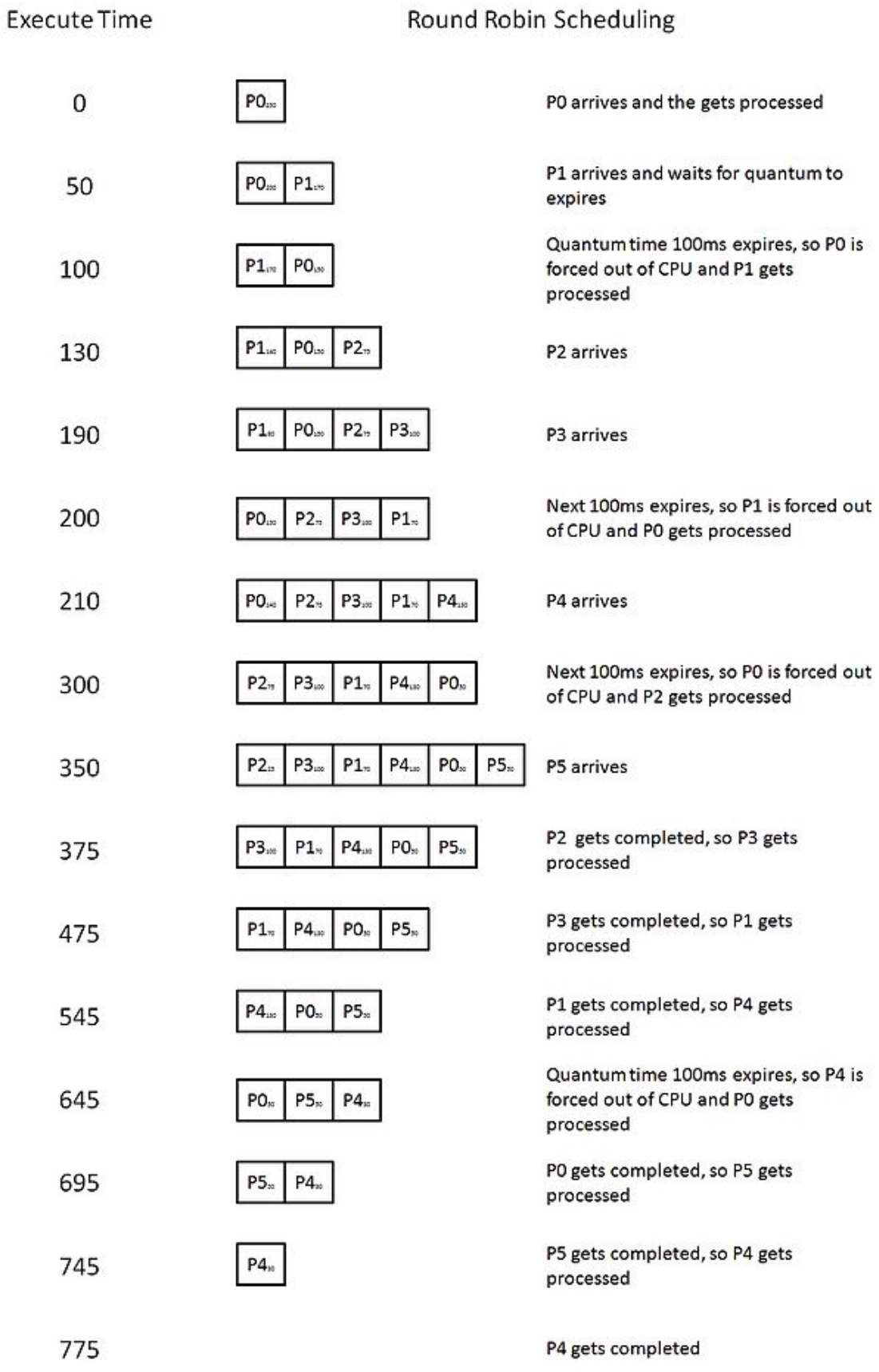

Figure 1: *Round Robin Scheduling*

Zadanie 8

Jądro czasu rzeczywistego

Jest to rodzaj systemu operacyjnego wykorzystywanego do operacji wymagających działania w czasie rzeczywistym. W takich systemach praktycznie nie ma opóźnień związanych z buforowaniem danych. Czasy wykonaia operacji to dziesiątki sekund bądź mniejsze interawły czasowe. Takie systemy są najczęściej `event-driven`, a więc aktualne zadanie zostanie przełączone wtw. pojawi się nowe z wyższym priorytem. Alternatywnie dane zadania mogą być zależne od czasu i wydarzeń (eventów) przy użyciu algorytmu round-robin.

Jądro dla systemów wbudowanych

Takie systemy charakteryzują sie operowaniem w czasie rzeczywistym, przy czym powinny być świadome zewnętrznych wydarzeń (np. jakiegoś układu, który zboczem sygnalizuje nowe dane do pobrania). Taki system także nie musi wspierać żadnego hardware'u, jako że programista sam może napisać odpowiedni program przy takiej budowie systemu. Dodatkowo, taki system pozwala na bezpośrednie używanie przerwań.

Jądro dla sieci sensorów

Podobnie jak w jądrach dla systemów wbudowanych, taki system jest ściśle zależny od zewnętrznych przerwać. Takie systemy muszą być mocno wyczulone na zmiany I/O dlatego też często projektuje je sie w sposób asynchroniczny bądź nieblokujący. Podany w przypisie TinyOS jest właśnie tego przykładem posiadając tylko jeden call stack oraz ansychroniczne I/O, gdzie operacje które wykonują sie dłużej niż pareset mikrosekund są wykonywanie asynchronicznie i po skończonych operacjach wykonują przypisany callback. Dodatkowo scheulder ma prawo np. przełożyć egzekucje zadania o niższym priorytecie na koszt aktualnie ważniejszego. Jako, że zewnętrzne komponenty mogą zgłaszać zadania, w takich kernelach jest stosowane kolejkowanie; najczęściej FIFO.