Managing Process Memory Size in Smartphone

Shun Nomura1 Yuta Nakamura1 Takuya Hattori2 Kyosuke Nagata2 Saneyasu Yamaguchi1,2

1. Introduction

Android runs on a variety of devices such as smartphones, music players, and tablet PCs[1]. Android operating system has an original memory management system, which is called ``*low memory killer*''. When enough memory is not available, *low memory killer* terminates processes according to its policy. However, the selection based on the policy does not always meet users' requirement. Thus, it sometimes harms users' convenience.

In this paper, we propose a termination policy based on time consumed by re-launch. Additionally, we implement the policy and evaluate the proposed policy.

2. Application launch procedure in Android

An Android application is launched according to the following procedures[2] as shown in Fig. 1. First, a user touches an icon of an application. Then, Home Applications sends Intent, which is a request to launch an application, to Activity Manager. Second, Activity Manager sends a request of generating a process to Zygote. Third, Zygote forks itself and generates a child process. Fourth, a new process is initialized. Fifth, *onCreate()*, *onStart()*, and *onResume()* in the lifecycle are performed.

3. Low memory killer

Low memory killer is software in Android kernel. It is not in Linux kernel but originally implemented in Android. It terminates Android applications when size of available memory is less than the thresholds, which is defined in "*/init.rc*".

The policy of selection of a process to be terminated is as follows. In "*/init.rc*", sets of *adj* and *minfree* are defined. *adj* indicates class of processes states. Processes on a state with lower *adj* are more important processes. For example, a process on foreground state has the least *adj*, thus it is the last process to be terminated. *minfree* defines a threshold of termination. If size of available memory is less than *minfree*, processes on higher *adj* states are terminated until available memory size is greater than *minfree*. Table 1 shows default *adj* and *minfree* of Android 2.3. If available memory size is 5000 [KB], which is greater than *minfree* of *ajd* 2 and less than *minfree* of *adj* 4, the system terminates processes whose *adj* is greater than 4 until size of available memory become greater than 6144 [KB], which is *minfree* of *adj* 4. The process with the highest *adj* is selected to be terminated. In case several processes have the highest *adj*, the process with the largest memory size is terminated.

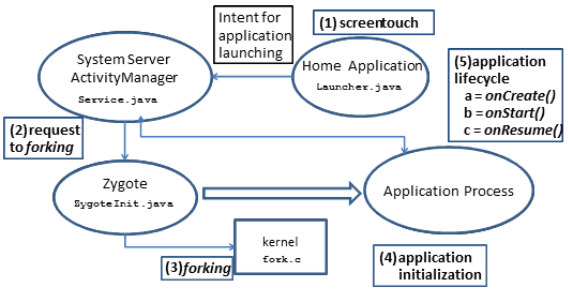

Fig. 1 Android Application Launching Procedure

Table 1 *adj* and *minfree* of Android 2.3

adj	*minfree* [KB]
0	2048
1	3072
2	4096
4	6144
7	7168
15	8192

4. Proposal

In this section, we propose a new selection policy of process termination. With the proposed policy *low memory killer* terminates processes with short launching time. It is expected that large time consumption by restarting long launch time processes can be avoided. With the proposed system, time to launch processes, which is time from user's icon touch to call of *onResume()*, are managed with the method which is proposed in [2]. If launching time of a process is longer than average launching time, *adj* of the processes is

1 Department of information and Communications Engineering, KOGAKUIN University
2 Electrical Engineering and Electronics Kogakuin University Graduate School

decreased by three.

5. Evaluation

We have implemented the proposed policy with an Android smartphone and evaluated the proposed system. The used smartphone is Nexus S. Its CPU is cortexA8 (Hummingbird) Processor 1GHz. Its memory size is 512 MB. Android 4.0.3 is used. The launching orders of experiments are shown in Fig. 2. Benchmark1 and Benchmark2 are our original applications. These do nothing except for allocating 40MB memory. The experimental results are shown in Fig. 3 and Fig. 4 and Fig.5. In these figures, the vertical axes are total time consumed by launching processes. The both figures indicate that the total launching time of the proposed method is less than that of standard Android operating system. We think that the proposed system is more suitable to most users.

6. Conclusion

In this paper, we introduced Android's memory management system, which is called ``low memory killer'', and its issue. For the issue, we have proposed a new policy which can avoid large time consumption by applications with long launching time. Our experiment demonstrated that the proposed policy can decrease total launching time of application. We plan to discuss other policies, such LRU, and compare with them.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers 22700039,24300034.

References:

[1] Nagata Kyosuke, Saneyasu Yamaguchi, Hisato Ogawa, "A Power Saving Method with Consideration of Performance in Android Terminals," The 9th IEEE International Conference on Autonomic and Trusted Computing (IEEE ATC 2012), ATC7-3, (2012)

[2] Kyosuke Nagata, Saneyasu Yamaguchi, "An Android Application Launch Analyzing System," 8th ICCM: 2012 International Conference on Computing Technology and Information Management, (2012)

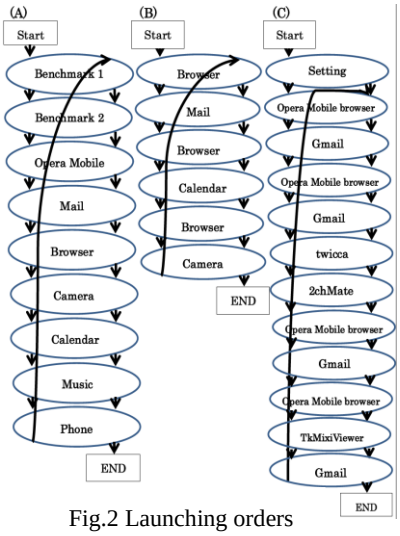

Fig.2 Launching orders

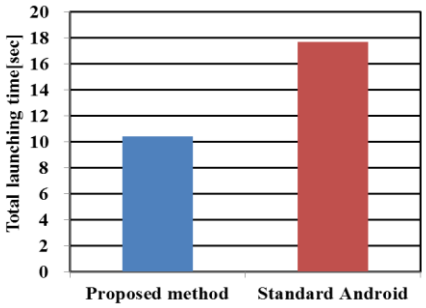

Fig.3 Total time of application launch (A)

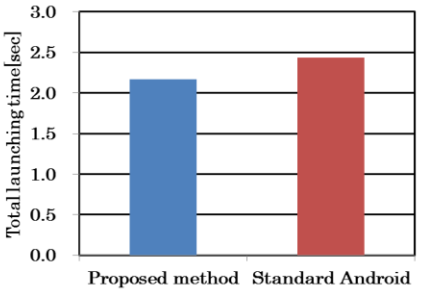

Fig.4 Total time of application launch (B)

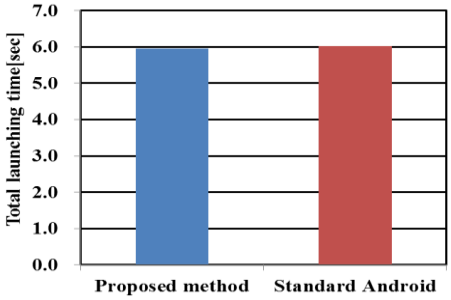

Fig.5 Total time of application launch (C)