Aperiodic Server Overview

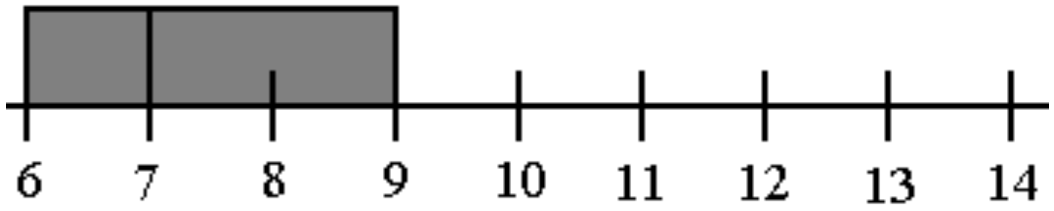
- Background Server gives aperiodic jobs the lowest priority and response time.
- Aperiodic server assigns higher priorities to sporadic jobs
  - An aperiodic server is a special server process which has a queue and a fixed execution periodic budget: e.g. T_k (2,7)
  - If we always have aperiodic jobs waiting to be executed, it's just like a periodic job

Aperiodic Server Overview

- If we don't have an aperiodic job to execute, we can either
  - Idle (discard the budget): wasted
  - Keep utilization, or density, but lower its priority (priority exchange): complex
  - Keep the budget: need to avoid overflow
    - Don't join the ready queue until you get an aperiodic job

Sporadic Server

- A server can maintain the max utilization (*density*)
  - Ex: Out of every 7 time units, SS can use at most 2
  - Budget Replenishment: How much should be replenished?
    - If we start execution at 6, then get a new job at 7, can't execute both jobs, or the density will be too high in the interval (6,13)
- Sporadic server (p_s, e_s) does not demand more processor time than a periodic task with the same parameters in ***any time interval***.



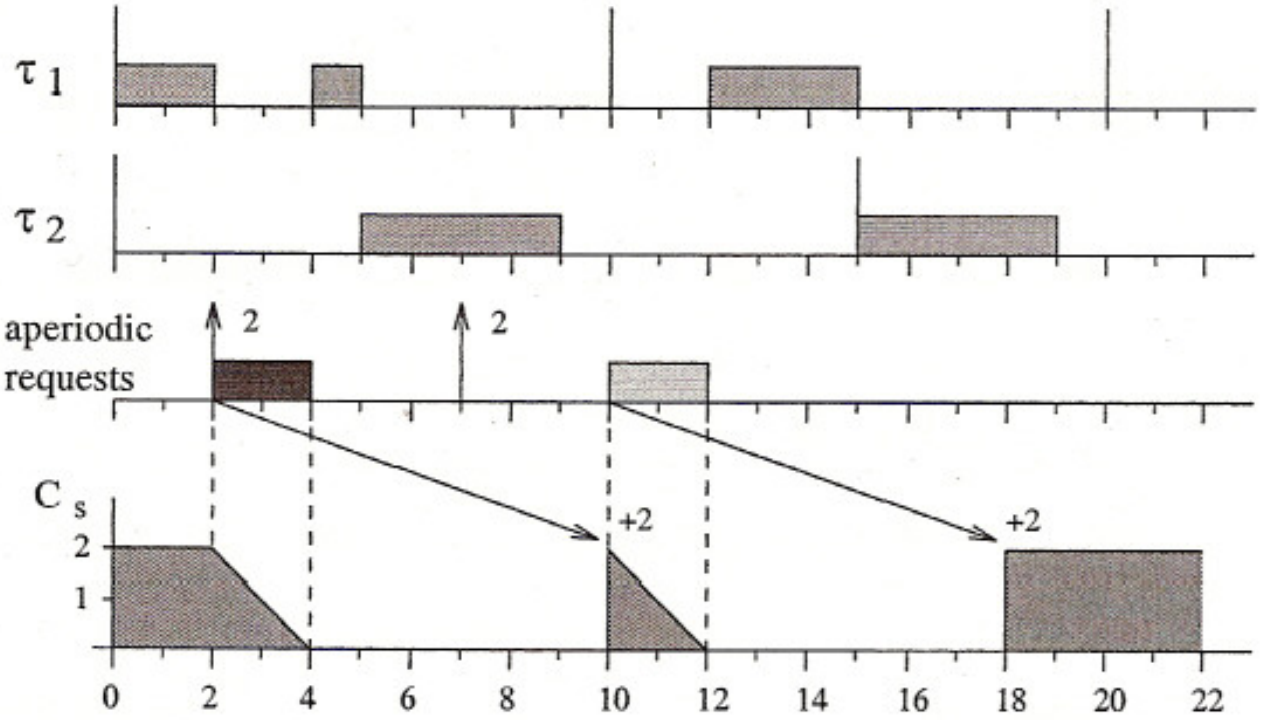
Sporadic Server

	C_i	T_i
τ_1	3	10
τ_2	4	15

Server

$C_s = 2$

$T_s = 8$



Sporadic Server

- Need to determine when to replenish the budget, and by how much

Theorem *A periodic task set that is schedulable with a task τ_i is also schedulable if τ_i is replaced by a Sporadic Server with the same period and execution time.* Sprunt-Sha-Lehoczky

