GLOSSARY OF REAL-TIME TERMINOLOGY

Atomic Action Indivisible program part. In reality: smallest uninterruptable programming module used to implement ->semaphores or ->bolts. Unfortunately, uninterruptability cannot be guaranteed if it is not a feature of the ->Realtime Operating System.

Bolts A generalized semaphore used to synchronize the access to resources which allow (exclusive) writing access as well as (shared) reading access.

Clock Device that adds the dimension time to the otherwise only logically arranged computer programs. The clock is used to measure time intervals and/or to synchronize processes. In distributed systems, a global clock is used for these purposes. Unfortunately, global clocks can only be synchronized within certain limits, typically a few microseconds.

Context Switching The switching of the processor from one activity to another. Context switching occurs as different processes are scheduled for execution.

Correctness Correct program execution should be achieved to guarantee ->deterministic behaviour which is of primary importance for real-time programs. Unfortunately, it is impossible to prove the correctness of a program. Therefore, the goal of software design must be to lower the risk to a defined level, depending on the importance of the application.

Determinism The action of a computer system that is completely forseeable and reproducible, a task that can only be accomplished fully in theory.

Embedded Systems (ES) An ES is an electronic system embedded within a plant or an external process. The external process may comprise both a physical/chemical/technical system (usually consisting of subsystems) and also humans performing some supervising or parameter setting tasks. Essential to the external system is that it obeys its own dynamics and does not allow complete restoration of a previous state in case of failure. Coupled with this property is the requirement that the ES has to respond to the external process within prescribed time constraints (->real-time requirement).

Exception An exception is an error situation which may arise during program execution (e.g., division by zero, memory protection violation, etc). An exception may be raised in several programming languages so as to signal that the error has taken place. An exception handler is a potion of program text specifying the response to the exeption.

Fault-Tolerance A method to ensure continued system operation in the presence of faults. Since it it not possible to write non-trivial programs fault-free and, since hardware components will have faults, it is crucial for the reliability of a system to be fault tolerant. It should be remembered that with the advent of Very Large Scale Integrated circuits (VLSI) and by using microprogramming techniques, the faults may already have been incorporated into the microprocessor hardware proper. Countermeasures are: introduction of redundant components both in software and hardware; ensuring ->graceful-degradation or ->exception handling.

Graceful Degradation To ensure -> fault-tolerance, failing system components (both in software and hardware) must be handled in such a way that the effect on the remaining system is only a partial degradation of system performance, not a total system crash.

Hard Real-Time Systems Systems designed to meet given deadlines under any circumstances. Late results are useless and tardiness may result in danger and high cost. Typical examples are fly-by-wire-systems used to control all important elements of an aircraft.

Interrupt Devices may want to interrupt normal program flow intermittently, e.g., to send data. This is done by requesting an interrupt. When the interrupt is acknowledged, the processor suspends the process currently executed (-> context-switching) and calls an interrupt-handler that services the requesting device. Upon completion of the interrupt handler, normal program execution continues, i.e., the processor is switched back to the process that was interrupted.

Interrupt latency The time between the occurrence of an external -> interrupt and the beginning of the execution of the interrupt handler. The interrrupt latency is a measure for real-time performance of a system.

Polling Cyclic checking of an external flag within a program. The program stalls until the flag is set.

Semaphore Originally a person using flags for signaling purposes. Here: a simple mechanism to ensure mutual exclusion or synchronization of separate processes. A semaphore is an integer variable on which two operations ("P" and "V") are defined. Prior to accessing a device, a process must issue a call to the procedure implementing the P operation.

If the value of the semaphore is 0, the process has to wait until the value is greater than 0. If it is greater than 0, "P" decrements the value of the semaphore and the process continues to execute. After releasing the device, the process must call the procedure "V" which increments the semaphore, thereby freeing the device for access by other processes.

INDEX