

During this time, the only use of the CPU is message system traffic to update its data space and calls to `open` and `close` to update file information,

- The primary process calls `checkpoint` to copy portions of its data space and file information to the backup process. It is up to the programmer to decide which data and files to `checkpoint`, and when.
- The primary process calls `checkopen` to checkpoint information about file opens. This effectively results in a call to `open` from the backup process. The I/O process recognizes that this is a backup open and treats it as equivalent to the primary open.
- The primary process calls `checkclose` to checkpoint information about file closes. This effectively results in a call to `close` from the backup process.
- The primary process may call `checkswitch` to voluntarily release control of the process pair. When this happens, the primary and backup processes reverse their roles.

When the backup process returns from `checkmonitor`, it has become the new primary process. It returns to the location of the old primary's last call to `checkpoint`, not to the location from which it was called. It then carries on processing from this point.

In general, the life of a process pair can look like Table 8-1.

TABLE 8-1. *Life of a process pair*

Primary	Backup
Perform initialization	
call <code>newprocess</code> to create backup process	
	Perform initialization
	call <code>checkmonitor</code> to receive checkpoint data
call <code>checkpoint</code>	Wait in <code>checkmonitor</code>
call <code>checkopen</code>	call <code>open</code> from <code>checkmonitor</code>
Processing	Wait in <code>checkmonitor</code>
call <code>checkpoint</code>	Wait in <code>checkmonitor</code>
Processing	Wait in <code>checkmonitor</code>
Voluntary switch: call <code>checkswitch</code>	Take over
call <code>checkmonitor</code> to receive checkpoint data	Processing
Wait in <code>checkmonitor</code>	call <code>checkpoint</code>
Wait in <code>checkmonitor</code>	Processing
Wait in <code>checkmonitor</code>	call <code>checkpoint</code>
Wait in <code>checkmonitor</code>	<i>CPU fails</i>
Take over	(gone)
Processing	