Sporadic Server

- Definitions

P_{exe}	Priority level of the task currently executing
P_s	Priority level associated with SS
Active	SS is said to be *active* when $P_{exe} \geq P_s$
Idle	SS is said to be *idle* when $P_{exe} < P_s$
RT	Denotes *replenishment time* at which the SS capacity will be replenished
RA	Denotes *replenishment amount* that will be added to the capacity at time **RT**

Sporadic Server: Replenishment Rule

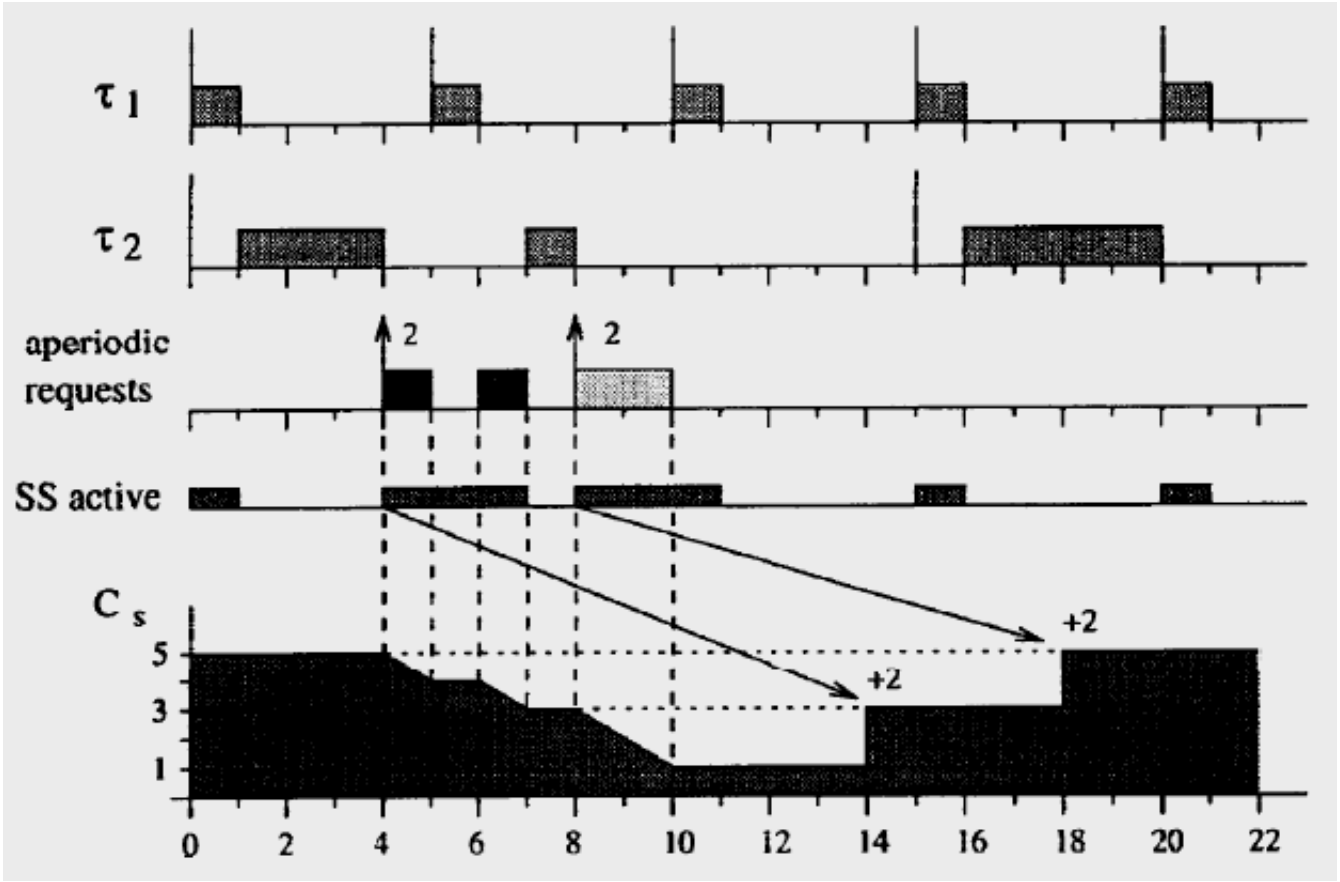
P_{exe}	Priority level of the task currently executing
P_s	Priority level associated with SS
Active	SS is said to be *active* when $P_{exe} \geq P_s$
Idle	SS is said to be *idle* when $P_{exe} < P_s$
RT	Denotes *replenishment time* at which the SS capacity will be replenished
RA	Denotes *replenishment amount* that will be added to the capacity at time **RT**

- **The replenishment time RT is set as soon as SS becomes active and $C_s > 0$. Let t_A be such a time. The value of RT is set equal to t_A plus the server period ($RT = t_A + T_s$).**

- **The replenishment amount RA to be done at time RT is computed when SS becomes idle or C_s has been exhausted. Let t_I be such a time. The value of RA is set equal to the capacity consumed in the interval $[t_A, t_I]$.**

Sporadic Server Example

- Periodic Tasks = (1,5) (4, 15) SS = (5, 10)



POSIX standard

- The Sporadic Server scheme is too complicated for most OS to implement.
- There is an easier way, which POSIX standard employs.
- POSIX has a real-time profile P1003.4 that is to support real-time applications.
  - Don't expect people to implement a sporadic server due to complexity
- PSS – POSIX Sporadic Server
  - Defines a service period / budget
  - Every priority level has a ready queue

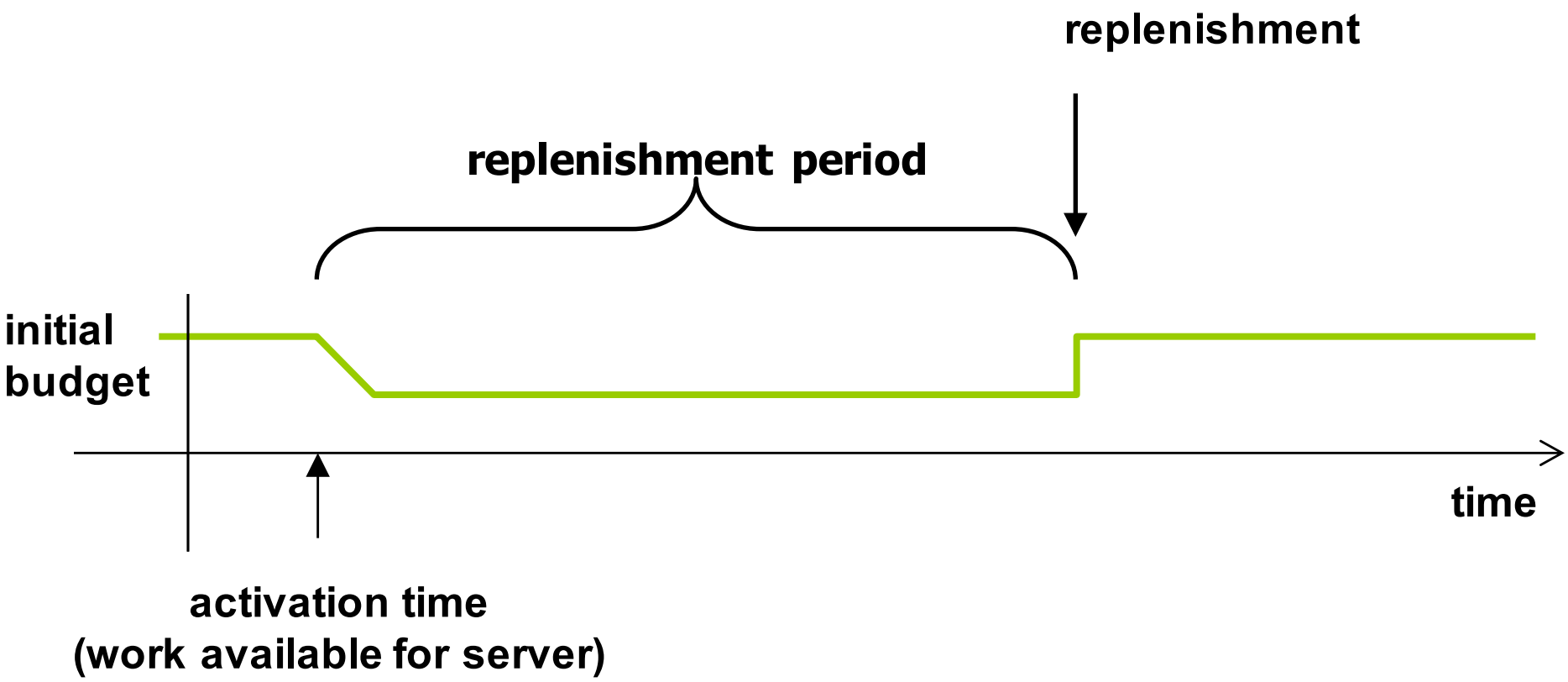
POSIX Sporadic Server Implementation

- A sporadic server assigns a limited amount of CPU capacity to handle events, has a replenishment period, a budget, and two priorities
- The server runs at a high priority when it has some budget left and a low one when its budget is exhausted
- When a server runs at the high priority, the amount of execution time it consumes is subtracted from its budget
- The amount of budget consumed is replenished at the time the server was *activated* plus the replenishment period
- When its budget reaches zero, the server's priority is set to the low value

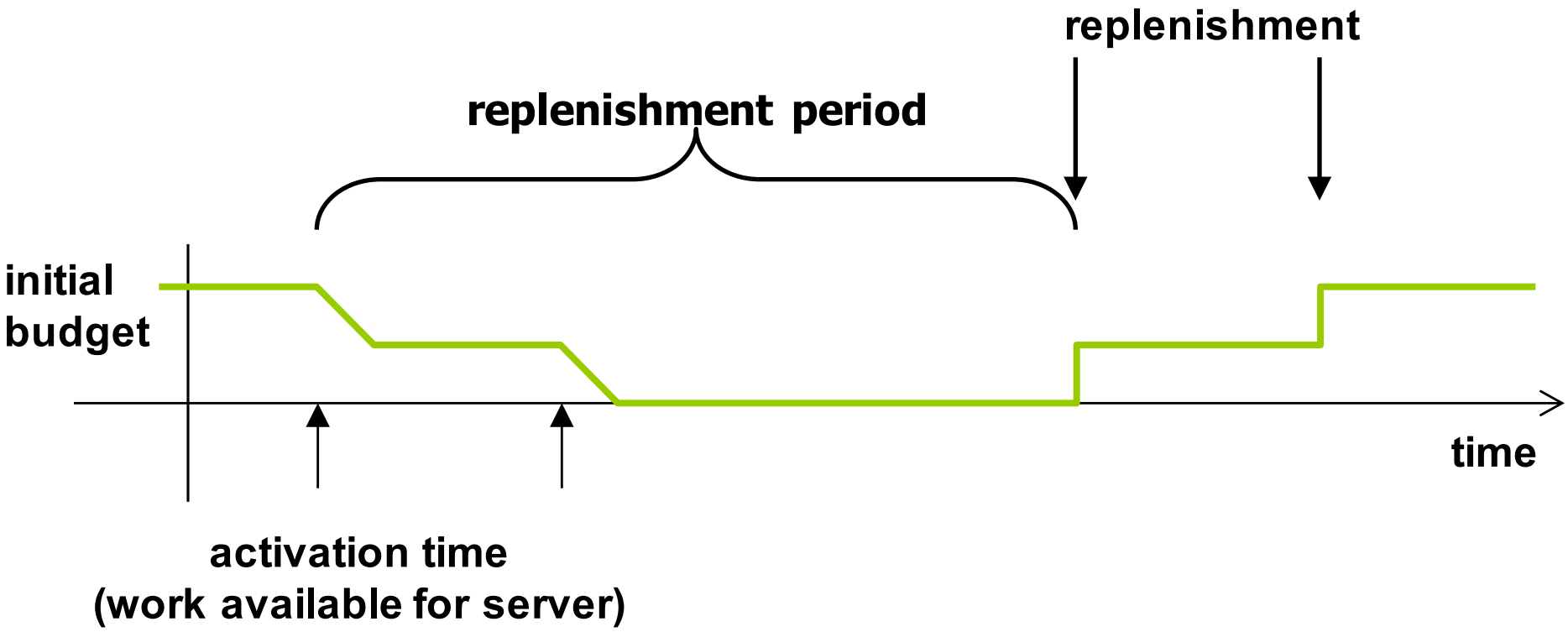
POSIX standard

- Every time an aperiodic job is added to the queue, if there is non-zero budget, the job records the time as the current "activation time"
  - Not the same as t_A if the system runs at a higher priority before the job comes
- The job will be executed if it becomes the highest priority
- When the job is removed from the queue (finished or budget used up) the replenishment time is set to the current "activation time" + P_s

Replenishment Policy



Bandwidth Preservation

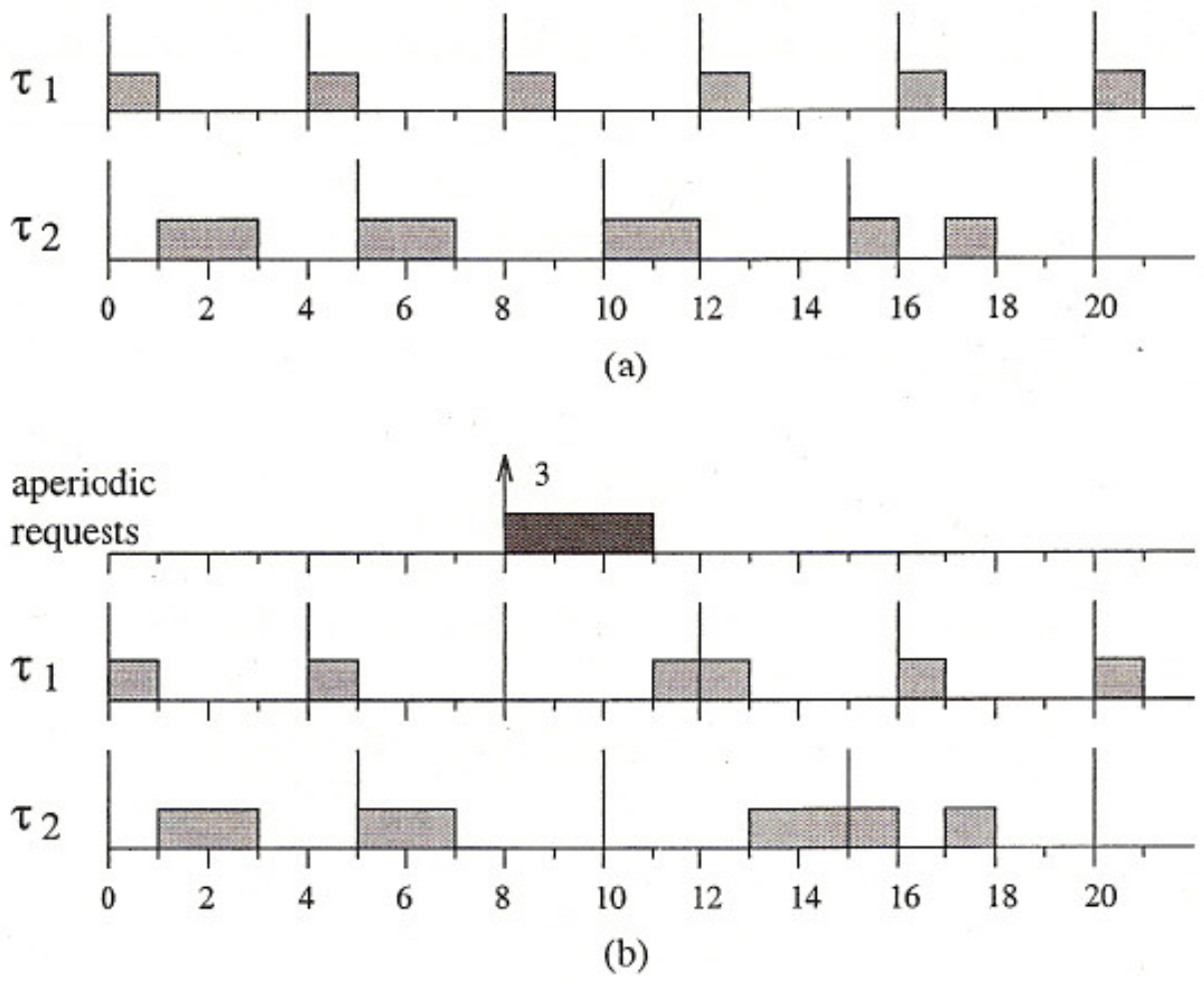


Slack Stealing Aperiodic Scheduling

- Compute the "slack" when an aperiodic job arrives, and decide which to do next.
  - "slack" is the amount of free time in the system.
- If an aperiodic job is important and we want to give it a better priority…

Slack Stealing Example

- If no aperiodic requests are pending, periodic tasks are normally scheduled by RM.
- Slack Stealer steals all the available slack from periodic tasks and uses it to execute aperiodic requests as soon as possible.



Comparison of Fixed-Priority Servers

	performance	computational complexity	memory requirement	implementation complexity
Background Service				
Polling Server				
Deferrable Server				
Priority Exchange				
Sporadic Server				
Slack Stealer